

Correspondence Project

Synopsis:

Analyze professional writing from an industry you'd like to work in, create your own professional writing (3 texts) based on the examples, and display your work—along with a short professional bio—on a website or blog.

Details:

Conduct primary, secondary, and tertiary research about the professional writing used by a specific company or industry you hope to work for in the future. Consider these analytical questions: What genres do they use to communicate? Who are their readers? What can you find out about the way they go about creating letters, emails, and memos? In other words, how do they say what they say to their intended audience? Analyze those examples, then use them to create your own texts. Create a blog or website and put your writing on display. On the blog or website, create a short (30-50 word) bio that conveys who you are as a professional.

Deadlines:

PART 1
RESEARCH

What is the industry/company name?

Computer Hardware Industry

What are their values and goals?

The goal of workplace writing in the computer hardware industry can range from business memos and emails for internal coordination, to technical documents for engineers. In each case the writing has an intended reader and stylistic approach that can differ between companies.

Professional Writing Genre #1 - (due by class time Mon, Jan. 26)

Genre Name **Technical Documentation**

General Analysis: What are the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style?

Hardware companies produce an extensive amount of technical documents such as IC datasheets, user guides, design outlines and more. These documents are highly specialized, formal, and detail oriented, as the intended audience is engineers, and developers. They assume the reader has adequate background knowledge, using industry specific terminology, and abbreviations. Since technical documentation is very detailed, long, and extensive they are usually written by entire teams of engineers and developers. The purpose of these documents is to teach others who are not directly involved in the development of a certain hardware how to use or understand the workings of a device.

Industry standard practice is highly structured documentation with sections, checklists, tables, diagrams, and step by step instructions. It avoids ambiguous language, instead opting for precise language with a single interpretation. Intentional use of visual diagrams that help readers understand the material. Heavy use of industry specific terminology such as GPIO, power rails, PCB, and DAC.

Find a Primary Source: Copy and paste an example of the genre. Respond to the analytical questions above. What are the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style? Speak generally about the genre, but also explain a bit about the primary source itself (i.e. the author, their specific purpose, where they work, etc.)

1.3.3 Chip Power-up and Reset Timing

ESP32's CHIP_PU pin can enable the chip when it is high and reset the chip when it is low.

When ESP32 uses a 3.3 V system power supply, the power rails need some time to stabilize before CHIP_PU is pulled up and the chip is enabled. Therefore, CHIP_PU needs to be asserted high after the 3.3 V rails have been brought up.

To reset the chip, keep the reset voltage V_{IL_nRST} in the range of (NA ~ 0.6) V. To avoid reboots caused by external interferences, make the CHIP_PU trace as short as possible.

Figure *ESP32 Power-up and Reset Timing* shows the power-up and reset timing of ESP32.

Table *Description of Timing Parameters for Power-up and Reset* provides the specific timing requirements.

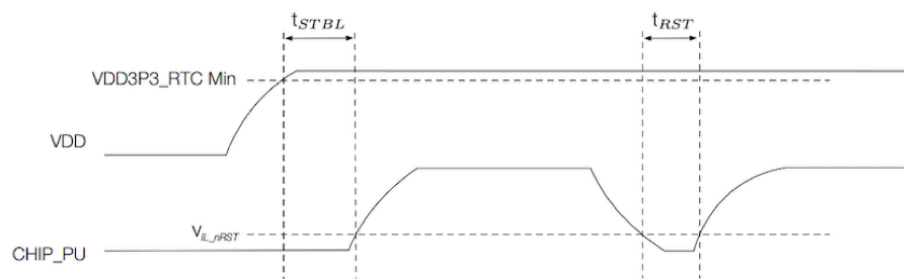


Fig. 4: ESP32 Power-up and Reset Timing

Table 2: Description of Timing Parameters for Power-up and Reset

Parameter	Description	Minimum (μs)
t_{STBL}	Time reserved for the power rails to stabilize before the CHIP_PU pin is pulled high to activate the chip	50
t_{RST}	Time reserved for CHIP_PU to stay below V_{IL_nRST} to reset the chip	50

