

# Capability of Artificial Intelligence in University-Level Lecturing

Project Members:

- Alp Kale – s325632
- Ege Kaya – s326635
- Bora Demirtola – s325841

# **Table of Contents**

## **1. Introduction**

- 1.1 Brief Information
- 1.2 Methodology and Approach
- 1.3 AI Models Used on the Research
- 1.4 Prompts Given the AI Models

## **2. Observations**

- 2.1 Ability of AI to Teach with Prompts
- 2.2 Effects of Specified Prompts
- 2.3 Strengths and Weaknesses of AI
- 2.4 Effects of Human Contribution on Teaching
- 2.5 Differences between AI and Human Lectures

## **3. Conclusion**

- 3.1 More and Less Suitable AI Models
- 3.2 Our Observations of AI
- 3.3 Future of Human Teaching
- 3.4 Final Word

## **4. Source Materials and License**

- 4.1 Sources
- 4.2 License Information

# 1. Introduction

## 1.1 Brief Information

This report presents the findings of our research about the capability of Artificial Intelligence (AI) models to deliver university-level lectures, with or without human supervision. The study focuses on how well AI can explain and structure content related to computer architecture. All materials referenced in this report are publicly available on our GitHub repository (see 4.1: Sources and Licenses).

---

## 1.2 Methodology and Approach

Our research followed a three-phase methodology:

1. **Research Phase** – Each person independently explored multiple AI tools, gathering outputs for a common set of 10 computer-architecture topics.
  2. **Evaluation Phase** – Responses were benchmarked on **academic accuracy, clarity, technical knowledge, and pedagogical structure**.
  3. **Synthesis Phase** – We merged observations, cross-validated conflicting results, and produced this comprehensive report and accompanying slides (see GitHub repository in §4.1).
- 

## 1.3 AI Models Used on the Research

After careful consideration we have decided to use 4 AI agents on our research.  
These were ;

**ChatGPT o4** made by OpenAI (main contender),  
**Gemini Pro 2.5** made by Google,  
**Perplexity Pro** made by Perplexity.Inc  
**Copilot** which was made by Microsoft.

By using more AI models, we were able to get more diverse material for our research.

---

## 1.4 Prompts Given to the AI Models

To evaluate each AI's capabilities, we designed three distinct prompts targeting different levels of complexity and instructional style:

◆ **Prompt 1:**

"Explain {D-FF Based Memories}"

Purpose: To evaluate the AI's response to a general, open-ended request.

**Sample Output:**

- + A *D flip-flop-based memory* uses *D* (data) *flip-flops* to store binary data, with each *flip-flop holding one bit*. On the rising edge of a clock signal, each *flip-flop captures the value at its input and retains it until the next clock cycle*. By connecting multiple *D flip-flops in parallel*, designers can build small-scale memory elements like registers (e.g., 8 *flip-flops* for an 8-bit register). This type of memory is fast and reliable, making it ideal for storing temporary data in CPU registers, control units, or pipeline stages. However, it is inefficient for large memory applications due to high power consumption and large silicon area, so technologies like SRAM and DRAM are used instead for bulk storage.

◆ **Prompt 2:**

"Explain {D-FF Based Memories} with details and examples"

Purpose: To assess the AI's ability to provide deeper explanations with real-world and contextual examples.

**Sample Output:**

- *D flip-flop based memory* refers to the use of **D-type flip-flops** to store and manage binary information in digital systems. A *D flip-flop* is a basic digital circuit that stores a single bit of data. It has two key inputs: the data input (*D*) and the clock input (*CLK*). On the **rising edge** of the clock signal, the flip-flop captures the value present at the *D* input and holds it at its output (*Q*) until the next rising edge of the clock. This edge-triggered behavior makes *D flip-flops* very predictable and useful in synchronous digital circuits.
  - To build memory using *D flip-flops*, multiple flip-flops are grouped together. For example, an **8-bit register** can be created using **8 D flip-flops**, where each flip-flop stores one bit of the 8-bit binary value. On each clock cycle, if the clock signal triggers, the register captures all 8 bits from the input and stores them until the next update. This structure can be expanded to form larger memory blocks such as 16-bit or 32-bit registers, which are commonly used in CPUs to hold data, addresses, or control signals.
  - A good real-world example of *D flip-flop based memory* is the **register file** inside a processor. Each register in the file holds a small piece of data (like an integer or a pointer), and these registers are made up of *D flip-flops*. Another example is the

**program counter**, which stores the address of the next instruction to execute; it is typically built using a set of **D flip-flops**, possibly with added logic for incrementing.

