Developers:
Bradford Westley
Michael King

Advisor:
Dr. William Kreahling

MIPS32 Sound Sequencer

Department of Mathematics and Computer Science

Western Carolina University

Introduction:

- Our capstone project consists of a music sequencer with a front-end GUI in a high-level language and a back-end in assembly language

- Sequencer also has the capability of generating music sequences from scratch.

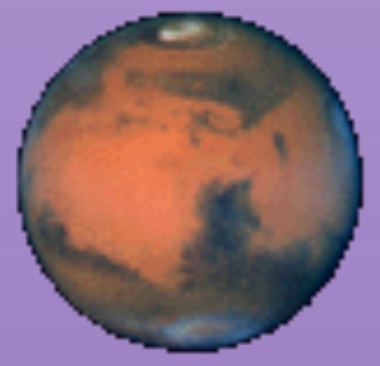
Back-end:

- Underlying code that is not directly accessible by the user

- Programmed in MIPS assembly language

MIPS TECHNOLOGIES

- Executed by MARS (MIPS Assembler and Runtime Simulator)

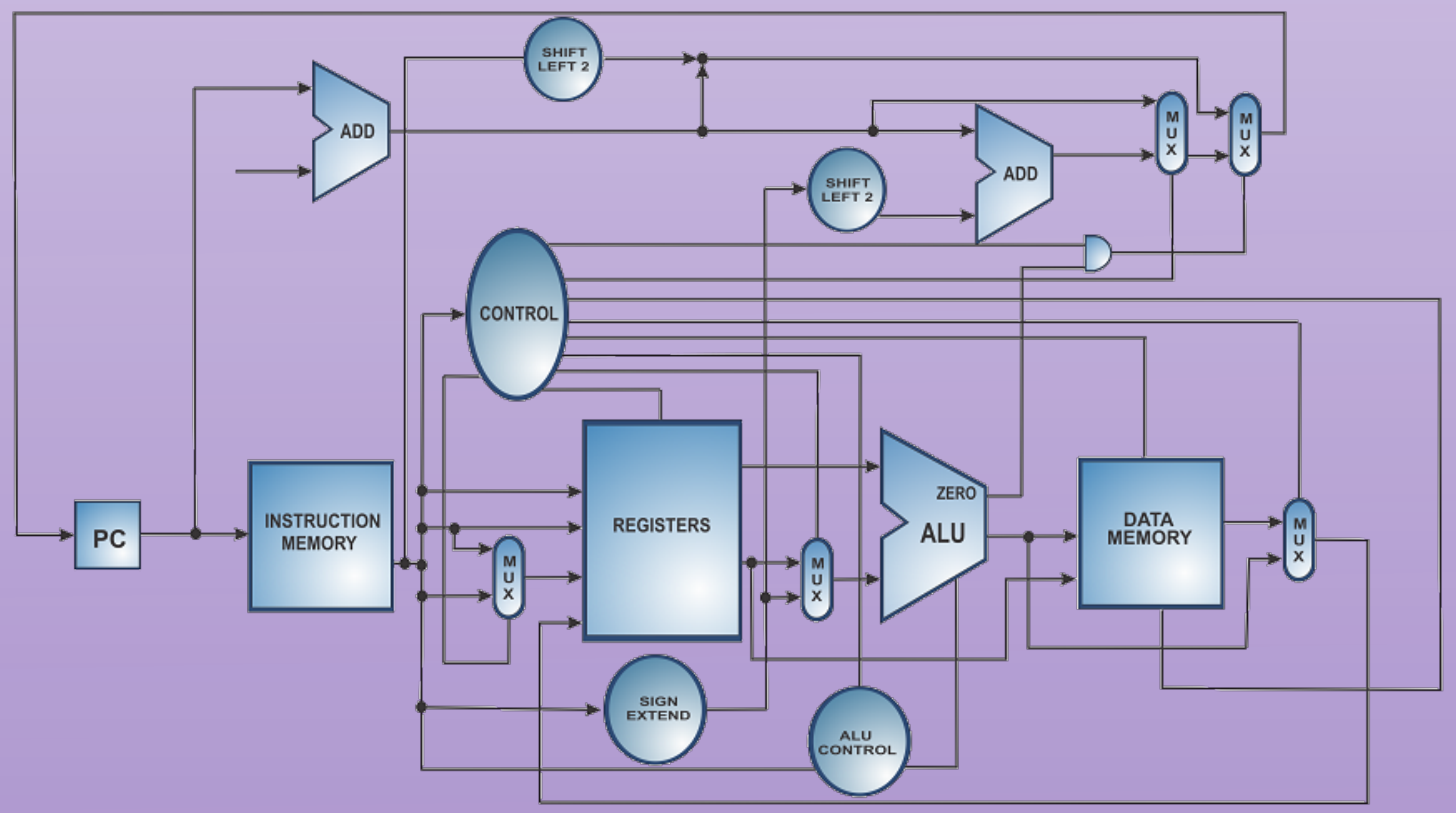


- The back-end plays the sequences that a user creates, and also can generate a sequence of notes