Attention:

- CHIP_PU must not be left floating.
- To ensure the correct power-up and reset timing, it is advised to add an RC delay circuit at the CHIP_PU pin. The recommended setting for the RC delay circuit is usually $R = 10\text{ k}\Omega$ and $C = 1\text{ }\mu\text{F}$. However, specific parameters should be adjusted based on the characteristics of the actual power supply and the power-up and reset timing of the chip.
- If the user application has one of the following scenarios:
 - Slow power rise or fall, such as during battery charging.
 - Frequent power on/off operations.
 - Unstable power supply, such as in photovoltaic power generation.

Then, the RC circuit itself may not meet the timing requirements, resulting in the chip being unable to boot correctly. In this case, additional designs need to be added, such as:

- Adding an external reset chip or a watchdog chip, typically with a threshold of around 3.0 V.
- Implementing reset functionality through a button or the main controller.

The following is a snippet from the “ESP32 Hardware Design Guidelines” published by Espressif Systems, a semiconductor company that designs the ESP32 system on chip (SoC). These guidelines are intended for hardware and application engineers to correctly integrate the ESP32 chip into their own commercial hardware products. It outlines best practices, constraints, and electrical requirements to meet the minimum performance standard specified by Espressif.

The snippet is specifically section 1.3.3 of the document which highlights some of the characteristics of technical documentation. Technical documentation is highly structured with sections, checklist, tables, diagrams, and explicit instructions. Authors often make use of visual references to help engineers understand the material better. The diagram (Fig 4) is an example of this conveying the same information as the table below it. However, it was created under the assumption that the reader is familiar with waveform analysis.

When writing technical documentation it's also important to acknowledge that readers will rarely read linearly. Readers will often jump between sections to find the specific information they need.

This has some added implication as certain sections may need to reference information from an earlier portion. Since readers don't usually read in a linear fashion, authors must make heavy use of citations to point the reader to where they could find the information: “Figure [ESP32 Power-up and Reset Timing](#) shows the power-up and reset timing of ESP32.” In this case referencing information within the same section. This also has the added benefit of avoiding the repetition of information between sections.

The actual writing of technical documentation makes heavy use of bullet points to convey information. While they do include short paragraph style writing, they typically try to avoid long paragraph blocks. This stems from authors expecting engineers to scan through a section, rather than reading it in its entirety. Technical documents also are highly detailed and benefit from having the information broken up into concise independent bullet points. They are less susceptible to the ambiguity of reading a string of sentences and piecing them together.

Secondary Source (for example, a person or industry expert talking ABOUT what the genre is & how people in the industry use it): Who is the person? What does the person say? Respond to the analytical questions above based on what the person says. Summarize their perspective here and include a link to the source.

<https://medium.com/slalom-build/how-to-write-documentation-that-people-will-actually-read-and-use-b26791fc1429>

The author of this article is Hal Deranek who works for Slalom, a business and technology consulting firm in Washington DC. He outlines a few critical criteria with the first being to write for your audience.

When writing documentation there must be a balance between assuming your audience is experienced or completely new. The way you get around this is by “writ[ing] additional documentation” and leveraging the highly structured nature of technical documents. Splitting information into discrete sections of information allows users to skip sections to reach the information they need. Good documentation would hand hold new readers through the technical information while allowing experienced users to go at their own pace.

The next step is to have it peer reviewed by someone “who can validate your documentation” before it is released to the public. This allows for constructive criticism that could fill in some of the gaps of necessary and important information for those who are new.

Tertiary Source (for example, a reference guide that explains the genre, including an encyclopedia, Credo, a credible industry database, Claude, Chat GPT and/or Perplexity. You can also use these platforms to find primary and secondary sources that are credible, authentic, and authoritative.) What did you discover? What is the platform and are they credible? What were the prompt/s you used to find this information? And answer the analytical questions above: what did you learn about the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style?