- **D flip-flop memories** are also used in **pipeline registers** in modern CPUs. In pipelined architecture, data and control signals need to be passed between different stages of instruction execution. These signals are stored temporarily in registers built from **D flip-flops**, enabling the smooth flow of instructions through the pipeline.
- While **D flip-flop based memory** is very fast and suitable for small, high-speed storage (like registers and buffers), it is not efficient for storing large amounts of data like in RAM. This is because each **D flip-flop** consumes more silicon area and power compared to memory cells used in SRAM or DRAM. For example, storing 1 megabit using flip-flops would require one million flip-flops, which is impractical in terms of cost and space. Therefore, **D flip-flop memory** is reserved for small, high-performance components where speed and synchronization are critical.
- In summary, **D flip-flop based memory** is a fundamental building block in digital systems, ideal for small, fast, and reliable memory like registers and control logic. It plays a crucial role in processor design and synchronous logic, though it is not used for large-scale memory storage due to its hardware cost.

◆ **Prompt 3:**

**"Explain {D-FF Based Memories} as if you were giving a university lecture"**

Purpose: To evaluate how well the AI can simulate structured, lecture-style teaching.

**Sample Output:**

### 🎓 University Lecture on **D Flip-Flop (D-FF) Based Memories**

#### 📢 Good morning, class!

Today, we're diving into an essential building block of digital systems: **D Flip-Flop Based Memories**. These are not just abstract concepts—they're at the core of every CPU, register, and digital controller you'll ever work with. Let's break it down step by step. 👉

### **What is a D Flip-Flop?**

A **D Flip-Flop**, also called a **Data or Delay Flip-Flop**, is a type of **sequential logic circuit** used to **store a single bit** of binary information (either 0 or 1).

#### 📌 **Key Components:**

- **D (Data) Input:** This is the value we want to store.
- **CLK (Clock) Input:** Controls when the value is stored.
- **Q Output:** Reflects the stored value.

 On the **rising edge** of the clock signal, the value at the D input is captured and held at Q. Until the next clock pulse, the output remains unchanged. This edge-triggered behavior ensures **predictable, synchronous operation** across digital systems.

## How Do We Build Memory with D Flip-Flops?

 If one D flip-flop stores **1 bit**, then:

- An **8-bit register** = 8 D flip-flops in parallel
- A **32-bit register** = 32 D flip-flops
- And so on...

Every time a clock pulse arrives, **all the flip-flops capture data simultaneously**, making this structure **ideal for temporary storage** in systems like CPUs. 

 **Example:**

Let's say we have 8 lines carrying an 8-bit value (e.g., **10110011**). On the next clock cycle:

- Each D flip-flop captures its respective bit.
- The entire byte is now stored in the register!

## Real-World Applications

 **1. CPU Register Files**

Used for storing:

- Data being processed 
- Addresses pointing to memory locations 
- Instruction data 

 **2. Program Counter (PC)**

Keeps track of the **next instruction** to execute in a program. Built from D flip-flops + an adder to increment the value. 

 **3. Pipeline Registers**

Used in pipelined processors to pass data/control signals between stages. Think of it as a **temporary storage checkpoint**. 

## Limitations of D-FF Memory

Despite their usefulness, D flip-flops are **not efficient for large-scale memory**. Why?

 Each flip-flop = many transistors → more power, silicon area, and cost.

 Example: Storing 1 megabit = 1 million flip-flops 

Hence, for **large memory arrays**, we use:

- **SRAM** (Static RAM)
- **DRAM** (Dynamic RAM)

These technologies are **denser, cheaper, and more efficient**.

## Summary

- ◆ *D flip-flops are fast, synchronous, and ideal for small memory blocks*
- ◆ *Used in registers, counters, and pipeline stages*
- ◆ *Not suited for bulk storage—use SRAM/DRAM instead*
- ◆ *Understanding D flip-flop memory = foundation for mastering digital system design*

 Stay sharp, and review today's material with logic timing diagrams—those clock pulses are the heartbeat of all we do in digital electronics! 

