Lab #6: Sampling Techniques: Interpolation, Decimation, Echo & Reverb

Purpose

Using the Codec interfaced to your DSP in the previous lab, we will now begin to experiment with sampling and playing out sound at different rates and buffer sizes. Specifically, we will experiment with different sampling rates to observe the effect of aliasing, oversampling and the effect of speeding up/slowing down sample playback. We will also look into changing the sampling rate of data using Linear Interpolation and Decimation. Finally, we will implement a simple real-time Linear, Time-Invariant systems to create reverb and echo audio effects. This should be a fun lab and everyone is allowed to choose their own sound input content.

Part I. Audio Mixing: SRAM Memory Tests in C

At this point you should have your LCD interfaced in 'C' via the DSP's onboard I2C Peripheral. You also should have your Codec functional using the McBSPB peripheral and the SPI-A peripheral.

1. First, create a circular buffer in your external serial SRAMs using bit masking in order to hold 16-bit audio samples. Assume: Fs = 48 KHz.

2. Next, use your Codec as a master as before and create an interrupt-driven routine that does the following:

Recording Function

If button1 (leftmost button) = pressed, // Start Recording
Sample the Codec and save the Codec A/D value into your buffer.
Light up LEDs =0001.

If button1 = pressed again, // Pause the Recording
Stop recording and turn off all LEDs.
Recording cannot happen again until button1 has been pressed again.

Playback Function

If button3 (rightmost) = pressed, // Playback, play out your current buffer contents.
Play out your buffer sample by sample to the Codec DAC and turn on LEDs = 0011.
Turn off all LEDs when playback has finished.

Mixing or No Mixing Function during Recording

If DIP Switches = 0, // No Mixing During Recording, LEDs=0001
If DIP Switches = 1, // Mix the sound in the buffer with your current sound being recorded.
Save this new sound to your buffer. If the current sound is longer than your original sampled sound, mix with zero sound as we have exceeded the original sampled buffer length. Turn on LEDs = 1001.

Zeroing Function

If button2 (center button) = pressed, // Zero your buffer
Light up all LEDs.

Note: We are looking for high quality (low noise) sound in/out the Codec. Best sound = the most points. We will experiment by layering more and more sound (DIP = 1 during record samples) and should only clear the sample buffer when PB2 is pressed.

Lab #6: Sampling Techniques: Interpolation, Decimation, Echo & Reverb

Part II. Basic Sampling at Different Rates (No Mixing is Required Here)

1. Once you have sound in/out of the digital domain, we can now begin to play with the sampling period to create and observe aliasing. To demonstrate this, set the Codec interrupt at 8 KHz (in/out) with a corresponding Nyquist frequency of 4 KHz . Sample and then play sound back out sound on one channel. To be sure that Fs = 8 KHz, you must also toggle a GPIO pin so that we can check the interrupt rate via your DAD LSA probe.

2. Once you are sure you are sampling and playing data at 8 KHz, input a 500 Hz sine wave with your DAD Function Generator. Look at both the A/D input and the D/A output with your scope. **How many samples are expected to be in one period of the sine wave? Is there an overall gain or attenuation in your system? Is there a phase shift in your output?**

3. Begin sweeping the input sine wave frequency in 1K Hz steps from 1 KHz Hz to 15 KHz. **Record what happens at the output. Look for and record changes in amplitude and phase for a given frequency input. Is there any aliasing? Yes/No, why does this occur or not occur?**

4. Repeat the above experiment with Fs = 32 KHz and 48 KHz. (See the attached Codec Sampling Rage table for more information. Note: **Use your binary DIP switch (rightmost three bits) to select a sampling rate so that only one ISR is required for all the different rates mentioned previously.**

5. Find a harmonically rich source (musical instrument, song, voice, etc.) and sample it at 8K Hz, 32 KHz and 48 KHz to listen to the differences when played out at the three different sampling rates.

