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.

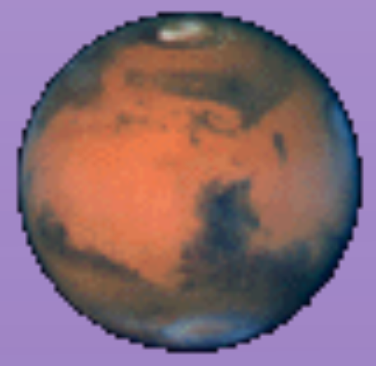
Back-end:

- 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)

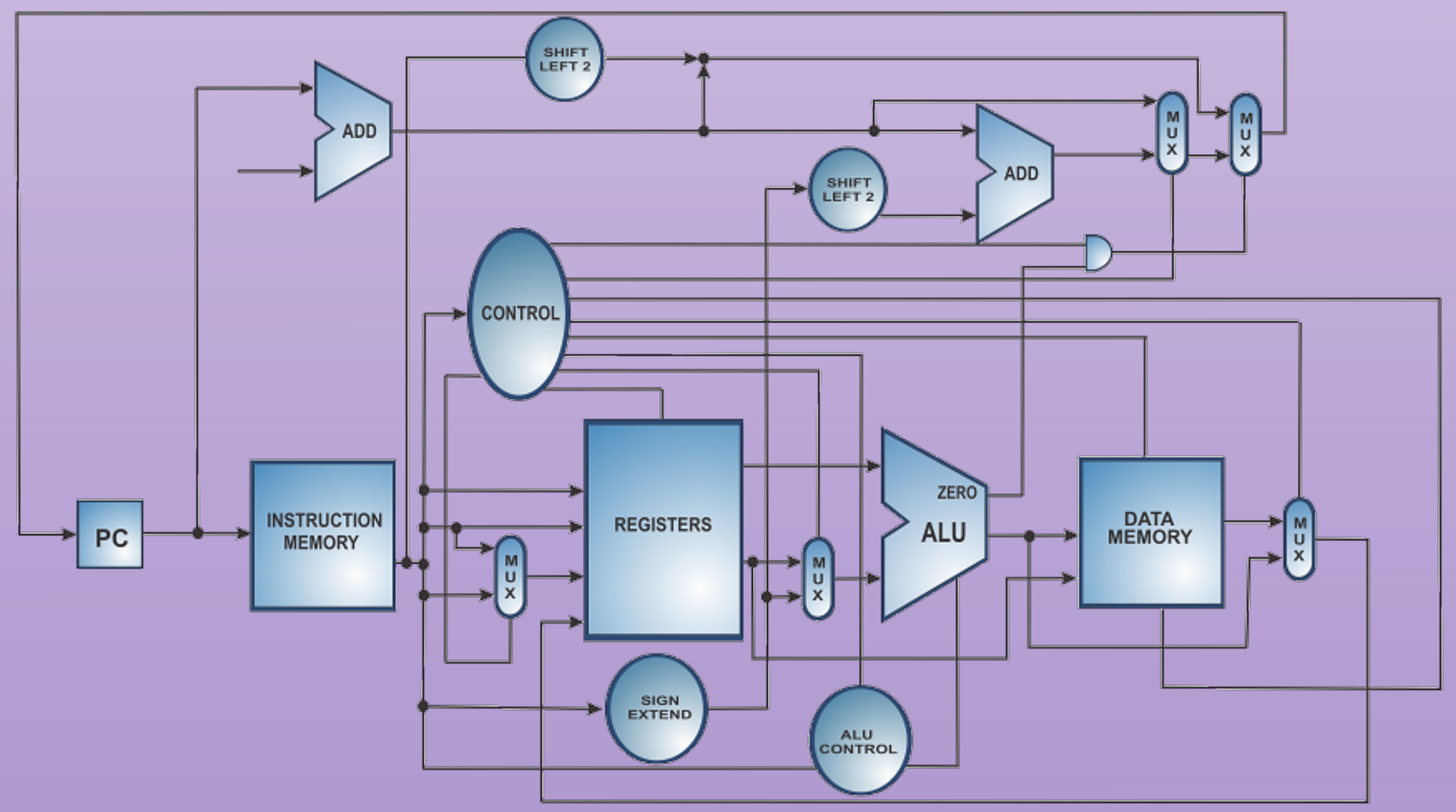


- The back-end plays the sequences that a user creates, and also can generate a sequence of notes