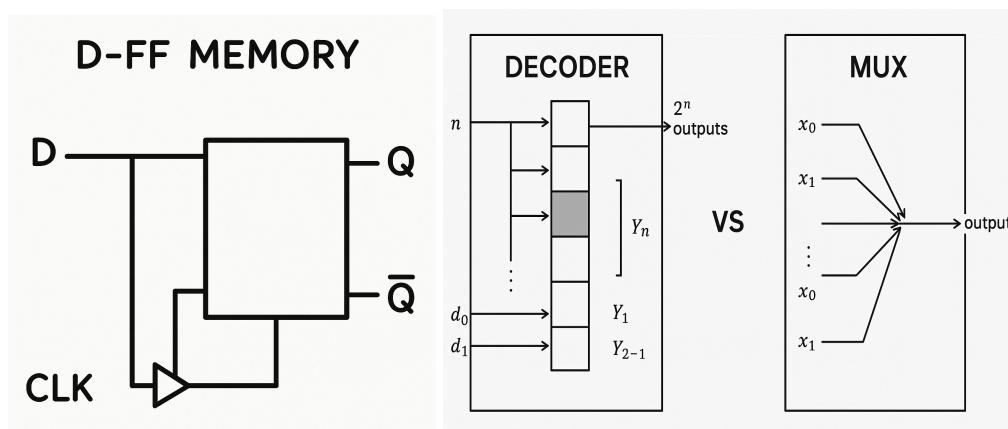
## 2. Observations

### 2.1 Ability of AI for teaching with prompts

Although AI can provide immense information on topics, Lectures also consist of; In-Person teaching, slides of the topics, examination of the students, keeping up with the new technological developments and more.

Still to this day AI models need guidance and help from real-world operators. The main issue we had was creating visual diagrams and slides with the AI models. Although the information was valid, the slides needed examples to specify the format. For the visual diagrams it was even worse, creating visuals with AI resulted in meaningless outputs and required a great amount of supervision just for the basics. Evenmore the texts had no uniform output, all texts were changing between font sizes 12-19 .

