ECE 271 – Microcomputer Architecture and Applications Lecture 24

Vince Weaver

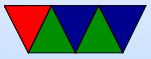
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

23 April 2019

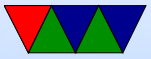
Announcements

- Read Chapter 12
- STM Rep visit next class



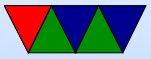
Lab#11 Stuff

- If things don't work, you have to debug!
- Remember to call your functions
- Check the values you are writing.
- Ask a friend to look over?
- Asking the prof should be a last resort!



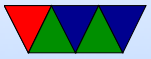
More Lab#11 Stuff

- Using fixed point. Why?
  If we didn't, lose the fractional part. Would end up rounding and the frequencies would be off.
- We use Decimal fixed point (milli-degrees). Could use binary, easier to see what's going on otherwise.



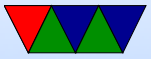
Lab#11 – Something Cool

- Making music
- Only about 5 extra lines plus some lookup tables
- See the textbook
- Musical notes, A4=440Hz. A4 it pitch#69
- $f = 440 \times 2^{(p-69)/12}$
- Octave has 8 notes, but really 12 notes if you could sharps/flats
- Have a countdown timer that is set for the length of the note and then counts down until it is done, then picks



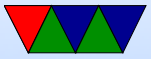
the next note.

- If 120BPM (bits per minute), then a standard note changes after 1/2 a 44100 cycle so count down from 22050*length.
- It will still sound electronic. To get instrument-like sounds you'd need to mess with the "envelope" (attack, sustain, decay, release)
- You can play multiple channels if you add different frequencies together, just make sure you divide so the value doesn't overflow 4096 (12-bits) or it will wrap around and sound weird.

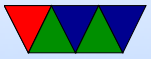


More advanced music

- One pin-GPIO
  - This is all older machines had
  - Can do a square wave of certain frequency. Hard on amplifiers + speakers (lots of higher harmonics)
  - Can use PWM. The speaker only has so fast a response, adds sort of like an average. So the average of the PWM output can approximate other waveforms.
- FM synthesis
  - So far have been doing AM (amplitude-modulation)

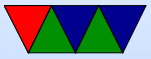


by modifying the amplitude of the sine wave
  - Can do FM (frequency modulation) where you rapidly change the frequency
  - 1980s synthesizers and DOS sound cards (OPL2 based Soundblaster)
- Sampled
  - Sample with ADC at some rate, maybe 44.1kHz
  - Store the 16-bit samples
  - Play back exact samples with DAC
  - Really good playback. What's the downside? lots of disk space. 44k*2bytes (16-bits)*2-channel (stereo) =



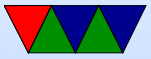
160k/second

- Can compress. Lossless (FLAC for example) or MP3/OGG etc (lossy)



Low-memory music playing

- For older systems you can use something called a "tracker" that looks sort of like a spreadsheet, and you put in a list of notes to play (plus length, and effects)
- They have patterns, which can repeat (such as refrains)
- Notes/instruments use lookup tables, so it can be fast

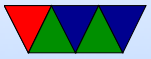


Apple II demo

- Making a low-frequency square wave from an interpreted language (Appleosft BASIC)

  ```
  FOR I=1 TO 1000: X=PEEK(-16336): NEXT I
  ```

  `PEEK(-16336)` is the same as doing a memory read of address `$C030`
- Show still-alive used "Electric Duet" by Paul Lutus, who went off-the grid in Oregon for a while writing Apple II sound software that could do two channels out the plain speaker, but needing most of the 1MHz CPU.



- Show still-alive using the Mockingboard, which is an add-in card with AY-3-8910 three-channel square wave, envelope effects, and noise.
  Programming this is a lot like what we do in class. There's a memory mapped timer and interrupt controller. Unlike our boards it's not a vectored interrupt, but only one, so you have to chain interrupt handlers.
- Looking at other classic games with sound: The Oregon Trail, Lemonade Stand, Lode Runner