

FPGA GUITAR MULTI- EFFECTS PROCESSOR

Haris Brkic

6.111 Final Project
Presentation

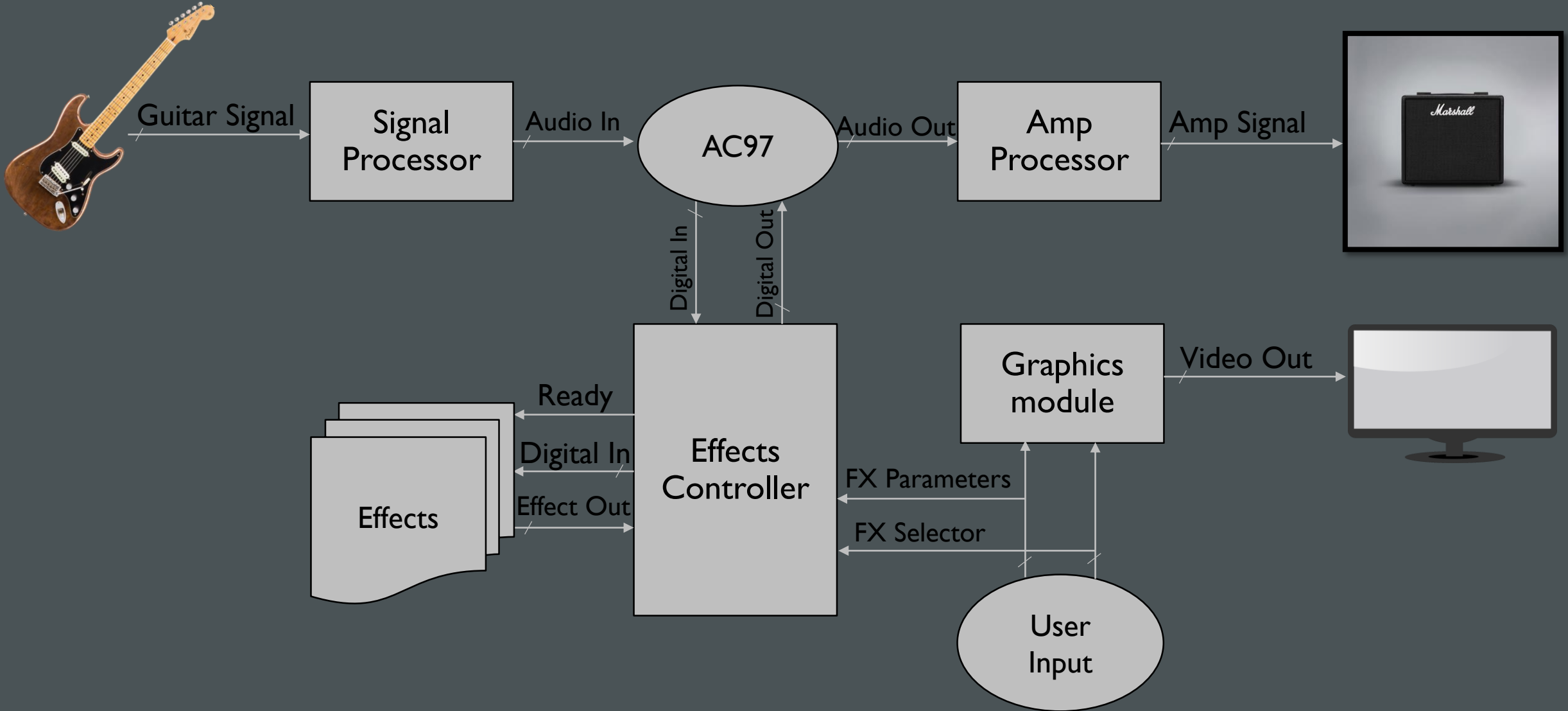


Guitar Effect Pedalboards

Motivation

- Analog Pedals are expensive
- Creating a good pedalboard requires a lot of different pedals
- Digital implementation would avoid the heavy wiring some of the pedals require
- We can create the versions of effects suitable to our personal preferences

