

University of Pennsylvania
Department of Electrical and System Engineering
Digital Audio Basics

ESE150, Spring 2020

Final

Monday, May 11

- Exam is a 2 hour timed quiz on Canvas.
 - We recommend you draft your answers in a text file (and save often) so that you don't lose any data if Canvas or your browser were to crash.
- You may take the exam any time during the 24 hour period of Monday, May 11 EST. We will use Canvas scheduling to control availability.
- Calculators and computers allowed. (MATLAB, spreadsheets, etc.)
- Open book = Text and notes allowed.
- Internet is discouraged but not prohibited.
 - We won't give you extra time or other considerations if you have a period of Internet outage during the exam.
 - Our strong recommendation is that you do not plan on using it. (e.g., download everything you think you might need before starting the exam.)
 - On the off chance that an Internet outage prevents you from submitting the exam to Canvas, capture (e.g., print to PDF, screen capture, answers in text file) and submit your answers via email (andre@seas.upenn.edu).
- The following are prohibited:
 - Getting help from anyone else (in the class or not)
 - Discussing exam with anyone (other than course staff) during the 24 hour exam period
 - * even commiserating (e.g., problem X was hard)
 - * even if you believe someone has finished the exam
- Questions during exam (e.g., if you suspect a typo or error in the exam) should go to a private piazza post (preferred so that anyone on course staff can answer) or email.
 - Of course, we will try hard not to make errors, but history suggests there's a chance they remain.
 - We won't promise to be responsive throughout the entire 24 hour period.
 - We will take a poll of when you expect to take it so we can try to be available.
 - As necessary, state the assumptions you make along with your answers.
- Show work for partial credit consideration.
- Unless otherwise noted, answers to two significant figures are sufficient.
- Remember Code of Academic Integrity statement (included next page).

Code of Academic Integrity

Since the University is an academic community, its fundamental purpose is the pursuit of knowledge. Essential to the success of this educational mission is a commitment to the principles of academic integrity. Every member of the University community is responsible for upholding the highest standards of honesty at all times. Students, as members of the community, are also responsible for adhering to the principles and spirit of the following Code of Academic Integrity.*

Academic Dishonesty Definitions

Activities that have the effect or intention of interfering with education, pursuit of knowledge, or fair evaluation of a student's performance are prohibited. Examples of such activities include but are not limited to the following definitions:

A. Cheating Using or attempting to use unauthorized assistance, material, or study aids in examinations or other academic work or preventing, or attempting to prevent, another from using authorized assistance, material, or study aids. Example: using a cheat sheet in a quiz or exam, altering a graded exam and resubmitting it for a better grade, etc.

B. Plagiarism Using the ideas, data, or language of another without specific or proper acknowledgment. Example: copying another person's paper, article, or computer work and submitting it for an assignment, cloning someone else's ideas without attribution, failing to use quotation marks where appropriate, etc.

C. Fabrication Submitting contrived or altered information in any academic exercise. Example: making up data for an experiment, fudging data, citing nonexistent articles, contriving sources, etc.

D. Multiple Submissions Multiple submissions: submitting, without prior permission, any work submitted to fulfill another academic requirement.

E. Misrepresentation of academic records Misrepresentation of academic records: misrepresenting or tampering with or attempting to tamper with any portion of a student's transcripts or academic record, either before or after coming to the University of Pennsylvania. Example: forging a change of grade slip, tampering with computer records, falsifying academic information on ones resume, etc.

F. Facilitating Academic Dishonesty Knowingly helping or attempting to help another violate any provision of the Code. Example: working together on a take-home exam, etc.

G. Unfair Advantage Attempting to gain unauthorized advantage over fellow students in an academic exercise. Example: gaining or providing unauthorized access to examination materials, obstructing or interfering with another student's efforts in an academic exercise, lying about a need for an extension for an exam or paper, continuing to write even when time is up during an exam, destroying or keeping library materials for one's own use., etc.

* If a student is unsure whether his action(s) constitute a violation of the Code of Academic Integrity, then it is that student's responsibility to consult with the instructor to clarify any ambiguities.

We've all been using Zoom and Google Meetings for the last half of the term. Like a mobile phone, Zoom uses digital audio for sound and digital video for pictures. Zoom adds the ability to support sound and video from a group of people. It builds on the same technologies we've been understanding during the term. In this final, we'll look at how it puts together the building blocks we've already seen and what optimization issues may come up in multi-person, online meetings.

