UNIVERSITY OF CALIFORNIA
College of Engineering
Electrical Engineering and Computer Sciences Department

145M Microcomputer Interfacing Lab

Final Exam Solutions May 13, 2006

1.1 Tri-State Buffer: Digital circuit with input and output lines, and a select line. When the select line is in one logic state, output = input. Otherwise the output is in a high-impedance state and neither drives nor loads any circuit connected to it.

1.2 Fourier Convolution Theorem: The Fourier transform of the convolution of two functions is the simple product of the Fourier transforms of the two functions

[2 points off for stating the Fourier Frequency Convolution Theorem]

1.3 Transition Voltages (A/D Converter): Specific input voltages at which the output switches between one output number and the next.

1.4 Infinite Impulse Response Digital Filter: A digital filter whose output depends on previous outputs

Note: The above definitions were taken directly from the textbook.

2.1 $r = a / b \qquad \partial r / \partial a = 1 / b \qquad \partial r / \partial b = -a / b^2$

$$\sigma_r^2 = \sigma_a^2 (\partial f / \partial a)^2 + \sigma_b^2 (\partial f / \partial b)^2 = \sigma_a^2 (1 / b^2) + \sigma_b^2 (a^2 / b^4)$$

$$\sigma_r^2 = 10^{-4} a^2 (1 / b^2) + 10^{-4} b^2 (a^2 / b^4) = 2 \times 10^{-4} (a^2 / b^2)$$

$$\sigma_r = 0.01414 (a / b) = 0.01414 r$$

3.1 A periodic waveform is the convolution of a time-limited waveform and a comb of delta functions. The Fourier transform of a periodic waveform is the simple product of the Fourier transform of the time-limited waveform and the Fourier transform of a comb of delta functions. The latter is a comb of delta functions in the frequency domain so the simple product is nonzero only at discrete frequencies. See Figure 5.26 of the textbook.

[7 points off if the periodic function is not described as the convolution of a time-limited function and a train of delta functions]

3.2 Sampling in the time domain is equivalent to a simple product of the waveform and a comb of delta functions in the time domain. The Fourier transform of the sampled data is the convolution of the Fourier transform of the wavefunction and the Fourier transform of the comb of delta functions. Since the latter is a comb of delta functions in the frequency domain, overlap will occur unless the original waveform is not frequency limited. This overlap causes aliasing and can be eliminated by increasing the sampling frequency or by bandwidth limiting the waveform so that the highest frequencies persent are sampled at least twice per cycle. See Figure 5.29 of the textbook.

[10 points off for only stating the Nyquist Theorem]

[5 points off for stating that aliasing is caused by frequency overlap but not stating that sampling is equivalent to multiplying by a train of delta function in the time domain]

3.3 Multiplying by a function that tapers to zero with zero slope at the edges of the sampling window reduces the discontinuities that produce unwanted frequency components. This is also equivalent to convolving the frequency spectrum of the untruncated waveform with the Fourier transform of the windowing function, which is designed to have only low frequency components.

3.4 Not using a windowing function is equivalent to multiplying the waveform with a rectangular time function which is equivalent to convolving the frequency spectrum of the untruncated waveform with the Fourier transform of the square wave, which has long range frequency components.

4.1 Required elements:

256 sensor circuits with 16-bit digital outputs

256 digital deglitching circuits, each consisting of 16 delays, 16 exclusive OR circuits, one 16-input OR circuit, and one16-bit transparent latch. (Similar to Figure 3.37 of the textbook)

256 tri-state buffers, each with 16 inputs and 16 outputs

one address decoder with input connected to 8 bits of output port and output used to select among the 256 tri-state buffers

32-bit I/O port, with 16 bits set to input and 8 bits set to output

computer

4.2 When one of the sensor circuit output changes the changed bits will generate different inputs to the corresponding exclusive OR circuits and these output will be combined in the OR circuit to put the transparent latch into hole mode during the glitch.