For the tertiary source, I used ChatGPT’s deep research feature to get a detailed overview of technical documentation as a workplace writing genre. ChatGPT functions as a tertiary source because it makes interpretations of existing first and secondary sources similar to platforms like wikipedia.

ChatGPT draws from a wide range of sources that are publicly available, cited, and credible from major hardware companies such as AMD and NVIDIA. While the tool has the tendency to occasionally hallucinate information or draw the wrong conclusions, it generally produces good and accurate summaries when used with caution. It becomes a powerful tool for building a general understanding of a subject rather than being a factual citation.

The prompt I used provided the genre, industry, and information that I was looking for. This includes purpose, audience, values, writing conventions and more.

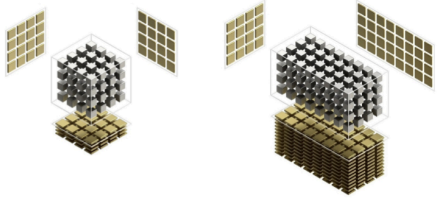
What I discovered lined up with my preemptive understanding of the subject highlighting that“ [t]hese documents are **highly formal and detail-rich**, intended for an audience of engineers, developers, and technical stakeholders. They assume a **specialized readership** with background knowledge, using industry-specific terminology and precise language.”

Third-Generation Tensor Cores in GA10x GPUs

Table 6. Comparison of NVIDIA Turing vs Ampere Architecture Tensor Core

	TU102 SM (RTX 2080 Super)	GA100 SM (A100)	GA10x SM (RTX 3080)
GPU Architecture	NVIDIA Turing	NVIDIA Ampere	NVIDIA Ampere
Tensor Cores per SM	8	4	4
FP16 FMA operations per Tensor Core	64	Dense: 256 Sparse: 512	Dense: 128 Sparse: 256
Total FP16 FMA operations per SM	512	Dense: 1024 Sparse: 2048	Dense: 512 Sparse: 1024

Below is a visual depiction of a single Ampere architecture Tensor Core and a single Turing architecture Tensor Core performing matrix math calculations and showing comparative throughputs of RTX 3080 vs RTX 2080 Super as represented by the stacks of completed operations performed over the same amount of time.



TURING ARCHITECTURE TENSOR CORE
(GeForce RTX 2080 Super)

AMPERE ARCHITECTURE TENSOR CORE with Sparsity
(GeForce RTX 3080)

With Sparsity enabled, the GeForce RTX 3080 delivers 2.7X higher peak FP16 Tensor Core operation throughput compared to a GeForce RTX 2080 Super with dense Tensor Core operations.

Figure 12. Ampere Architecture Tensor Core vs Turing Tensor Core

With a total of 68 SMs per GPU and a boost clock of 1710 MHz, the GeForce RTX 3080 GPU delivers 119 peak FP16 Tensor TFLOPS with FP16 accumulate, and with Sparsity enabled, 238 peak FP16 Tensor TFLOPS with FP16 accumulate. RTX 3080 delivers 238 peak INT8 Tensor TOPS and 476 peak INT4 Tensor TOPS, and double those rates with Sparsity enabled.

NVIDIA Ampere GA102 GPU Architecture

25

ChatGPT used NVIDIA’s GPU architecture whitepaper as an example, which conveyed complex information in a highly structured, objective, and detail oriented manner. The document clearly defines their audience, and purpose while using diagrams, tables, and clear section headings that allow readers to skim and find specific information easily.

The paper itself a high level overview of the design and architecture of NVIDIA GPU more coincided with a briefing or presentation paper.

Based on the overview it gave I drew an understanding that technical documentation is formal, technically detailed, structurable, non linear, and skimmable.

<https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf#:~:text=The%20family%20of%20new%20NVIDIA%C2%AE,GPU%20Tensor%20Core%20Architecture%20Whitepaper>

NOW IT'S YOUR TURN

How would you use this genre? Who is your audience? What would you say? What is your purpose?