Background:

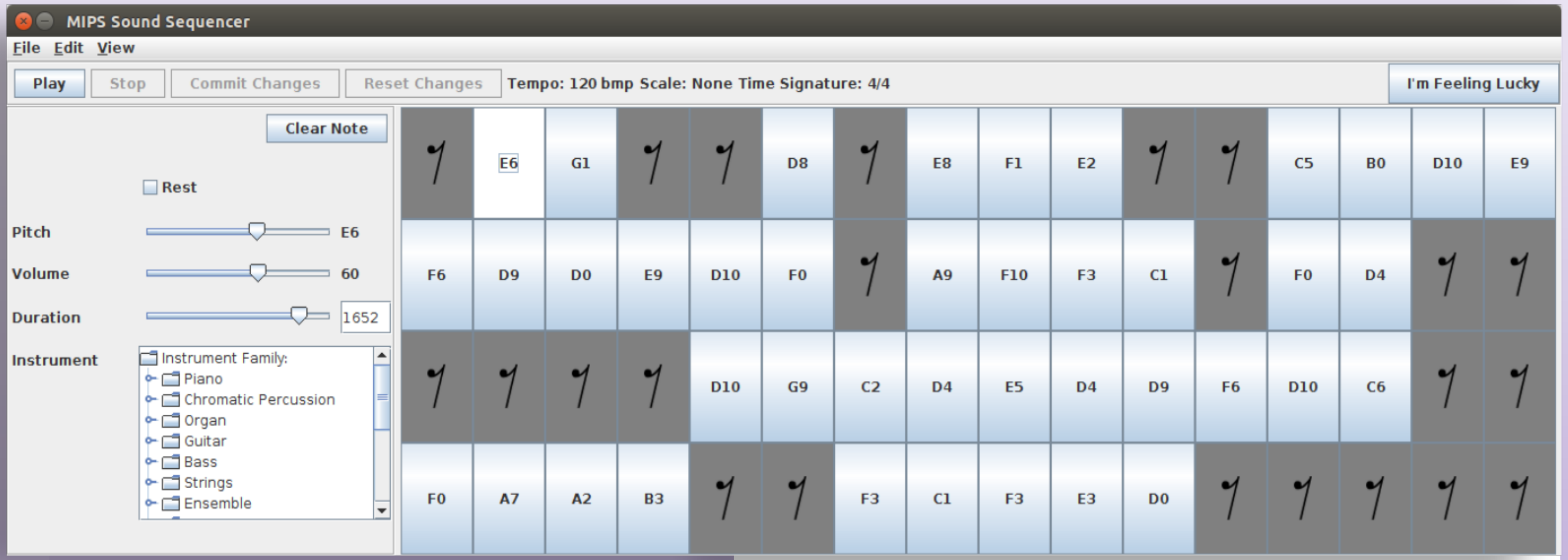
- MIPS uses registers and memory to store data and execute instructions.

- 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: 120 bmp Scale: None Time Signature: 4/4

□ Rest

Pitch E6

Volume 60

Duration 1652

Instrument

Instrument Family:
- Piano
- Chromatic Percussion
- Organ
- Guitar
- Bass
- Strings
- Ensemble

𝄾	E6	G1	𝄾	𝄾	D8	𝄾	E8	F1	E2	𝄾	𝄾	C5	B0	D10	E9
F6	D9	D0	E9	D10	F0	𝄾	A9	F10	F3	C1	𝄾	F0	D4	𝄾	𝄾
𝄾	𝄾	𝄾	𝄾	D10	G9	C2	D4	E5	D4	D9	F6	D10	C6	𝄾	𝄾
F0	A7	A2	B3	𝄾	𝄾	F3	C1	F3	E3	D0	𝄾	𝄾	𝄾	𝄾	𝄾

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™