[10 points off if glitch rejection not described]

4.3 (1) Set 16 bits to input and 8 bits to output, set m = 0

(2) Wait until the seconds counter changes

(3) set n = 0

(4) select tri-state n to switch from high impedance to transparent mode

(5) read 16-bit input bus

(6) select the milliseconds timer and read the 16 bits

(7) select the minutes timer and read the 16 bits

(8) store the data and time into an array[n, m]

(9) set n to n + 1

(10) loop back to (4) until n = 256

(11) set m to m + 1 and loop back to (2)

5.1 $f_1 = 20$ kHz $G_1 = 0.99$

choose n = 8 $f_1/f_c = 0.784$ and $f_c = 25.5$ kHz

$G_2 = 0.001$ $f_2/f_c = 2.371$ $f_2 = 60.5$ kHz

$f_s > f_1 + f_2 = 80.5$ kHz

[5 points off for an incorrect or missing anti-aliasing filter, or using a sampling frequency of 40 kHz]

[2 points off for designing for 1% frequency accuracy at 1 Hz- in part 5.1 the frequency range was 20 to 20,000 Hz]

[2 points off for sampling at a frequency above 200,000 Hz- in part 5.1 the frequency range was 20 to 20,000 Hz]

5.2 Since we need to determine the frequency of harmonics as low as 20 Hz to an accuracy of 1% (0.2 Hz) we need to sample for at least 2.5 seconds. Even though the frequency bins are 0.4 Hz apart, we will be able to average to get 0.2 Hz accuracy.

Steps:

(1) Multiply waveform from part 5.1 by Hann window

(2) Take FFT

(3) Detect harmonic peaks, combine amplitudes in quadrature, average power to get frequencies of each harmonic, store frequency and combined amplitude

(4) Stop at 100^{th} harmonic or 20 kHz, whichever comes first.

[5 points off if harmonic amplitudes are not determined and stored]

[2 points if Hann window not used- otherwise large amplitude harmonics could interfere with small nearby harmonics]

5.3 Sampling at 80.5 kHz for 2.5 second, requires storing about 200,000 numbers.

5.4 The frequencies and amplitudes for the first 100 harmonics only requires storing 200 numbers.

5.5 (1) Computer detects keys pressed. (2) Computer looks up and combines frequencies and amplitudes stored in memory for keys pressed. (3) Computer does IFFT to generate the waveform that corresponds to notes played (4) computer sends for waveform to D/A converter (5) power amplifier (6) speakers. Re-compute when keys pressed changes. Necessary to perform a 250k IFFT in a few ms to reduce the delay.

[2 points off if no keyboard or means for the musician to play notes]

[2 points off if no lookup of harmonic frequencies and amplitudes corresponding to note played]

[2 points off if no inverse FFT to convert harmonic components to note waveform]

[2 points off if no D/A]

[2 points off if no amplifier and speaker]

Nlog n = 250k x 18 = 4 million operations. Can be done in 4 ms using modern chips that can do 10^9 floating point operations per second.

145M Final Exam Grades:

Problem	1	2	3	4	5	Total
Average	38.0	16.8	31.8	39.2	32.9	158.7
rms	3.0	4.2	10.4	7.3	7.2	19.3
Maximum	40	20	45	50	45	200

145M Numerical Grades:

	Short labs	Long labs	Lab Partic.	Midterm #1	Midterm #2	Final	**Total**
Average	77.5	394.4	100	93.8	85.1	158.7	**912.6**
rms	1.7	7.9	0	3.9	14.0	19.3	**30.9**
Maximum	105	400	100	100	100	200	**1005**

Note: The average of labs 1, 8, 10, 22, and 24a was 1 point per lab higher than the average of labs 2, 9, 21, 23, and 24b (omitting the lowest long lab grade). This was due to the nature of the labs and small differences in grading standards. One bonus point was added to the long lab total for each of labs 2, 9, 21, 23, and 24b.

145M Letter Grade Distribution

Letter Grade	Course Totals (1000 max)
A+	961
A	958
A–	921, 922
B+	902, 908, 912
B	891, 894
B-	
C+	857
C	
C–	
D+	
D	