Part III. Changing Sampling Rate, Interpolation, Decimation

1. Sent up Fs to be 48 KHz. Record 5 seconds of a complex ***musical*** waveform into your external SRAM. Play it back out at 8K Hz. **What happens when we decrease the playback rate?**

2. Sent up Fs to be 8 KHz. Record 5 seconds of a complex ***musical*** waveform into your external SRAM. Play it back out at 48 KHz. **What happens when we increase the playback rate?**

3. ***Up Sampling w/Interpolation***. Record a couple seconds of a complex musical waveform into your SRAM at 8 KHz. Use linear interpolation to create three new samples for every original sample to Interpolate by 4X i.e. If N = number of samples in your original sampled buffer, you will now have 4*N samples to play back out. Play back should be at 32 KHz to maintain the original note/musical frequencies. Also try playing back at the original 8 KHz sample rate. ***Note: If you don't have enough memory for 4*N samples, compute the new sample on the fly/real-time in your interrupt routine during playback. Use your binary DIP switch to select between the sampling playback rates so that you don't have to keep reloading code to the DSP!***

The main use of interpolation is to create a high definition video image from a low definition image. i.e. HD video from non HD sources on a modern TV.

4 . **Decimation & Aliasing.** Record several seconds of a waveform into SRAM at 32 KHz and this time remove samples such that the decimated rate is 4X smaller than the original rate. i.e. Keep 1 sample, remove 3 the next three samples, keep 1 sample, remove the next 3 samples, repeat until the end of the buffer. Play this waveform out at 32KHz, 48 KHz and 8 KHz rate. Test your Decimation with music and also with input Sines (one frequency = one test) at 500, 1000, 2000, 4000, 6000, 8000, 10,000 Hz. **What happens our input signal is near or above the Nyquist frequency of the 8 KHz playback?** ***Use your binary DIP switch to select between the sampling playback rates so that you don't have to keep reloading code to the DSP!***

Lab #6: Sampling Techniques: Interpolation, Decimation, Echo & Reverb

Part IV. Real-Time LTIs: Echo, Reverb (Fs = 48 KHz)

1. Create a real-time reverb effect where the delay time (increments of 10 milliseconds) can be controlled from your switches. The equation for a reverb system is as follows:

$$y[n] = (1-a)x[n] + ax[n-p]$$

Where a is the volume of your reverb (0-1) and p is delay time in samples.
p can be related to delay time in seconds through the following equation:

$$p = \lceil t * F_s \rceil$$

Where t is the desired delay in seconds and Fs is the sampling rate in Hz.

Given the above equation, what is the smallest delay we can achieve in terms of Fs?

2. Create a real-time echo effect where the delay time (increments of 0.25 seconds) are again controlled in real-time by your switches. The equation for an echo system is as follows:

$$y[n] = (1-a)x[n] + ay[n-p]$$

Where a is the volume of your reverb (0-1) and p is delay time in samples

Parts I - IV. Pre-Lab Requirements

Answer the bolded questions. Submit via Canvas to your TA. Also, submit all your C code.

Part I. In-Lab Requirements.

Demonstrate to your TA that you can sample and playback sound. Briefly explain your code.

Part II. In-Lab Requirements.

Quickly demonstrate to your TA the "sweep sine experiment" results. Is there aliasing in the output?

Part III. In-Lab Requirements.

Demonstrate to your TA the effect of changing the sampling rate, interpolation & decimation experimental results. In each case, you should be able to quickly explain your code also for maximum points.

Part IV. In-Lab Requirements.

Demonstrate to your TA echo and reverb.

Lab Point Break-down

In-Lab Quiz (Full Lab Time)	20%	
Part I - IV. Pre-lab Materials	10%	;2.5% per section
Functional Part I In-Lab Demo & Explanation	20%	
Part II In-Lab Demo & Explanation	10%	
Part III In-Lab Demo & Explanation	20%	
Functional Part IV In-Lab Demo & Explanation	20%	

Lab #6: Sampling Techniques: Interpolation, Decimation, Echo & Reverb

Codec Sampling Rate Table (MCLK = 12.288 MHz)

In normal mode, the following ADC and DAC sampling rates, depending on the MCLK frequency, are available:

MCLK = 12.288 MHz

SAMPLING RATE		FILTER TYPE	SAMPLING-RATE CONTROL SETTINGS				
ADC (kHz)	DAC (kHz)		SR3	SR2	SR1	SR0	BOSR
96	96	2	0	1	1	1	0
48	48	1	0	0	0	0	0
32	32	1	0	1	1	0	0
8	8	1	0	0	1	1	0
48	8	1	0	0	0	1	0
8	48	1	0	0	1	0	0

Is your MCLK is different? Check your Codec board! 12.288 MHz divided by 256 = 48 KHz. 12.288 MHz/384 = 32 KHz, 12.288 MHz/1536 = 8 KHz. Use your Codec's master clock (crystal) value to determine your actual sampling rates if it is not 12.288 MHz. i.e. Look up the crystal part number on line, ATS????B.