As a hardware developer my work is only as useful as my ability to clearly explain it to others. I would use the genre of technical documentation to describe the architecture and design of the circuit I am creating. This documentation would serve as a reference for implementation details, constraints, and design decisions.

Since this is a highly technical document the intended audience would be other hardware engineers who may need to understand, modify, or build upon my design. Because the intended audience is technically knowledgeable, I would be able to make assumptions like familiarity with common terminology, and electronic concepts.

The overall purpose of this documentation would be to communicate how the system works, how it can be implemented or extended, the limitations, and trouble shooting strategies. Good documentation wouldn't have gaps in necessary or important knowledge as it will be used in a number of ways. Writing in a structured, formal, and detailed manner leaves little room for ambiguity.

The document will be organized into discrete sections covering each component of the circuit. Information will be conveyed primarily through short, concise paragraphs, bullet points, tables, and diagrams. Written in a form that accounts for skimming and non linear reading.

Professional Writing Genre #2, (due by classtime Wed, Jan. 28):

Genre Name: Change Logs & Revision Notes

General Analysis: What are the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style?

A change log or revision note is a common type of chronological, structured list of updates made to a project. The primary authors are developers, typically the person who made the changes. Change logs are intended to make it easier for users or developers to see the notable changes made between different versions of a project. They are often short, primarily use bullet points, and concise allowing readers to scan through it quickly. Change logs are primarily for developers or contributors but secondary audiences can also be anyone.

Find a Primary Source: Copy and paste an example of the genre (or include a link to it). Respond to the analytical questions above. What are the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style? Speak generally about the genre, but also explain a bit about the primary source itself (i.e. the author, their specific purpose, where they work, etc.)

Rolling Release Changes

These changes have not yet been incorporated into a stable release, but if you are on the latest version of the rolling release channel, you can take advantage of these new features and fixes.

🔗 New Features/Changes

Code Quality/Technical Changes

- If you are mounting a filesystem that has folders or paths that are not accessible by AzuraCast, the software will skip these directories instead of them preventing regular operation.

Bug Fixes

- Fixed a bug preventing setup from completing correctly.
- Fixed a bug where UTC offsets that weren't a full hour (i.e. +6:30 UTC) were not displayed correctly in dropdowns.

AzuraCast 0.23.2 (Dec 11, 2025)

New Features/Changes

- The embeddable player widget builder has been greatly expanded and now supports custom colors, several preset layouts, enabling or disabling components in the player, and saving/loading custom templates.
- Administrators can now create login tokens for any account that function as "magic links" that complete the authentication process in one step, as opposed to only being able to create password reset links. Both types of links can be created from the Administration panel.

Code Quality/Technical Changes

- If already installed, instances of Rocket Streaming Audio Server (RSAS) or Shoutcast can now be uninstalled from the web interface.
- Notifications on the dashboard homepage have unique IDs, meaning they can be individually targeted by CSS for styling or other customization purposes.

Bug Fixes

- Several bugs causing the Media Manager's cache to become out-of-date from the actual contents of the filesystem have been resolved, so navigating around the station filesystem should accurately represent its contents in many more cases.
- Fixed a bug where live streamers/DJs with a disconnect delay would never be reactivated.

<https://github.com/AzuraCast/AzuraCast/blob/main/CHANGELOG.md>

The following is a snippet from the AzuraCast change log hosted on github. AzuraCast is a free and open source, self hosted web radio station software used by people who want to broadcast their music across the internet. This change log, as the name implies, outlines the changes made to the newest version of the application. It uses a structured list where each change is split into its various types like new features, bug fixes, security updates, deprecated(no longer supported features), or general changes to existing features. The change log uses no paragraph style writing primarily using bullet points instead. Each bullet point is completely independent from each other highlighting a unique change made in the project. The bullet points themselves are short, and concise leaving little room for ambiguity. The software itself is targeted towards people who have

the technical skills to set up AzuraCast. As such, they make use of technical terminology. However, authors don't expect readers to understand every change made to the project. Similar to documentation readers often look for information that they need rather than reading it in its entirety. Since this is an open source project these change logs have many authors from the community of contributors that manage it.

