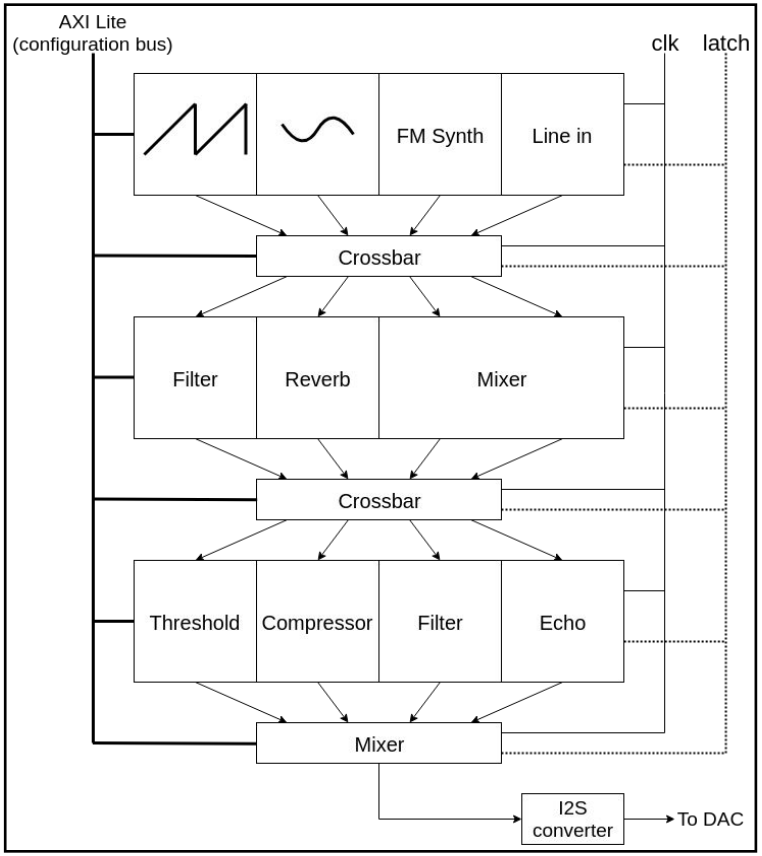
GHOST Synth
Final Presentation

Sriharsha H S
Marco Merlini
Mathew Hildebrand
Austin Liolli

Table of Contents

Overview and Requirements

- SoC-Based Synthesizer
- Multiple voices
- Layer effects with crossbar
- Configure over AXI Lite bus
- I2S Converter sends samples to DAC



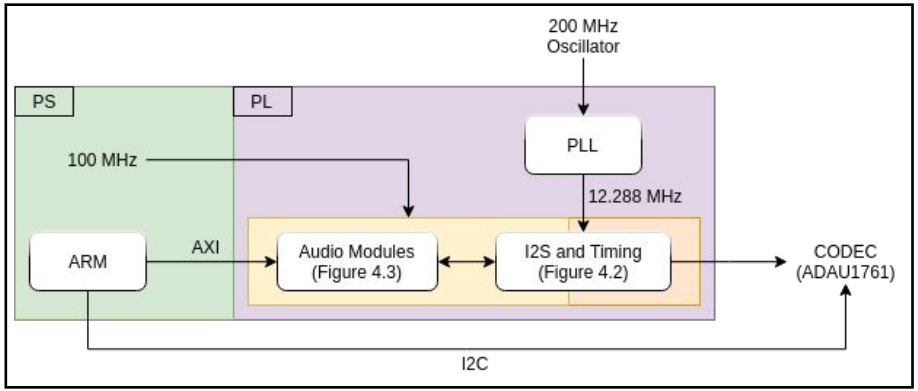
Design Environment

- Vivado HLS 2017.2 for FPGA Image
- Petalinux toolchain 2016.2
- Vivado SDK 2017.2
- Revision control through Git

