

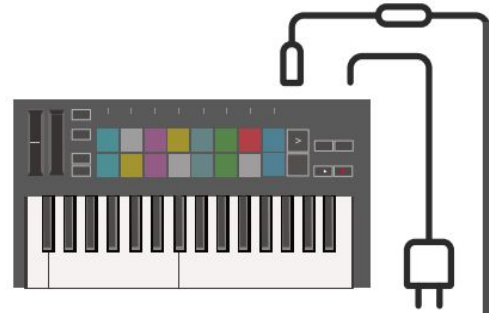
A close-up photograph of a person's hands playing a MIDI controller. The controller features a white piano-style keyboard on the left and a grid of 16 color-coded buttons (green, yellow, red, purple, blue, grey) on the right. The person is also using a laptop, with their right hand on the trackball and left hand on the keyboard. The laptop has an Intel i7 logo and a 'POWER BY DESIGN' logo. The background is slightly blurred, showing green leaves.

# conFFTi - FPGA music synthesizer

Team D8 - Michelle Chang, Hongrun Zhou, Jiuling Zhou

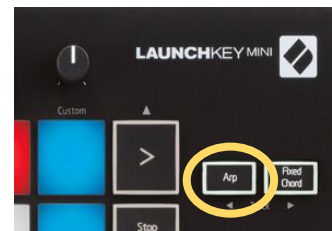
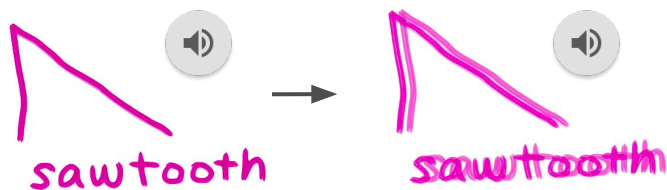
# Application Area

- Accepts real-time input from a MIDI keyboard
- Easy user control over sound generation and mixing on FPGA
  - Waveform generation
    - tri, sin, sqr, saw
  - ADSR Envelope
- Effects that aid music composition
  - **Arpeggiator**
  - **Harmonizer**



# Solution Approach

- DE2-115 Cyclone IV FPGA
  - Large number of logic elements and RAM
- Launchkey Novation MINI MkIII
  - Intuitive knobs and pads
- Implementation of **harmonizer**
  - Feedback loop on waveform oscillator
- Interface of **arpeggiator**
  - Arpeggiator mode and normal mode
  - Hardware support

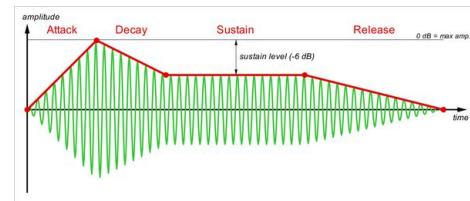
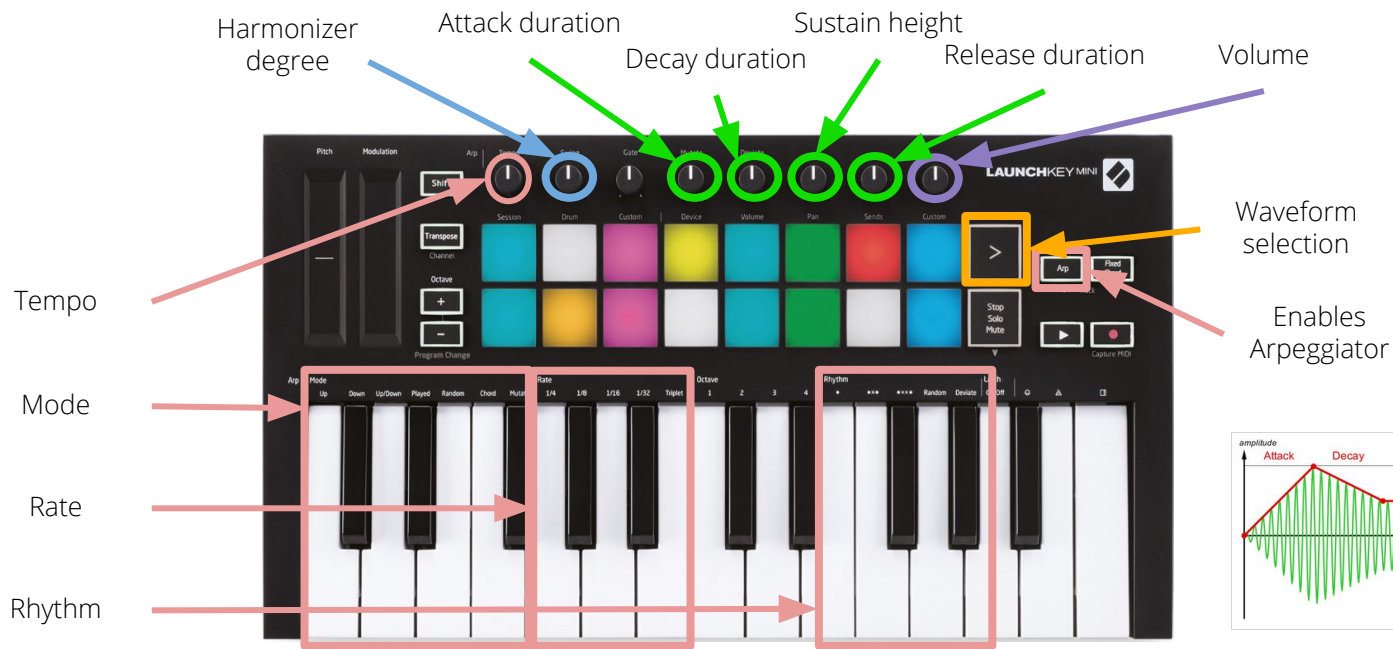


