Introduction:

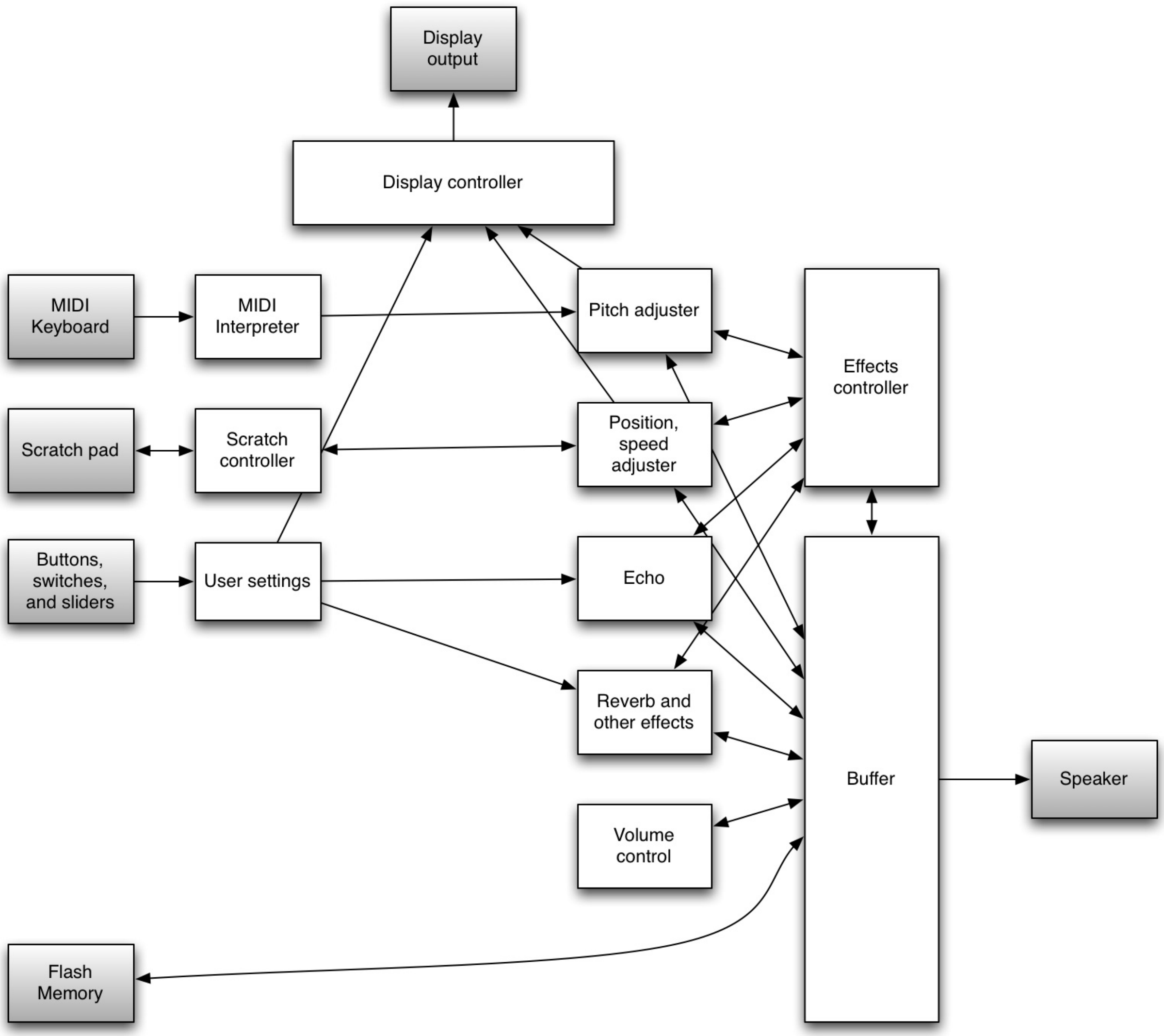
Our goal is to design a DJ mixing setup. With interfaces including buttons, switches, sliders, and motors, the project would make it possible to control the playback and sound of one or more recordings of music.

The major features will include: layering tracks, fading between tracks, changing the pitch of tracks, slowing down or speeding up tracks (with or without changing pitch), lining up the beats of multiple tracks, and "scratching" tracks by rapidly jumping within them.

A simultaneous visual display would present information including: where playback was in the waveform of the track, how much the pitch and speed of playback had changed from the original, what frequencies were currently dominant, and which tracks were coming up.

We plan to store pre-recorded tracks—and record the result of remixing tracks—on flash memory.

Overall schematic:



Input Devices:

There will be several input devices.

1. **MIDI Keyboard**
   An attached MIDI keyboard will trigger pitch changes in the music.

2. **Sliders/knobs**
   Potentiometers, connected to A-to-D converters, will make it possible to change the volume of different tracks and fade between them.

3. **Scratch Pad**
   The scratch pad will make it possible to change the playback position and speed of the music. The position of the scratch pad will be measured with an optical encoder.

Flash Memory:

The flash memory will be used in two different ways.

First, songs will be streamingly read from the flash ROM; Data is read in 16 bit chunks, which are buffered in a FIFO. For a 32 MBit chip, the read spead is 110 ns per read cycle; this should be fast enough, depending on the size of chunks we are editing and the speed at which the filters work. In the worst case, the FIFO can be enlarged.

Second, filtered songs will be recorded back onto the flash. The write speed is around 6.8 microseconds per Byte of written data.

One difficulty will be reading and writing to the flash memory simultaneously. This involves constantly switching WE (write enable) signal, which introduces more timing delays, as well as other setup delays. To solve this problem, data will be read and written in seperate chunks, who are stored in read and write buffers. The size of these buffers is yet to be determined.

Effects controller:
The effects controller will trigger different effects upon the buffered sample in a given order.

Buffer:
The buffer will be where music samples sit while they are operated on by various filters before being output to the speaker.

Filters:

1. Echo
Echo can be accomplished simply by repeating scaled, delayed copies of the audio output and adding this to the original signal.

2. Reverb
Reverb will be done by convolving the audio out with the impulse response of a resonant room.
If this proves to be too difficult or is ineffective, a backup solution is to simply repeat attenuated copies of the signal delayed by some small period of time (similar to echo, but with the echoes in much more rapid succession).

3. Pitch changes
A phase vocoder will be used to change the pitch of a piece of music. The phase vocoder will use a DTFT module and either an inverse DTFT module (if one exists) or a lot of multipliers (if an inverse DTFT module doesn't exist).

4. Speed changes

The 'speed' of the song, or duration, can be modified in a similar to the pitch using the Vocoder. Using a STFT, we calculate frequency changes in the Fourier domain on a different time basis, and then employ an inverse STFT to regain the time domain representation of the signal.

Outputs:

Visual Display:

The visual display will show the name of the current song, as well as the waveform, current effects applied, and a selection of other songs to play.

Scratch Pad:

As well as serving as a controller, the scratch pad will allow the user to see how quickly the song is being played back by the speed of rotation.