https://github.com/mathild7/ECE1373_GhostSynth.git

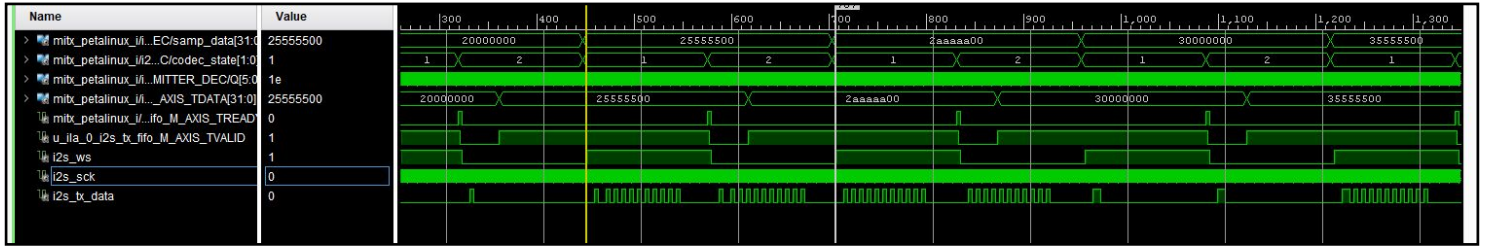
Design Infrastructure

- AVNET Zynq Mini ITX
- ADAU1761 audio CODEC
- 100MHz & 12.288MHz clocks
- FPGA Shell for easy integration



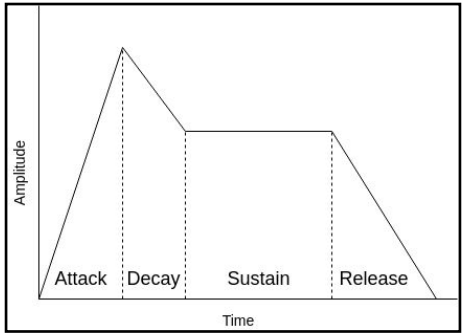
Verification and Testing

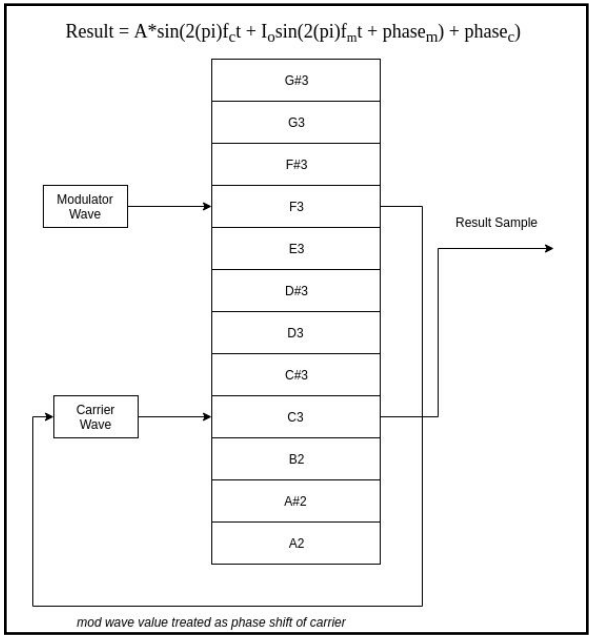
- Incremental inclusion into bitstream
- I2S core, PLL and ATG modules tested first
  - Verify clocking and resets to shell modules
  - Verify audio CODEC can produce audio
- Modules verified using Csim then in Vivado