1. **Multi-source Audio:** One key difference is that Zoom combines live audio from multiple sources. An MP3 player only produces one (already combined) audio source. A cell phone is only sending one audio stream each way.
 - (a) Using what you learned in the course about human psychoacoustics, explain why the output stream can be the same size as any of the input streams and provide the same quality. (Quiz 2, 5pts)
 - (b) Using what you know about MP3s and human psychoacoustics, briefly describe how you would take multiple 128Kb/s MP3 streams and combine them into a **single** 128Kb/s output MP3 stream to send to a single participant. (Quiz 3, 5pts)

2. **Unique Audioscapes** Continuing on with multi-source audio—in practice, we might like each participant to get a unique combination of input signals. In particular, at least, we don't want to include the participant's own sound in the combined output in order to prevent echos. We might like to customize it further, perhaps allowing each participant to include a unique subset of sounds. That raises questions of where we should do the combining. Continue to assume 128Kb/s MP3 audio streams. Assume adding two streams requires 3 Million instructions per second.

- What are the worst-case bandwidth and compute requirements for a 15 person meeting in the following cases:

(a) Server receives all 15 input streams from participants, creates 15 unique combinations, and produces 15 output streams, one for each participant. (Quiz 4, 6pts)

Server	input bandwidth	
	output bandwidth	
	computational cycles/s	
Participant Computer	input bandwidth	
	output bandwidth	
	computational cycles/s	

(b) Server receives all 15 input streams from participants and sends 14 streams (omit self stream) to each participant. Each participant's computer performs the combination locally. (Quiz 5, 6pts)

Server	input bandwidth	
	output bandwidth	
	computational cycles/s	
Participant Computer	input bandwidth	
	output bandwidth	
	computational cycles/s	

3. **Multi-source Video** Zoom provides both video and audio. As we've seen from time-to-time, video compression exploits similar concepts. Furthermore, each individual is able to configure their videoscape as well (configuration and size of individual video streams, including shared desktop stream). Assume for simplicity that each video stream (cameras from participants, desktop view from presenter, video output to be displayed for each participant) contains 1024×1024 32b pixels.
- (a) The limitation here is the computer display window (or maybe our bandwidth budget). Describe how this limit constraining the output video for a participant is an analog to limits on human sound perception. (Quiz 6, 5pts)
 - (b) Considering only single image frames at a time from the video streams, in order to pack the composite image into a single image (e.g. pack 4 1024×1024 input images into an array of 4×4 images on a 1024×1024 composite image), some form of compression is required. Identify the form of compression (or explain how to perform the compression) and classify it as lossy or lossless. (Quiz 7, 5 pts)

4. **User Interface for Multi-source Video and Audio** As noted, above (and demonstrated in Zoom), we may want to allow each participant to customize the audio and video streams they see and hear. To contain the scope, let's focus the audioscape and allow:

- each participant to select which of the participants to hear (listen-to)
- each participant can select which participants they will allow to hear them
- each participant can alter their choices during a session (e.g., choosing to whom they will talk, choosing to whom to listen)
- participants may join and leave the session

Assume sound combining for a single participant is handled in a process that receives all the input streams. This could be a process on the server or on the participants local machine.






- (a) What configuration is needed in each such participant process to control the combination process? (Quiz 8, 2pts)
- (b) Score each of the following UIs according to the evaluation metrics given (similar to what we used in Lab 12):
 - User Time
 - Cognitive Load
 - Error Prone
 - Self Describing

Rate each 1–5 (1-bad (high time, load, error rate, not self-describing); 5-good (low time, load, error rate, completely self-describing)). Give a short comment on what makes good/bad.

- i. Conference has a text control console. As participants join and leave, a line is printed on the console. Each user has two commands, “listen” and “allow” that each take a binary string, where each bit in the string represents a participant that they either want to listen to (listen command) or allow to listen to them (allow command). Ordering of bits in string is alphabetical based on last name of current participants. By default new participants are classified as neither allow or listen. (Quiz 9, 3pts)

```
Notice: Conference started.
Notice: Welcome Mike Jones.
Notice: Participant Sally Watkins joined.
Notice: Participant John Smith joined.
Notice: Participant Mary Baker joined.
Action> listen 1011
Action> speak 0011
Notice: Participant Ted Abrams joined.
Action> listen 11011
```