Secondary Source (for example, a person or industry expert talking ABOUT what the genre is & how people in the industry use it): Who is the person? What does the person say? Respond to the analytical questions above based on what the person says.

Summarize their perspective here and include a link to the source. Include a link to the source.

<https://keepachangelog.com/en/1.1.0/>

Change logs aren't standardized across the industry however good change logs often follow core principles outlined in this article. The article is written and maintained by Oliver Lacan whose career is centered around technology education. They pioneered the shields metadata badge project with over 1 million GitHub repositories using them in their descriptions. They have had a profound impact on the way developers manage descriptions, documentation, and change logs. Similar to the core principles of workplace writing, the focal point is on the reader: "change logs are for humans, not machines." This article outlines common strategies to reduce the burden of writing change by simply keeping an unreleased section of changes, serving two purposes. One so that people can see upcoming plans for a new project release and two when its time to release a new version you can simply move the written changes into the change log. The key is to keep dates consistent, acknowledging deprecations, and removing any unnecessary comments or information. The goal for change logs is not to be detail oriented but a quick high level overview of changes.

Tertiary Source (for example, a reference guide that explains the genre, including an encyclopedia, Credo, a credible industry database, Claude, Chat GPT and/or Perplexity. You can also use these platforms to find primary and secondary sources that are credible, authentic, and authoritative.) What did you discover? What is the platform and are they credible? What were the prompt/s you used to find this information? And answer the analytical questions above: what did you learn about the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style? Include a link to what it generated & your prompt/s.

<https://en.wikipedia.org/wiki/Changelog>

Wikipedia is a widely known tertiary source that summarizes information from primary and secondary sources. It's often useful for gaining an overview of a subject or sourcing more credible sources. Due to wikipedia being an interpreted summary of sources and

its open wiki the quality and accuracy of a wikipedia article can vary. The author could be anyone regardless of their background. However, they do have standards and often contain dozens of primary and secondary sources. Through wikipedia is how I discovered a secondary source which I had also seen referenced in many developer forums.

The wikipedia article itself covers the basics of what a change log is, as “primarily a change made to software”. They acknowledge the broad readership for change logs where they can be written in a more technical manner which is harder for the average reader to understand or simplify. They cover common naming conventions for changing log files, and typical formats that are commonly used.

NOW IT’S YOUR TURN

How would you use this genre? Who is your audience? What would you say? What is your purpose?

Since I will need to work on a team its important that any changes I make towards a project are communicated to the rest of my team. Since hardware projects often have many dependencies, small changes could impact the development of another person's work. My audience will primarily be for my development team, and if the change logs are made public they would additionally be for the general user. The change log would use primarily bullet points and be short and concise.

Professional Writing Genre #3 (due by class time on Mon, Feb. 2)

Genre Name Design Docs

General Analysis: What are the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style?

Design docs are commonplace both in software and hardware engineering. They are typically informal pieces of documentation that give a high level overview of the implementation strategy and design decisions that place an emphasis on the trade offs considered in the design process. They are similar to technical documentation with key distinctions, for one they are typically made prior to the start of the project. As a result they use less deep technical terminology, simply identifying the plan, issues being tackled, and ensuring consensus around a design. They are often made within organizations as a pitch to get approved. The audience is both members of the project team, and those with stakes within the project.

of a design before a project is begun. They are similar to technical documentation but

Find a Primary Source: Copy and paste an example of the genre (or include a link to it). Respond to the analytical questions above. What are the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style? Speak generally about the genre, but also explain a bit about the primary source itself (i.e. the author, their specific purpose, where they work, etc.)