Background:

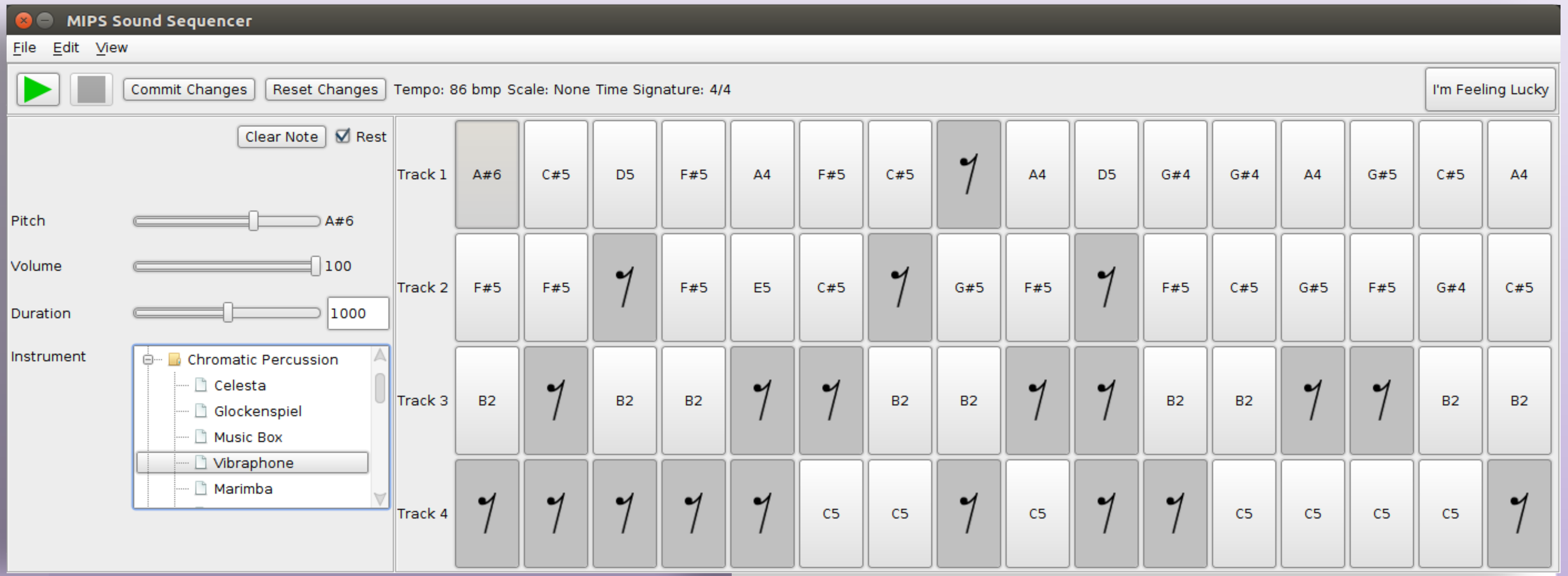
- MIPS uses registers and memory to store data and execute instructions.

- Data flows through a multistage path (pictured below) within the CPU to achieve this.



- Our sequencer utilizes MIDI (Musical Instrument Digital Interface) to generate digital audio tones

MIDI



Tempo: 86 bmp Scale: None Time Signature: 4/4

☑ Rest

Pitch A#6

Volume 100

Duration 1000

Instrument
- Chromatic Percussion
  - Celesta
  - Glockenspiel
  - Music Box
  - Vibraphone
  - Marimba

Track 1	A#6	C#5	D5	F#5	A4	F#5	C#5	𝄾	A4	D5	G#4	G#4	A4	G#5	C#5	A4
Track 2	F#5	F#5	𝄾	F#5	E5	C#5	𝄾	G#5	F#5	𝄾	F#5	C#5	G#5	F#5	G#4	C#5
Track 3	B2	𝄾	B2	B2	𝄾	𝄾	B2	B2	𝄾	𝄾	B2	B2	𝄾	𝄾	B2	B2
Track 4	𝄾	𝄾	𝄾	𝄾	𝄾	C5	C5	𝄾	C5	𝄾	𝄾	C5	C5	C5	C5	𝄾

Sequence Generation:

- We developed an algorithm responsible for generating a sequence in MIPS assembly.

- Each track will generate a different melody, but will be harmonized to each other track.

- The sequencer can also generate rhythms that would carry the "beat" of the sequence and use MIDI purcussion instruments.

Front-end:

- Interface between the user and the back-end

- Uses a subset of the Java programming language called Java Swing.

- We used Java Swing's ability to make grid-like patterns for an easy-to-understand interface

Java™