BLOCK DIAGRAM



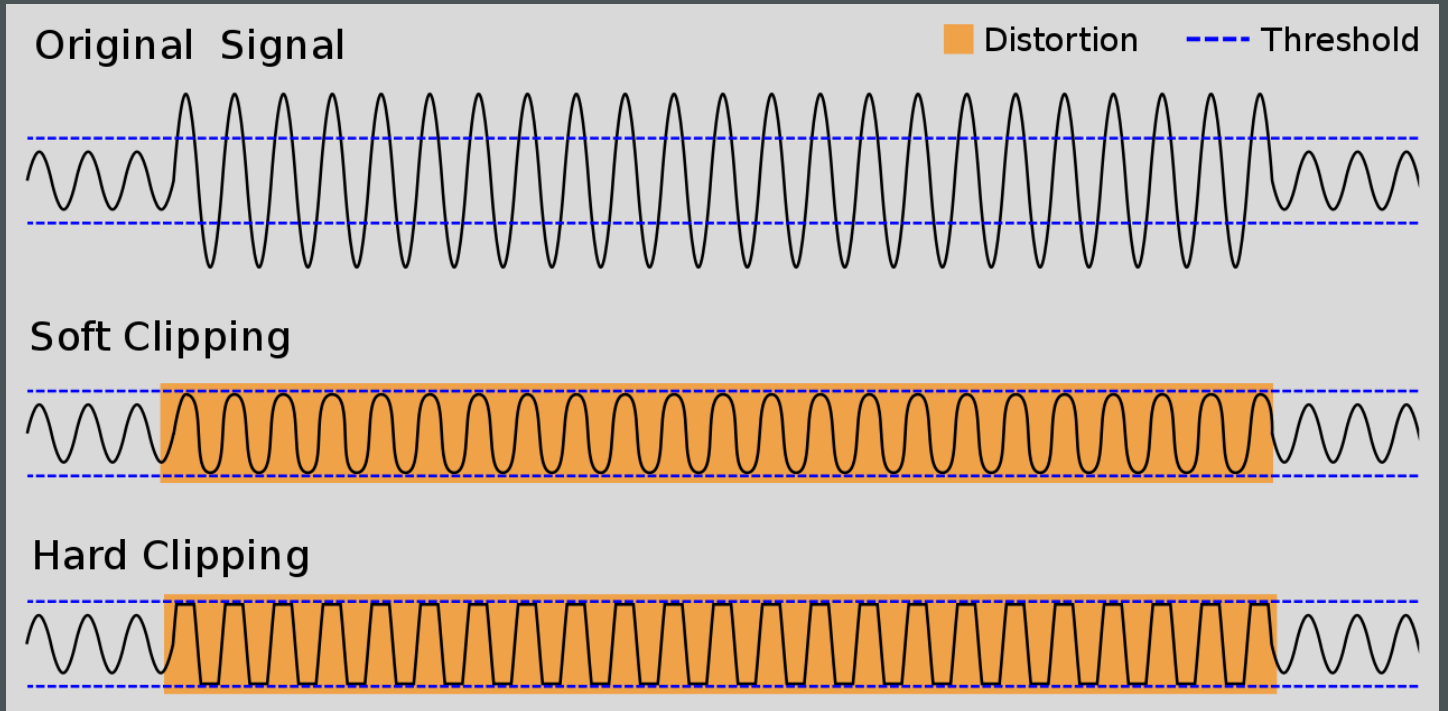
Modules

- *Signal Processor* transforms the signal to allow the AC97 to use it
- *Amp Processor* converts the AC97 output to a guitar amplifier input
- *Effects Controller* sends the digitized signal through the proper chain of effects
- *Graphics Module* displays the currently used effects and parameters

Effects

- *Distortion* – distort the sound and add overtones
- *Delay* – makes an echo effect
- *Looper* – records and loops a phrase
- *Chorus* – simulates a choir
- *Phaser* – creates a slight rippling effect
- *Pitch Shifter* – changes the pitch of the notes played
- *Reverb* – simulates spacious sounds
- *Wah-wah* – creates vowel-like sounds

DISTORTION EFFECT



TIMELINE

- **Week of November 6th:**

- Implement the basic effect modules: *distortion* and *delay*
- Write the *Signal Processor* and *Amp Processor* modules

- **Week of November 13th:**

- Implement the *looper*, *chorus*, *wah-wah*, and the basic *Graphics Module*
- Test if sound passes through the AC97 and is audible from the amplifier
- Implement the more complex effects (*phaser*, *pitch shifter*, *reverb*)

- **Week of November 20th:**

- Test and debug the *distortion*, *delay*, *looper*, *wah-wah*, and *chorus* modules
- Write the *Effects Controller* module
- Check the current *Graphics Module*

- **Week of November 27th:**

- Test and debug the more complex effects
- Implement a more appealing user interface

- **Week of December 4th:**

- Finish testing and debugging all the modules