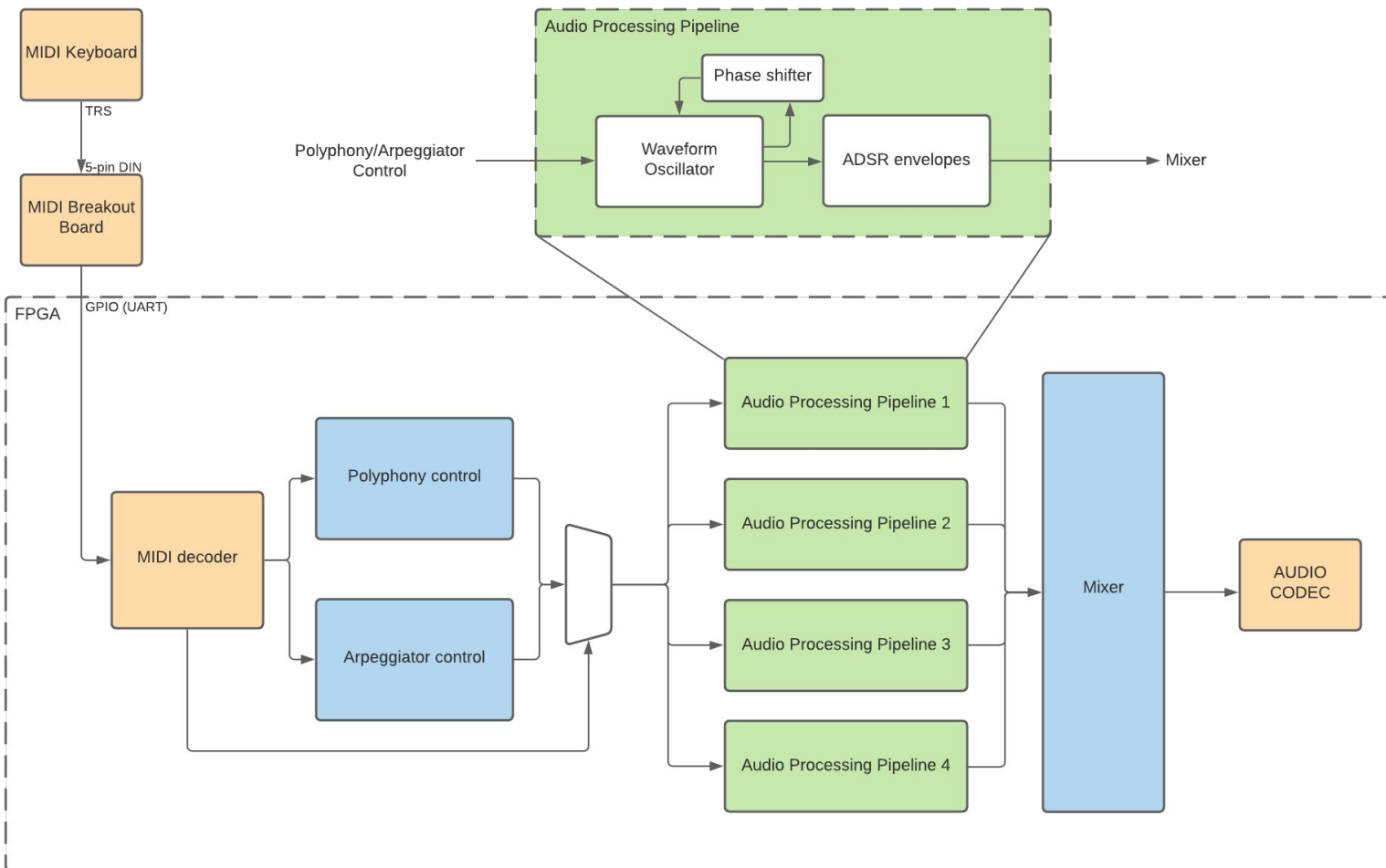
# System Specification - User Interface

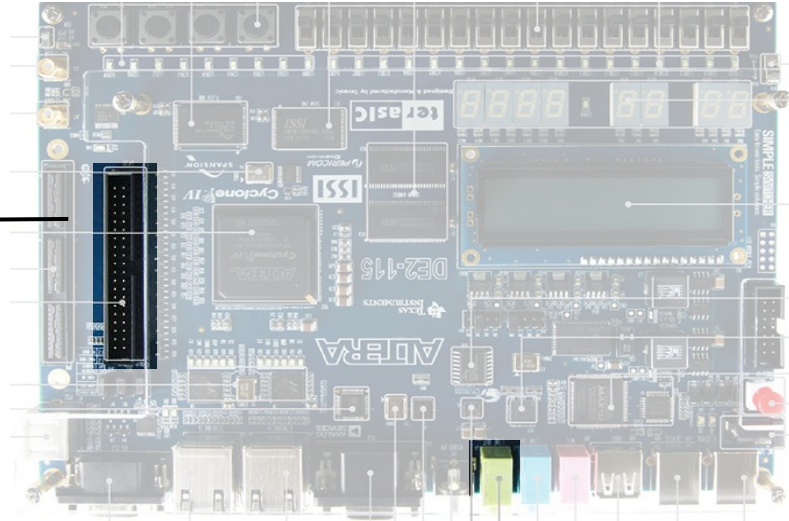
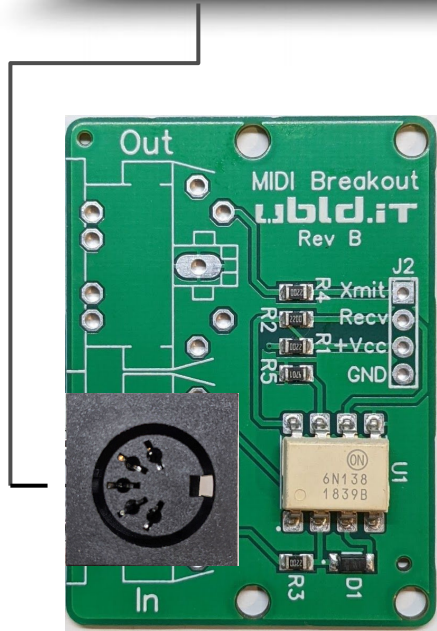
**Arp mode:** Holding the Arp key allows for **Arp** configurations

**Normal mode:** Adjust other effect parameters (**ADSR**, **harmonizer**)

Across both modes: **Volume**, **waveform type**







Line out

# Metrics

- **Industry standard audio quality**
  - 44.1kHz, 24-bit, single channel output
- **Low latency**
  - < 10 ms, from MIDI input to codec
- **Low frequency distortion**
  - < 5 %
- **Low pitch deviation**
  - < 1 %

UART	3 byte/MIDI message * (8 + 2) bits/byte / 31250 baud rate	0.96 ms
Decoder	Combinational	Order of ns
Polyphony	Single cycle	Order of ns
Oscillator	Single cycle	Order of ns
ADSR	Single cycle	Order of ns
Harmonizer	Single cycle	Order of ns
Mixer	Single cycle	Order of ns
Total		< 1 ms

