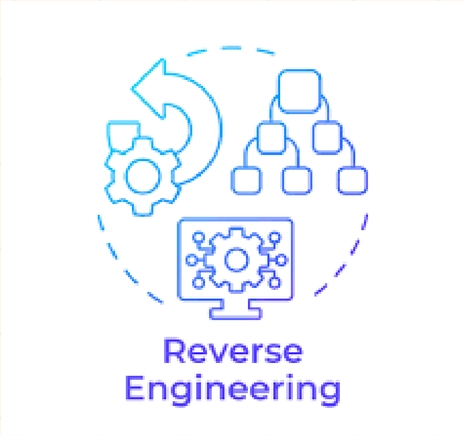
MINI PROJECT
EVALUATION



Reverse Engineering and **Emulation** of **NES Hardware** with Integrated Debugger

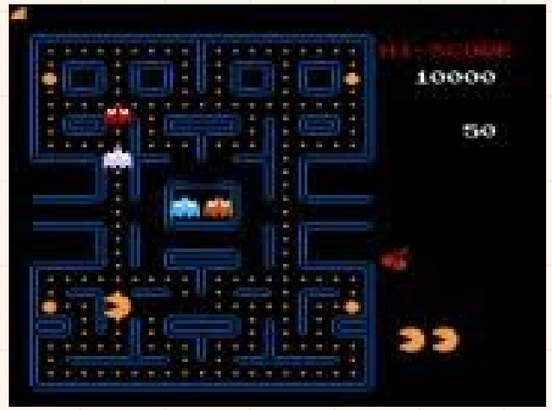
Ojas Kumar – 23bcs059
Talatiya Mitraj – 23bcs092

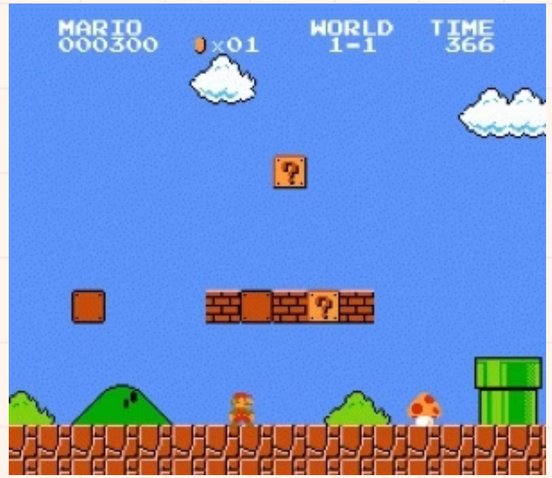
Mentor/Guide – Dr. Sonika Gupta

Problem Statement

Emulation is a core area of **systems programming and reverse engineering** , allowing **legacy hardware** to be replicated in software. The **Nintendo Entertainment System (NES)**, launched in the **1980s**, is one of the most studied consoles **(CPU 6502)** due to its influence on computer architecture and gaming history.

So, **how** can legacy hardware (NES) be reverse engineered and emulated in Rust while also **providing modern debugging** and **profiling features** to enhance usability for both learners and developers?





OBJECTIVES

- To **study** and reverse engineer the **NES hardware architecture** (CPU, PPU, APU, Memory Bus)
- To design and implement an emulator in **Rust** with modular and extensible architecture.
- To **integrate a monitor/debugger** supporting instruction tracing, breakpoints and profiling
- To **analyze performance and correctness** of the emulator against existing test ROMs.