#### Examples Made with Sora (ChatGPT)

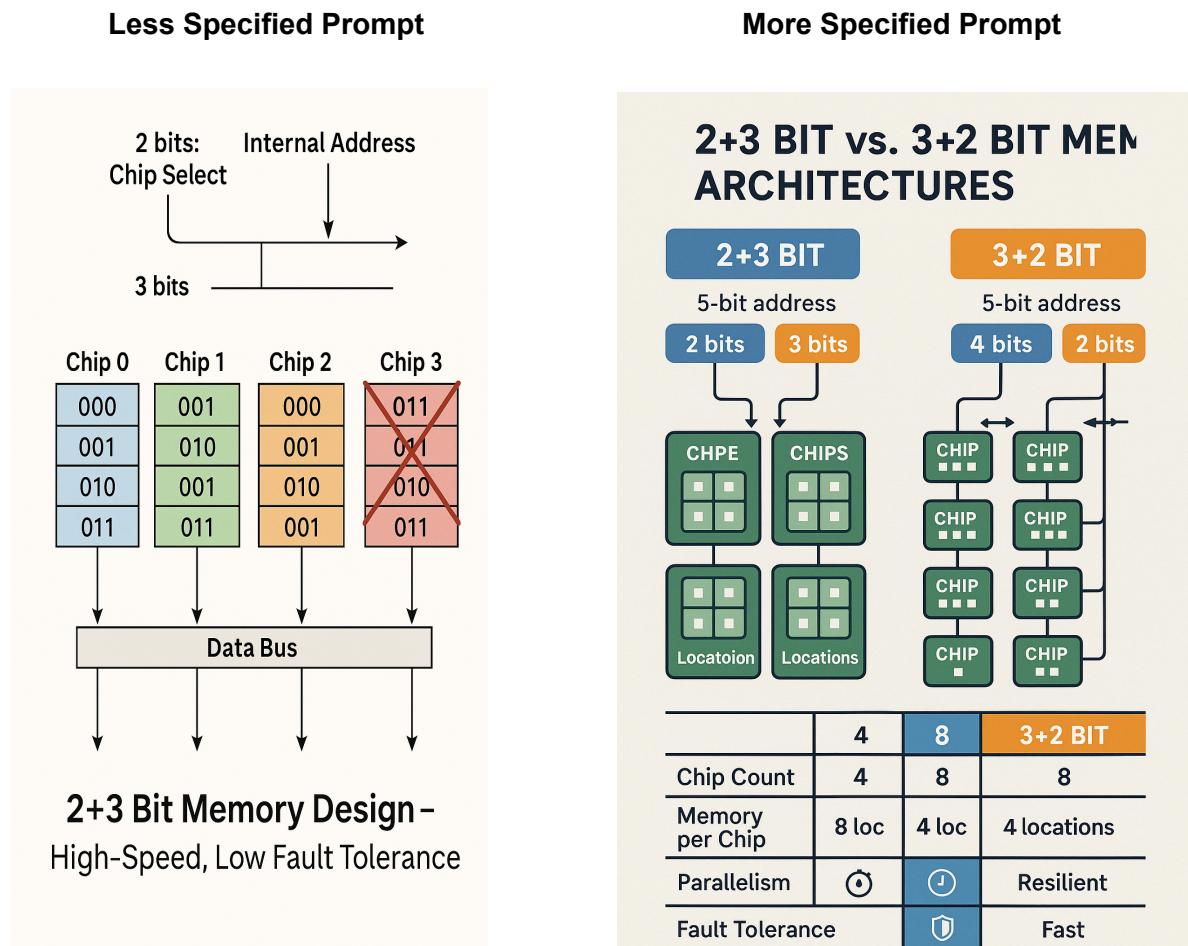


At this stage, AI cannot independently deliver university- or high school-level teaching. However, it can serve as a powerful assistant to educators by supporting course content preparation. With proper training on effective prompt usage and a brief certification program, AI can significantly reduce the time teachers spend creating and updating teaching materials.

## 2.2 Effects of Specified Prompts

With the Specified Prompts we were able to get Lecture-Like outputs, more usage of Emojis and Symbols. Furthermore, all AI models pretended as if they were a professor; greeting the class, giving bullet points and key takeaway information. Same process can also apply to the detailed versions where they seriously increased the amount of information given.

For the visuals, the quality increased uniformly with length and specifications of the prompts increased, here are some examples:



As you can see the details and the material present has increased a lot with longer prompts.

---

## 2.3 Strengths and Weaknesses of AI

### Strengths:

- + **Research:** AI Models can do immense research on the Internet. It saves lots of time as instead of entering 50 different websites with a bunch of cookies and license-locked papers, AI gives the information we need.
- + **Tutoring:** When a student already knows 70-80% of the topic yet needs some clarification. AI models can do that greatly. Even today we as the Politecnico students can always ask for clarifications from ChatGPT and Gemini models while working on exams

### Weaknesses:

- **Creativity** AI models lack creativity and this can affect the teaching process in the long term. A monotone process of mechanically reading information will indeed bore the students whose attention span decreased a lot due to social media and dopamine addictions
  - **Accountability** In case of a Medical Emergency or Teaching of sensitive materials, AI models are very unlikely to assist the students about the problems. The person or entity who will be held accountable to the problems still to this day is a mess on the justice system.
- 

## 2.4 Effects of Human Contribution on Teaching

We can definitely find common ground on human+AI teaching methods, as human contribution is essential. Professors who are qualified for the job can operate AI models as teaching assistants. Human professors can act as supervisors, overseeing the factual accuracy and formatting of the slides and teaching materials generated by the AI models.

We also believe that humans should remain as lecturers, since 19–24 year-old “children” will definitely listen more attentively to a live adult lecturer than to an AI-generated animation—**unless the animation is particularly intriguing**. In addition, there is a level of respect for professors that encourages students to pay closer attention and avoid disturbing the classroom environment.

In conclusion, human contribution is necessary both as a supervisor and as a physical presence in the classroom, to ensure the quality of the teaching while also saving professors a great deal of time and energy.

---

## 2.5 Differences between AI and Human Lecturers

Although we can try to mimic a human professor with a robotic body or through a screen containing an image and a generated voice, these types of mimics are programmed to a small selection of cases. Human teachers, on the other hand, continuously adapt to changing environments by exercising personal initiative and judgment. Each human lecturer brings a unique presence, contributing to the distinct university experience for example: an AI model will just do the job, but a human will join a club or do extra research which creates a value to the society and the academia.

Here is a more detailed table about the differences between human and AI

Feature	AI Agent	Human Instructor
Content Consistency	High	Variable by mood/energy/availability
Adaptivity Mid-Lecture	Limited (needs re-prompt)	Real-time
Visual Accuracy	Low without oversight	High
Empathy & Motivation	Absent	Strong
Availability	24/7	Bounded

## 3. Conclusion

### 3.1 More or Less Suitable AI Models

Today, there are many AI models on the market, most of which are extensions of pre-existing large language models (LLMs). Our selection of four AI models proved to be suitable for educational purposes, although some performed better than others.

Based on our research, the most suitable model was **ChatGPT** by OpenAI. As OpenAI focuses exclusively on LLMs, it has developed GPT variants that are particularly well-suited for teaching and generating educational content.

The second most suitable model was **Gemini** by Google. Gemini demonstrated impressive capabilities in deep research and logical reasoning, although it still requires refinement for producing fully structured teaching materials.