<https://sdmay25-19.sd.ece.iastate.edu/sdmay25-19-DD.pdf>

4.1.3 Technical Complexity

Our design presents a variety of technical challenges:

1. Since we are using open source software, none of our team members are familiar with the tools we are using. This provides a barrier of entry to getting started with our actual design.
2. ReRAM is an emerging technology, and there are very few opportunities available for fabrication of ReRAM chips. Also, there is little information about ReRAM usage for compute-in-memory applications, so we are exploring a new frontier.
3. We are including four different ReRAM architectures in our design, which of course increases the complexity of our design. However, this will likely help us down the line: if we find that one of our architectures doesn't work as intended, there are still three others to test.
4. We are also integrating a number of components acting as the peripheral circuitry of the design. This includes S&H circuits, TIAs, DACs, and ADCs. While our team doesn't have direct experience with each of these components, they are very well studied in academia and in industry, so finding documentation about them shouldn't be a struggle.

4.2 Design Exploration

4.2.1 Design Decisions

1. New ReRAM Architectures

The first design decision we had to make was what kinds of crossbars we would use in our unique ReRAM architectures. For our first crossbar, we decided to do a similar 1T1R matrix as the two senior design teams that came before us, but ours would be different by parallelizing the bitlines and sourcelines in our layout. The previous two teams had their sourcelines parallel to their wordlines. We decided to change things up with our second crossbar design and explore a true ReRAM crossbar, which uses no transistors and only uses a matrix on ReRAM cells. We felt that this would offer a unique solution that maximizes density within our crossbar. The operations on this crossbar are more difficult and noise-prone, but the potential for high-density ReRAM computation could be worth the risk.

2. ADC Resolution

We had a few options for ADC resolution in our project. We could use the 1-bit ADC from team sddc23-08, the 4-bit ADC from team sddc24-13, or design our own new ADC at a different resolution. We were partly swayed in our decision due to verification difficulties with the 4-bit ADC (seemingly due to differences in our tool installations), but we ended up using the 1-bit ADC in our design. The advantage to this decision is that each bitline has its own ADC, so there is no need for additional circuitry for sharing a single ADC among each line. This simplistic nature should work well for a research chip that is more for proving the concept than getting precise results.

The following is a snippet of the ASIC Design of ReRam-based AI Accelerators Design Doc from the Iowa State University. It provides a high level overview of the project, including but not limited to requirements, the problem statement, goals, constraints, standards, project timeline, management, architectural design, and testing methodology. In no section does the document dive deeply into technical deeply because this is meant to be a preliminary before the project has not begun. In the section 4.1.3 outlines

some of the technical complexities and challenges they have identified that could alter the project timeline or be a barrier to development. They purposely use plain english providing a simple here is the context and why it is an issue. It is intended this way since this design docs likely audience will be people who may be funding the project that are likely not technically literate in the field.

Section 4.2.1 starts explaining the design decisions which are more technical than most other sections. However, the focus in the writing is still high level and also places an emphasis on how they are different from other reference models. They explain why they made certain decisions and the implications it could have.

Due to the informal nature of design docs there isn't a specific style being used, and is up to what the designers see best fit. The document was written by engineering designers and is meant for engineers and supervisors.

Secondary Source (for example, a person or industry expert talking ABOUT what the genre is & how people in the industry use it): Who is the person? What does the person say? Respond to the analytical questions above based on what the person says.

Summarize their perspective here and include a link to the source. Include a link to the source.

<https://www.industrialempathy.com/posts/design-docs-at-google/>

The following article was written by Malte Ubl who is the CTO of Vercel, and previously was the principle engineer responsible for Google Search Rendering and Engineering Director for Google Search. In this article he outlines the common structure of Design Docs at google. He makes note of their informal nature highlighting that “[d]esign docs are informal documents and thus don’t follow a strict guideline”. However, that is not to say there are certain criteria that good design docs should meet.

A design doc should provide a context and scope, what is the system being built and why. Goals and non goals should be a short list of bullet points. Importantly highlighting non goals, which are goals you aren't aiming for but your solution may end up implementing anyways such as meeting a compliance standard.