- ii. Each user has a set of pictures/names for other participants and two associated buttons – one to allow the participant to hear and one to listen to the participant. Each picture/name displays the current allow and listen state. Participants that have joined are flagged until the user makes a choice on them. (Quiz 10, 3pts)

Who	Listen To		Allow Hear		
 Ted Abrams	Turn On	Turn Off	Turn On	Turn Off	Just Joined
 Mary Baker	ON	Turn Off	Turn On	OFF	
 Mike Jones	ON	Turn Off	Turn On	OFF	
 John Smith	ON	Turn Off	Turn On	OFF	
 Sally Watkins	ON	Turn Off	Turn On	OFF	

- iii. Each user has a list of allowed and listen-to users. The user can remove a participant from one of the lists with a remove button for the user. A user can add someone to the list by a scroll selection, that also allows the user to type the prefix of the participants last time (or last name, first name) to jump forward in the scroll. When new user's join, a window pops up for 15s to announce their joining. There are also special buttons to add or remove all users to each of the lists. (Quiz 11, 3pts)



Listen

Mary Baker Remove

Mike Jones Remove

Mary Baker

Add

Add All

Remove All

Allow

Mary Baker Remove

Mike Jones

Sally Watkins

Add

Add All

Remove All

- (c) Which user interface is likely to be easiest for the users?
 - i. for a small group of 5 participants (Quiz 12, 3pts)
 - ii. for a large group of 1000+ participants, such as a large lecture course (Quiz 12, 3pts)
- (d) Based on your selected UI for c.i (Quiz 12), how do you extract the information that each participant output stream process needs (as identified in Problem 4a)? (Quiz 14, 2pts)
- (e) Suggest at least one way to improve your chosen UI or describe a UI that is likely superior. Explain why, include relation to evaluation metrics. (Quiz 15, 3pts)

5. Processor and Network handling multiple sessions

One processor, with a single network connection, can potentially handle multiple, multi-participant (e.g., Zoom) sessions. Assume a system with:

- Single processor running 2 billion instructions per second (2×10^9 instructions/s).
 - Single network port that can handle 1 billion bits per second input and 1 billion bits per second output.
 - Each Audio+Video stream (input or output) requires 1MB/s (10^6 Bytes/second)
 - A meeting session with P participants will require $3 \times 10^6 P^2$ instructions per second. (This assumes server is generating the per-participant output streams.)
- (a) Start by considering a processor handling a single, multi-participant session. How could the computer know which packets come from which participant input audio+video stream? (Quiz 16, 3pts)
- (b) Moving to multiple, multi-participant sessions (meetings), how can the processor keep track of multiple sessions? (Quiz 17, 3pts)
- (c) How many total input streams can the single network port handle? (Quiz 18, 3pts)
- (d) Assuming a set of 10 person meetings, how many meetings can the processor computation support? (Quiz 19, 3pts)

6. **Processing requirements for “wake” words** Smart audio digital assistants (e.g., Alexa, Siri) often process voice recognition on network servers. However, to save bandwidth and computation they will typically not send audio data for remote processing unless they first hear a “wake” word that tells them to pay attention to your commands. The “wake” words are processed locally by simple, low-energy processors that are always processing sounds. Let’s understand how these wake-computations may be simpler than arbitrary sound processing.

First, we know the “wake” words can be perceived by humans. That suggest certain limitations on what it will take to recognize them. Furthermore, they are understood widely by almost all humans, so they will not require particularly high frequencies. Let’s assume they can be recognized with no frequency components above 4KHz.

Assume we use a dot-product, Fourier Transform as in lecture and Lab 8 to extract frequencies (there are more efficient ways, but they are beyond what we covered in class, so we will stick with this simplification). As a result, for a sample window of N samples, it will require $2N$ multiplications and N additions for each frequency component computed. Assume it takes an addition $7N$ instructions for control and loop handling per sample, for a total of $10N$ instructions per frequency component.

- (a) Assuming the assistant must handle MP3-quality audio for some tasks, at what rate must it be able to sample audio input? (Quiz 20, 3pts)
- (b) During idle periods while the assistant is waiting for a “wake” word, at what rate must it sample in order to recognize “wake” words? (Quiz 21, 3pts)?
- (c) Assuming the Fourier Transform on 25 ms windows is the dominant computation required to recognize wake words, how many instructions per second must the local processor on the digital assistant support (Quiz 22, 6pts)?
(Hint: how many samples in the 25 ms window? How many frequencies can you extract from that window?)
- (d) Continuing to assume 25 ms windows, how much more processing would be required if you performed the processing on the full MP3 frequency range? [State assumptions as necessary.] (Quiz 23, 4pts)?

