Microprocessor-Based Systems (E155)

D. Money Harris **Fall 2009**

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, provided.

Discussion

The score of Fur Elise is provided in lab5base.asm 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 $\tau_1 = 1.6\ \mu s$. The duration is given in units of $\tau_0 = 51.2\ \mu s$. Assume that the system clock runs at 20 MHz, so the instruction clock is 5 MHz. You can configure Timer 0 to use a prescalar of 256 so that each count is 51.2 μs. Similarly, configure Timer 1 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).

This is a great time to erase your PROM for the FPGA to ensure that FPGA outputs don't contend with PIC outputs in this or future labs. Verify that your timing is correct on an oscilloscope. Then try connecting the PIC output pin directly to the speaker. Listen to the results; the PIC doesn't generate enough output current to play satisfactory music. Hook up the LM386 audio amplifier between the PIC and speaker and see if the results improve. 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.

What to Turn In

When you are done, have your lab checked off by the instructor. You should thoroughly understand how it works and what would happen if any changes were made. Turn in your lab writeup including the following information:

- Your assembly language code
- Schematics of the breadboarded circuit
- How many hours did you spend on the lab? This will not count toward your grade.

For one point of extra credit, 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_2**
Sixteenth	0.03125	0x0262
Eighth	0.0625	0x04C4
Quarter	0.125	0x0989
Half	0.25	0x1312
Whole	0.5	0x2625