Microprocessor-Based Systems (E155)

D. Money Harris & K. Wang **Fall 2011**

Lab 5: Digital Audio

Requirement

Build a system to play music on a speaker. Use your PIC microcontroller, a LM386 audio amplifier, and an 8-ohm speaker. The PIC microcontroller should read a list of notes specifying the period and duration of each note. It should generate a sequence of square waves with the corresponding periods and durations. A period of 0 indicates a rest (silence for the given duration). A duration of 0 indicates the end of the song. Program the PIC in assembly language. Poll two timers to measure the period and duration. Test your system on the score of Fur Elise, which is provided.

Discussion

The score of Fur Elise is provided in lab5base.S on the course website. Place this code at the end of your lab5_xx.S file. The score is represented as a sequence of numbers specifying the period and duration of each note. Both are 16-bit values. The period is given in units of τ_1 = 1.6 μs. The duration is given in units of τ_0 = 51.2 μs. Set the peripheral clock to Fosc/8 = 5 MHz. You can configure Timer 1 to use a prescalar of 256 so that each count is 51.2 μs. Similarly, configure Timer 2 to use a prescalar of 4 so that each count is 0.8 μs, or half a period unit (convenient to set the number of units of time for a high output and for a low output). Both timers should use the 5 MHz peripheral clock as their source. Verify that your timing is correct by looking at the PIC output on an oscilloscope.

The PIC doesn't generate enough output current to play satisfactory music directly on the speaker, so use an LM386 audio amplifier between the PIC and the speaker. Do not connect the PIC directly to the speaker, as the current draw could damage the chip. The datasheet shows AC coupling from the amplifier to the speaker, but you can leave out the capacitors and resistors and produce an acceptable square wave. Volume control is optional. There are only a limited number of speakers available in the lab so ***please leave the speakers in the supply cabinet when you leave lab***. Do not leave them attached to your breadboard when you are done working.

As in Lab 4, use the modified crt0.S and elf32pic32mx.ld files since you are writing in assembly and do not need the extra startup code.

Extra Credit

Up to one point of extra credit can be earned if you compose and play a different tune. The following information may help as you compose your music.

Recall that the A above middle C (called A4) is 440 Hz and that an octive spans a factor of 2 in frequency. There are twelve notes in an octive spaced evenly on a geometric scale, so each is separated in frequency by a factor of $2^{(1/12)}$.

Note	**Frequency (Hz)**	**Period (in units of τ_1)**
A3	220	`0xB18`
A sharp / B flat	233.1	`0xA79`
B3	246.9	`0x9E2`
C3 (middle C)	261.6	`0x954`
C sharp / D flat	277.2	`0x8CE`
D3	293.7	`0x850`
D sharp / E flat	311.1	`0x7D8`
E3	329.6	`0x768`
F3	349.2	`0x6FD`
F sharp / G flat	370.0	`0x699`
G3	392.0	`0x63A`
G sharp / A flat	415.3	`0x5E0`
A4	440	`0x58C`
A sharp / B flat	466.2	`0x53C`
B4	493.9	`0x4F1`
C4	523.3	`0x4AA`
C sharp / D flat	554.4	`0x467`
D4	587.3	`0x428`
D sharp / E flat	622.2	`0x3EC`
E4	659.2	`0x3B4`
F4	698.4	`0x37E`
F sharp / G flat	740.9	`0x34C`
G4	784.0	`0x31D`
G sharp / A flat	830.6	`0x2F0`
A5	880	`0x2C6`

The duration depends on an arbitrary choice of tempo (speed at which the piece is played). If a whole note is chosen to be ½ second long, other notes follow accordingly:

Duration	**Seconds**	**Units of τ_0**
Sixteenth	0.03125	`0x0262`
Eighth	0.0625	`0x04C4`
Quarter	0.125	`0x0989`
Half	0.25	`0x1312`
Whole	0.5	`0x2625`