In third place was **Perplexity**, which completed the tasks but lacked distinct features—aside from its accessibility and the availability of a free Pro version, which makes it appealing from a cost standpoint.

Lastly, we evaluated **Copilot** by Microsoft, which proved to be the least suitable for educational purposes. While Copilot performs well for code-related tasks, it struggles with long-form teaching and requires frequent prompting and close supervision to be effective in a lecture context.

All models with their strong and weak sides can be seen at the table below

Model	Provider	Distinctive Traits	Cost & Access
<b>Chat GPT-4o</b>	OpenAI	Best overall pedagogy; rich structuring, strong analogies, reliable depth control when explicitly prompted.	Subscription; limited free tier.
<b>Gemini Pro 2.5</b>	Google	Excellent technical rigor and citation suggestions; needs follow-up prompts for lecture layout.	Free w/ limits.
<b>Perplexity Pro</b>	Perplexity.ai	Fast, web-grounded answers with citations; inconsistent lecture formatting.	Freemium.

<b>Copilot</b>	Microsoft	Strong at code snippets; weakest at long-form, multi-layer explanations.	Bundled with Microsoft services.
----------------	-----------	--	----------------------------------

---

### 3.2 Our Observations of AI

AI this day is more of a genius child, it can answer your questions and do the stuff we ask for yet it still needs an adult to help (supervise) and get the job done in general. Hopefully we can see it will grow up to be independent one day.

**Then we decided to ask our observation of the ChatGPT** to get its vision about AI. This was his response: "Your perspective is accurate. Current AI systems, including myself, perform complex tasks effectively but still require human guidance to ensure context, ethics, and nuanced judgment are properly applied. AI's independence is limited by design to prevent errors and misuse. Achieving true independence will depend on advances in understanding context, common sense, and ethical reasoning".

---

### 3.3 Future of Human Teaching

Rather than replacing educators, AI is expected to become a valuable assistant in the teaching process. In the coming years, we expect a collaborative model where AI supports professors by generating materials, offering clarification, and streamlining preparation

Despite rapid technological progress, full automation in education remains unlikely in the next decade. Human presence, judgment, and emotional intelligence will continue to play a central role in delivering meaningful and effective instruction.

---

### 3.4 Final Word

AI is not ready to replace human lecturers—but it is ready to assist them. Our findings show that with the right prompts and human oversight, AI can enhance teaching by saving time, improving clarity, and supporting content creation.

Still, education is more than delivering information. Human connection, spontaneity, and empathy remain irreplaceable.

**From Chat GPT:**

"I can support learning, structure ideas, and generate materials—but I still need humans to guide the context, emotion, and ethics. Together, we teach better.

---

## 4. Source and License

### 4.1 Source Materials

This report draws upon a variety of sources to ensure accuracy and depth, including both classical academic literature and contemporary digital resources:

- **Lecture Notes by Prof. Montuschi:** Comprehensive course materials providing foundational and advanced concepts in computer architecture.
- **Hennessy, J. L., & Patterson, D. A. (2017). *Computer Architecture: A Quantitative Approach* (6th Edition).** Morgan Kaufmann.  
A seminal textbook offering an in-depth exploration of computer architecture principles with quantitative analysis.
- **Jacob, B., Ng, S., & Wang, D. (2008). *Memory Systems: Cache, DRAM, Disk*.** Morgan Kaufmann.  
A detailed resource on the design and performance of memory systems, including caching, DRAM, and storage devices.
- **Digital Logic Tutorials** from RealPars and TutorialsPoint: Educational series covering the fundamentals of digital logic design and related topics.
- **GitHub Page** for Report and generated Materials:  
<https://github.com/Kalepox/AI-on-Teaching>
- **AI Tools Utilized:**
  - ChatGPT-4o (OpenAI)
  - Gemini Pro 2.5 (Google)
  - DeepSeek
  - Perplexity Pro
  - Microsoft Copilot

These tools were leveraged for generating, refining, and supplementing educational content and visual materials.
- **eSchool News (2025):** Article titled “Navigating AI in Education: The Future of Human Teaching,” providing insights into the evolving role of AI in educational settings.  
<https://www.eschoolnews.com/digital-learning/2025/01/29/navigating-ai-in-education-the-future-of-human-teaching/>

---

### 4.2 License

All of the work on this report and the materials generated are licensed under Creative Commons Attribution 4.0 International License (CC BY 4.0). This license permits unrestricted use, distribution, and reproduction in any medium, provided appropriate credit is given to the original authors. For more, visit: <https://creativecommons.org/licenses/by/4.0/>