Then, the actual design overview where you start in general and go a little more into detail highlighting tradeoffs you made, why you made them and the benefits of those trade off. It's important to tie in relevant context like your goals and non goals, and why your current solution will satisfy them.

Diagrams are commonly used as they can be very useful for developers to visualize a plan or design that isn't just in words. Design Docs should also acknowledge the

constraints being faced and how it influences the direction of the design. This may also mean you'd have to include alternative solutions to show that your project will know what to do if the constraints halt development.

Design Docs ultimately serve a purpose to make sure a team has a uniform consensus on the project. It also allows them to seek advice from senior engineers or developers for assistance and review.

Tertiary Source (for example, a reference guide that explains the genre, including an encyclopedia, Credo, a credible industry database, Claude, Chat GPT and/or Perplexity. You can also use these platforms to find primary and secondary sources that are credible, authentic, and authoritative.) What did you discover? What is the platform and are they credible? What were the prompt/s you used to find this information? And answer the analytical questions above: what did you learn about the characteristics of the genre? What are its purposes? Who is the audience? Who are the authors? What is the writing style? Include a link to what it generated & your prompt/s.

Design docs is a lesser known genre of writing and information on it is scarce. That stems from its informal nature. I decided to use ChatGPT as my tertiary source due to the lack of a wikipedia page on design docs. I prompted ChatGPT to use the internet to gather information on what a design doc is and give me a high level overview of it. I confirmed the cited sources it used and is how I discovered my primary source. ChatGPT draws from a range of primary and secondary sources and thus can produce relatively accurate information.

What I discovered was the common characteristics of the genre, structured sections such as problem statements, requirements, constraints, design decisions and trade offs. Explicit discussion of assumptions, risks, and limitations. The use of diagrams, tables, and high level often simple language

I also learned more about the purpose of the genre, to purpose and evaluate a possible technical solution. Ensure engineers and those with stakes in a project agree before implementation. Document engineering reasoning and trade offs. Reduce the likelihood of error and miscommunication when in development.

They are commonly used by engineers, technical leads and reviewers. A secondary audience being project managers or administration. They are typically written by the project team themselves consisting of engineers.

The writing style is informal, neutral, clear and concise.

NOW IT'S YOUR TURN

**How would you use this genre? Who is your audience? What would you say?
What is your purpose?**

I had to create a design doc for a project I worked on in a lab. In that document I outlined my plan for implementation, some project goals I was aiming for, and the technical solution I came up with. I also outlined some of the key decisions I made and the trade off I had to make.

The audience was the professor I was working with so I was free to use technical terminology. The purpose was to get my project approved before diving into the implementation of the design. It primarily on keeping the project organized with a clear focus on the goals rather than loosely developing.

PART 2 - due by classtime Wed, Feb. 4

BIO & PHOTO & WEBSITE

Write 35-50 words that positions you as a qualified industry professional.

Copy and paste a photo of you in the space provided that displays your ethos as a qualified industry professional.

Website Options: Blog, Google Site, Canva, Wordpress (We will work on this in class, but you will also need to do it for homework, see syllabus for deadline)

Choose one & explain why this is the best choice (i.e. what are the "available means of persuasion"?)

PART 4 (We will do this in class the day the project is due)

REFLECTION

DO THIS REFLECTION & POST IT ON YOUR WEBSITE:

What approach did you take in composing these texts? How did looking at samples teach you to write something on your own? What did you learn about the company's audience, and reader-centered writing?

REFERENCES TO KEY TERMS (In case you need it!)

Primary: 1st hand text, original artifact (example: a letter that the company created)

Secondary: An article that talks about the texts the company uses, or provides some insight into the company's communicative genres

Tertiary (this can include ChatGPT, Claude, Perplexity): What encyclopedic, generative, or otherwise large language resource sheds light on the way this company creates letters, emails, and memos?