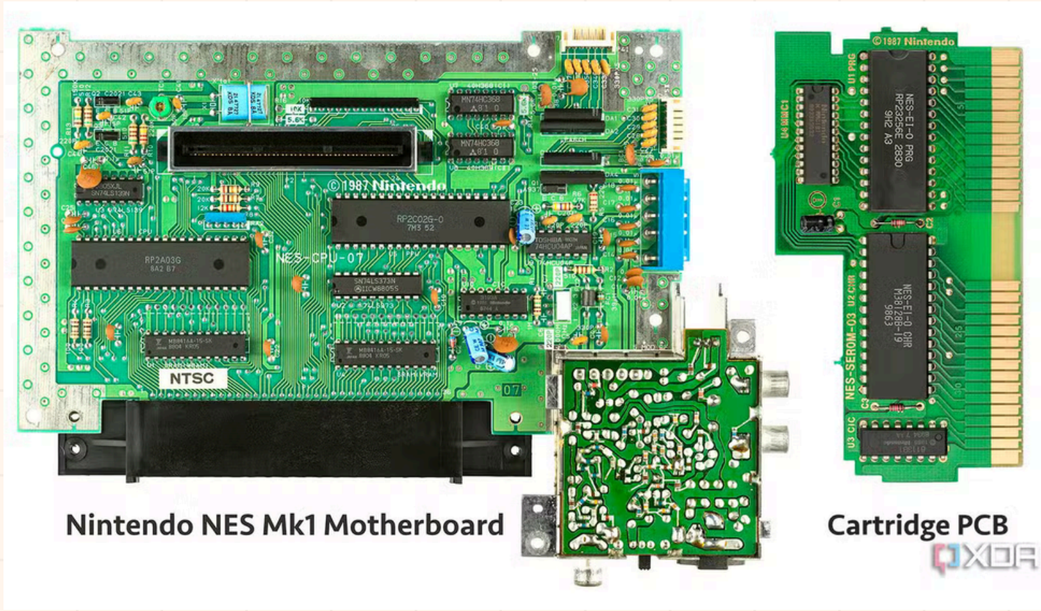
PROPOSED SOLUTION & METHODOLOGY

- Literature Survey & Reverse Engineering
  - Study NES Architecture (6502 CPU, PPU, APU)
  - Review existing emulators (FCEUX, Nestopia, Mesen) to identify strengths and limitations
- Design & Implementation
  - Implement CPU instructions set and memory bus using Rust structs and enums
  - Simulate PPU (video rendering) and APU (sound) subsystems
  - Develop modular architecture seperating core emulation and debugging tools.
- Debugger & Monitoring System
  - Build Step-Through execution engine with breakpoints
  - Collect statistics (instruction frequency, CPU/GPU cycles, memory usage).
- Testing & Validation
  - Run official NES test ROMs and compare behaviour with reference emulators.

THE 6502 CPU

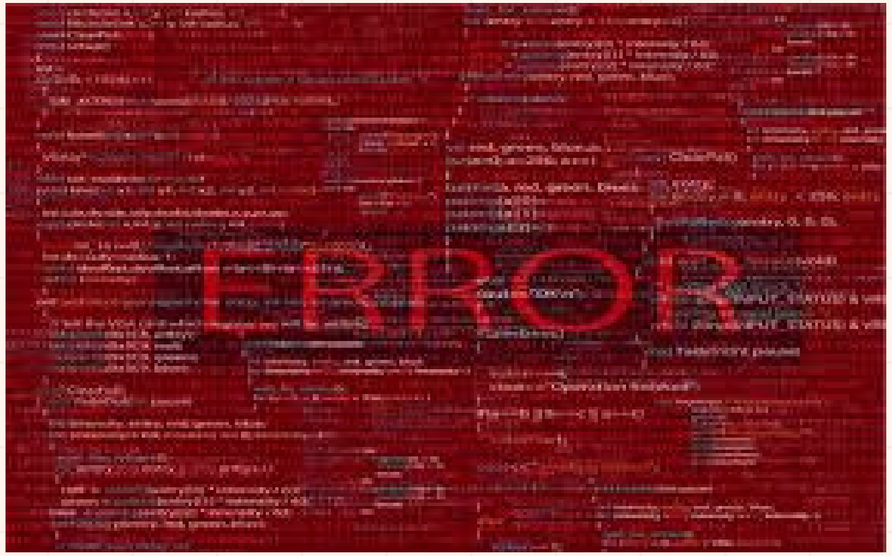
OUR PROGRESS TILL DATE

- **CPU** – Emulates the 6502 processor, running entirely in interpreted mode and lacking support for floating-point (BCD) operations.
- **Bus** – Acts as the communication backbone, connecting and synchronizing all NES components to operate in perfect cycle harmony.
- **PPU** (Picture Processing Unit) – Handles all graphical rendering for the NES, operating on a precise scanline-by-scanline basis.
- **Cartridge** – The game's ROM module that loads the program data directly into the NES system.
- **JoyPad** – The iconic NES input controller, delivering classic and responsive player interaction.
- **APU** (Audio Processing Unit) – Generates system audio, though sound remains partially distorted as the DMCA channel emulation is still in progress.



CHALLENGES FACED & HOW WE ADDRESSED THEM

- Writing a Rust equivalent for every single 6502 CPU instruction was highly time-consuming and error-prone.
- The PPU (Picture Processing Unit) had complex addressing mechanisms, making accurate emulation difficult.
- Frequent program hangs occurred due to synchronization issues between CPU and PPU.
- Required a shift to a multithreaded architecture to maintain stability and responsiveness.
- Achieving cycle-accurate timing was essential to prevent graphical glitches and crashes.



FUTURE WORK & SCOPE

- Implement the DMC (Delta Modulation Channel) to complete NES audio emulation.Implement the DMC (Delta Modulation Channel) to complete NES audio emulation.
- Add support for multiple cartridge mappers beyond the currently supported NROM.
- Develop a more graphically enhanced debugger for better real-time analysis and visualization.
- Integrate a PPU tile viewer to inspect background and sprite rendering in detail.

REFERENCES

- Ricci, A., Programming the 6502, Sams Publishing, 1983
- NESDev Wiki: https://www.nesdev.org/
- Matt Godbolt, “6502 CPU Emulation in Practice,” IEEE Software, 2018
- FCEUX Emulator Documentation: http://www.fceux.com/
- NesDev Forums (community contributions on NES Architecture and emulation)
- Rust Programming Language Book: https”//doc.rust-lang.org/book/

WE THANK YOU FOR YOUR KIND ATTENTION
WE WISH YOU A HAPPY DAY !