HLS in GHOST Synth

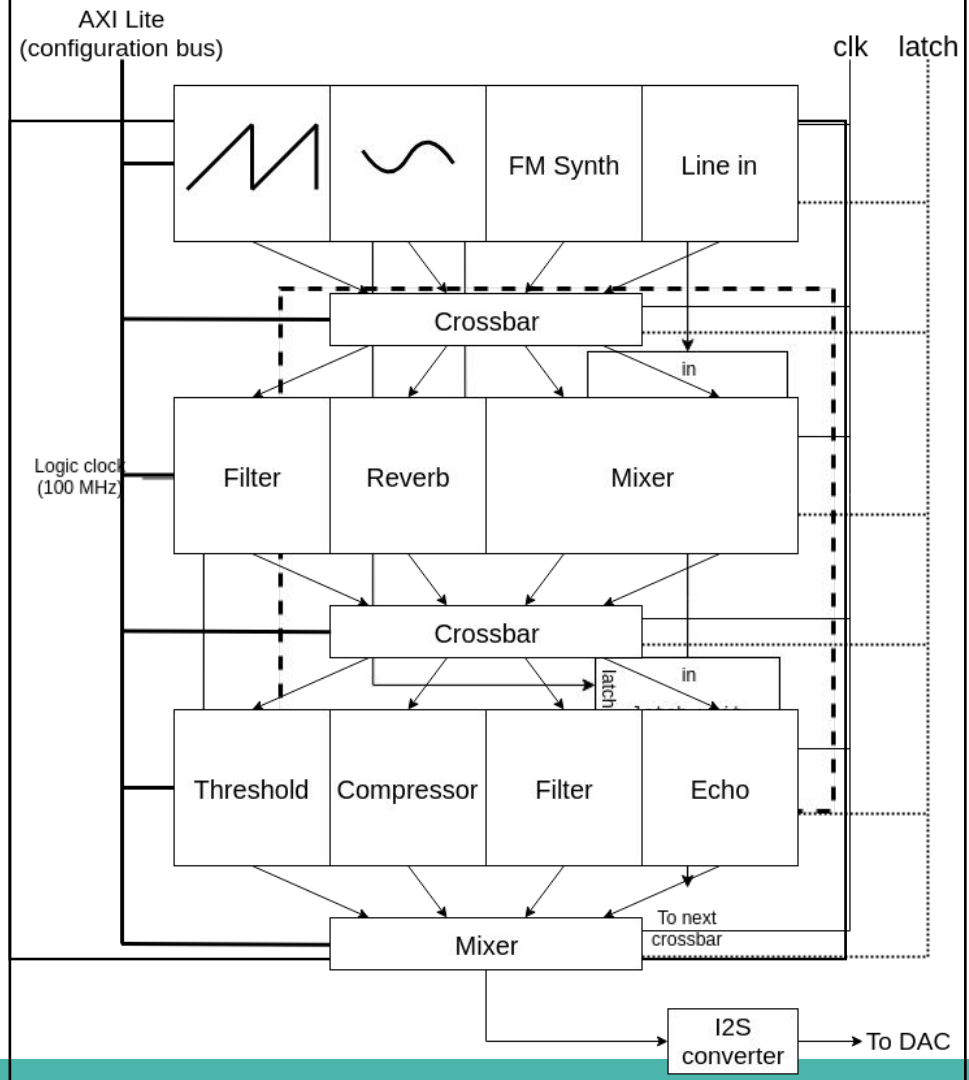
- **Sources**
  - *Saw Wave Generator*
  - *FM Synthesizer*
  - *White Noise Generator*
- **Effects**
  - *Digital Filter*
  - *Echo/Reverb*
  - *Mixer*
  - *Envelope Generator*
  - *Tremolo*
  - *Compressor*
  - *Vibrato*
- **Misc**
  - *Convertor*
  - *Latcher*





Module Interfacing

- `ap_ctrl_none`
- Blocking reads/writes
- Crossbars



Pros and Cons of Using HLS vs. HDL

Pros	Cons
• Easily manage interface protocols • HLS manages wires for AXI stream interface • Natural blocking behaviour	• Weak converting program logic and calculations to efficient hardware • Lack of flexibility in synthesized hardware

- When performance is not critical
  - HLS is 2x faster
- Otherwise
  - About equal

Overall Impressions of HLS

- Easy to use, hard to master
  - If performance is not limiting, implementation is straightforward
  - Significant effort for optimizations
- Information present, but hard to locate

Demonstration

Thank You!

Questions?