7. **Logic to Count Streams to Mix** Show how to compute the number of streams you will need to mix to generate the output stream for a single participant from a 4-input allow (to hear) and is-listening vector as developed in Problem 4. The inputs will be the vectors relevant to a single output.

- listen-to[i] = This participant is listening to participant i .
- allow[i] = Participant i is allowing this participant to hear them.

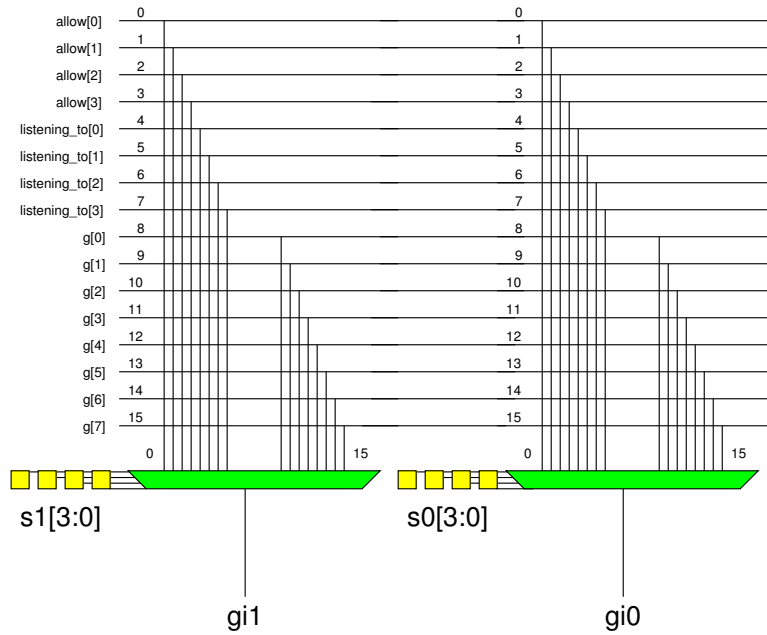
(a) Design logic to compute the count using primitive logic gates (1-input NOT gate, 2- or 3-input AND, NAND, or OR gates) and 3-input, 2-output Full Adder (FA) gates. (Quiz 24, 5pts)?

(Hint: given the 4-input bit-vectors, how many bits do you need to hold the result?)

You may write as functional equations, e.g.

```
t1=AND2(a,b);
t2=AND2(b,c);
t3=AND2(a,b);
carry=OR3(t1,t2,t3);
```

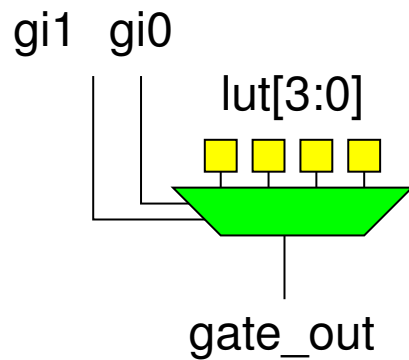
(b) How should you program the interconnect multiplexers shown to select gate inputs (gi1, gi0) as allow[0] and listen-to[0] ? (Quiz 25, 8pts)?



Provide the binary values that need to be stored in s1[3:0] and s0[3:0] to perform the intended selection:

s1[3:0]	
s0[3:0]	

- (c) Program this mux-based programmable gate to perform the AND function on the two inputs gi0 and gi1 (where gi0 and gi1 may be the outputs produced from Question 7b (25)). (Quiz 26, 4pts)?



Provide the binary values that need to be stored in lut[3:0] so that it performs the intended AND function:

Human auditory critical bands:

Band Number	Low	High
1	20	100
2	100	200
3	200	300
4	300	400
5	400	510
6	510	630
7	630	720
8	720	920
9	920	1080
10	1080	1370
11	1270	1480
12	1480	1720
13	1720	2000
14	2000	2320
15	2320	2700
16	2700	3150
17	3150	3700
18	3700	4400
19	4400	5300
20	5300	6400
21	6400	7700
22	7700	9500
23	9500	12000
24	12000	15500