# Verification

- Correctness
  - **Oscilloscope:** Take data points (t, v) from waveform generated by FPGA
  - **Matlab scripts:** Take data points (t, v\_ref) from reference waveform
    - Ensure that  $v\_ref == v$
  - If a test fails?
    - Use oscilloscope for debugging code
- Latency
  - **Oscilloscope:** Quantitatively measure latency using the screenshot function
- Frequency distortion
  - **Oscilloscope:** Apply FFT to waveform generated by FPGA
  - Calculate percentage of distortion of each frequency
  - Calculate total distortion
- Pitch accuracy
  - **Commercial tuner**



# Project Management and Risk Factors

- **Checkpoint 1 (3/20):** Single key input produces single output sound
  - MIDI keyboard, FPGA interface; Sine waveform generation
- **Checkpoint 2 (3/27):** Maximum 4 key input produces mixed sound
  - Polyphony, mixer; Square, sawtooth, triangle waveform generations
- **Checkpoint 3 (4/6):** Effects, basic arpeggiator
  - Harmonizer, ADSR; Arp mode note capturing, looping
  - (risk 1) Arp mode: extensive test cases for FSM
  - (risk 2) Harmonizer: datapath and FSM for feedback loop
- **Checkpoint 4 (4/17):** Arpeggiator effects, verification
  - Control mode, control rhythm

**Risk control:** System integration test and verification performed at the end of each checkpoint.

# Division of Labor and Collaboration

- Michelle
  - LUT and mapping setup
  - Waveform oscillator
  - Harmonizer
- Hongrun
  - MIDI keyboard interface
  - ADSR
  - Audio output
  - Waveform verification
- Jiuling
  - Polyphony control
  - Random number generation
  - Arpeggiator set up
- Collectively
  - Set up local environments for simulation and synthesis
  - Advanced arp features

Feb 26 - Feb 28 **3 d** Matlab scripts - waveform

Mar 1 - Mar 2 **2 d** Matlab scripts - ADSR

Mar 12 - Mar 14 **3 d** Ckpt1 - Sine module LUT

Mar 15 - Mar 17 **3 d** Ckpt1 - Note freq mapping

Mar 18 - Mar 20 **3 d** Ckpt1 - (SV) Sine oscillation

Mar 23 - Mar 24 **2 d** Ckpt2 - (SV) Square oscillation

Mar 25 - Mar 26 **2 d** Ckpt2 - (SV) Sawtooth oscillation

Mar 27 - Mar 27 **1 d** Ckpt2 - (SV) Triangle oscillation

Mar 30 - Apr 6 **8 d** Ckpt3 - Harmonizer (feed back loop)

Feb 28 - Mar 1 **2 d** Matlab scripts - random number generation

Mar 6 - Mar 8 **3 d** Simulation - random number generation

Mar 9 - Mar 12 **4 d** Block diagram - polyphony control

Mar 13 - Mar 17 **5 d** Simulation - polyphony control

Mar 18 - Mar 25 **8 d** Block diagram - Arpeggiator basic

Mar 26 - Mar 29 **4 d** Synthesis & integration - polyphony control

Mar 26 - Apr 2 **8 d** Simulation - Arpeggiator basic

Apr 3 - Apr 8 **6 d** Synthesis & integration - Arpeggiator basic

Feb 27 - Mar 1 **3 d** Matlab scripts - frequency modulation

Mar 13 - Mar 14 **2 d** Ckpt 1 - MIDI keyboard interface

Mar 15 - Mar 19 **5 d** Ckpt 1 - MIDI decoder

Mar 19 - Mar 20 **2 d** Ckpt 1 - Audio CODEC

Mar 23 - Mar 27 **5 d** Ckpt 2 - Develop mixer module to aggregate four pipeline outputs

Mar 30 - Apr 3 **5 d** Ckpt 3 - ADSR Envelope

3 d Ckpt 3 - Normalization

9 days Ckpt 4 - Verification and testing

Mar 2 - Mar 8 **7 d** Simulation environment

Mar 9 - Mar 12 **4 d** Project structure

2 days Ckpt1 - integration

2 days Ckpt2 - integration

2 days Ckpt3 - integration

9 days Arpeggiator

2 days Ckpt4 - integration

14 days Slack time

**2 days of integration  
after each ckpt  
2 weeks of slack time**

