

# INTEL 8080 EMULATOR

## Building an Intel 8080 emulator in C that can play classic arcade games

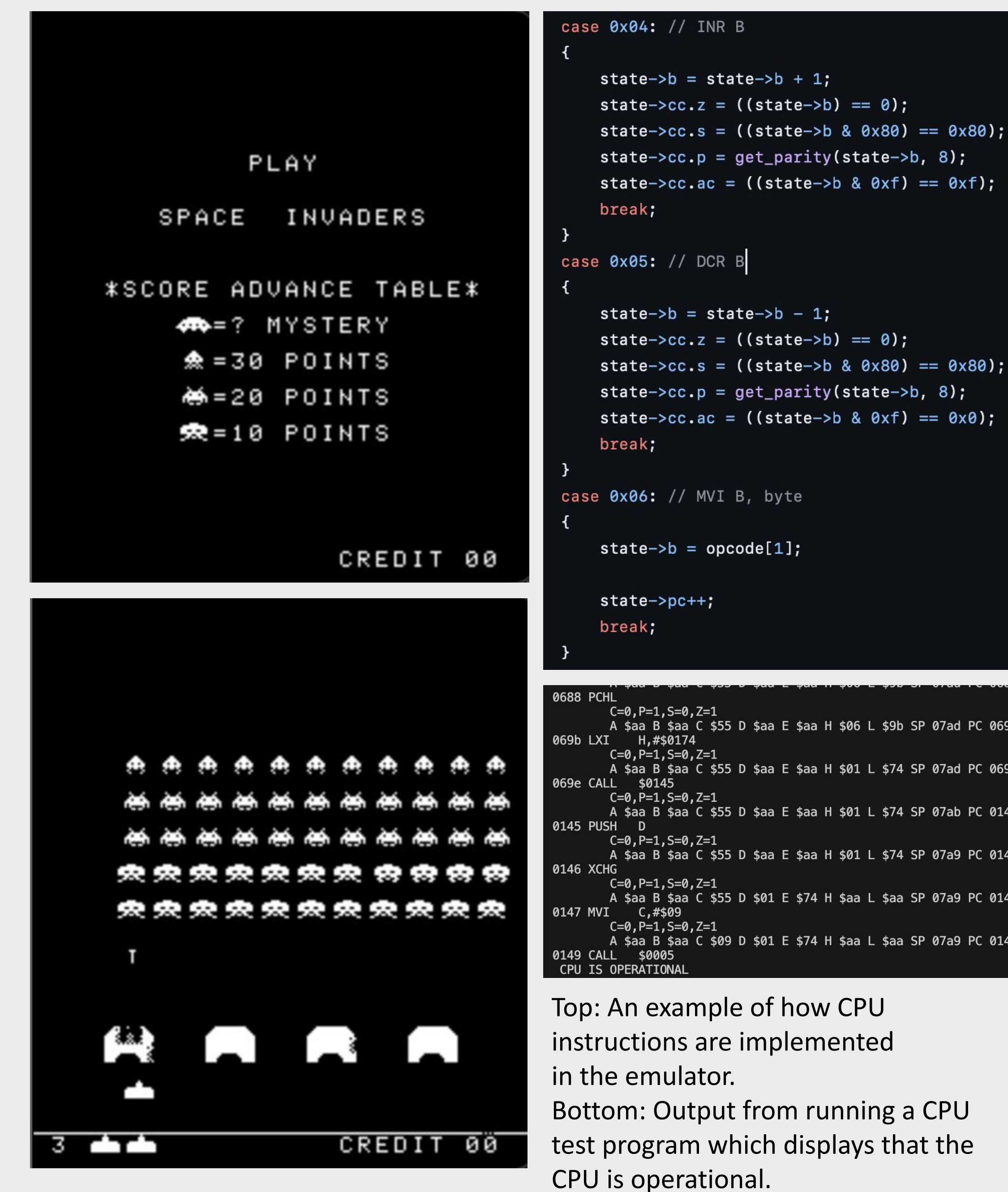
### Team

Christopher Tu (tuchr@oregonstate.edu)

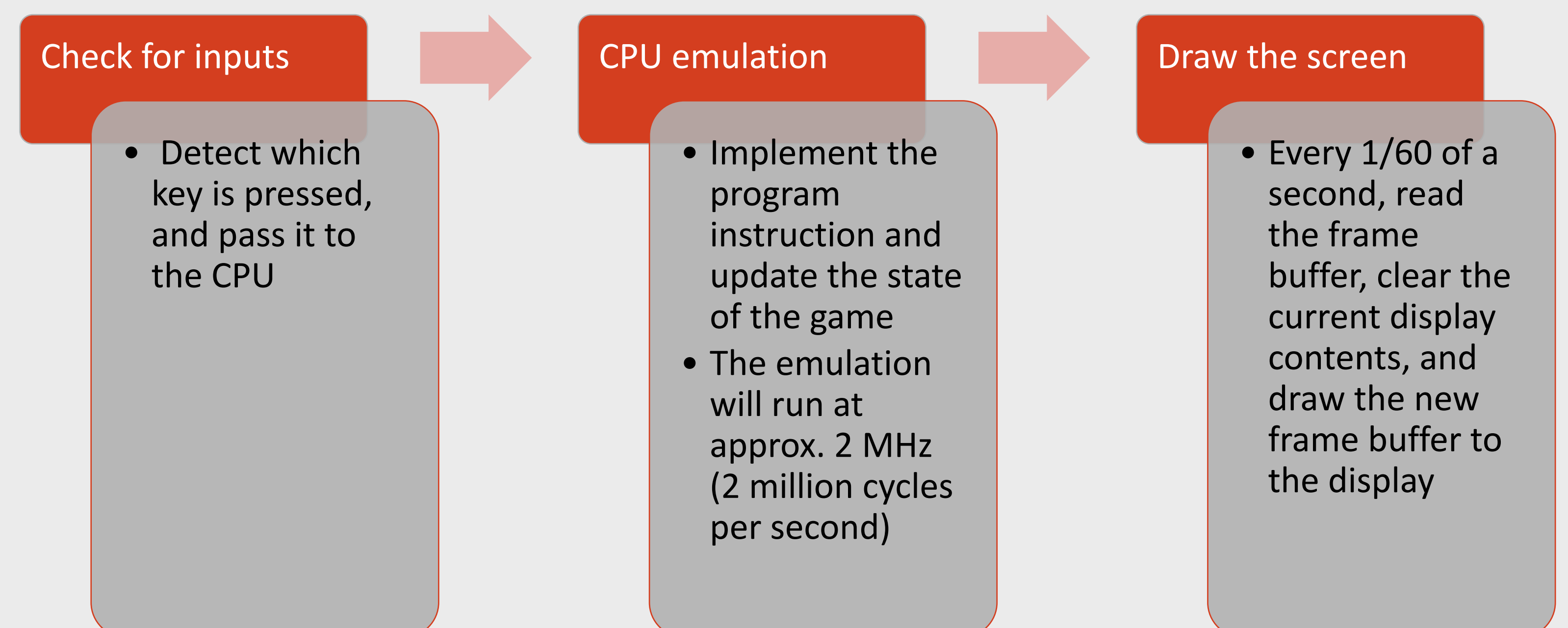
Hannah Boehm (boehmha@oregonstate.edu)

### BACKGROUND

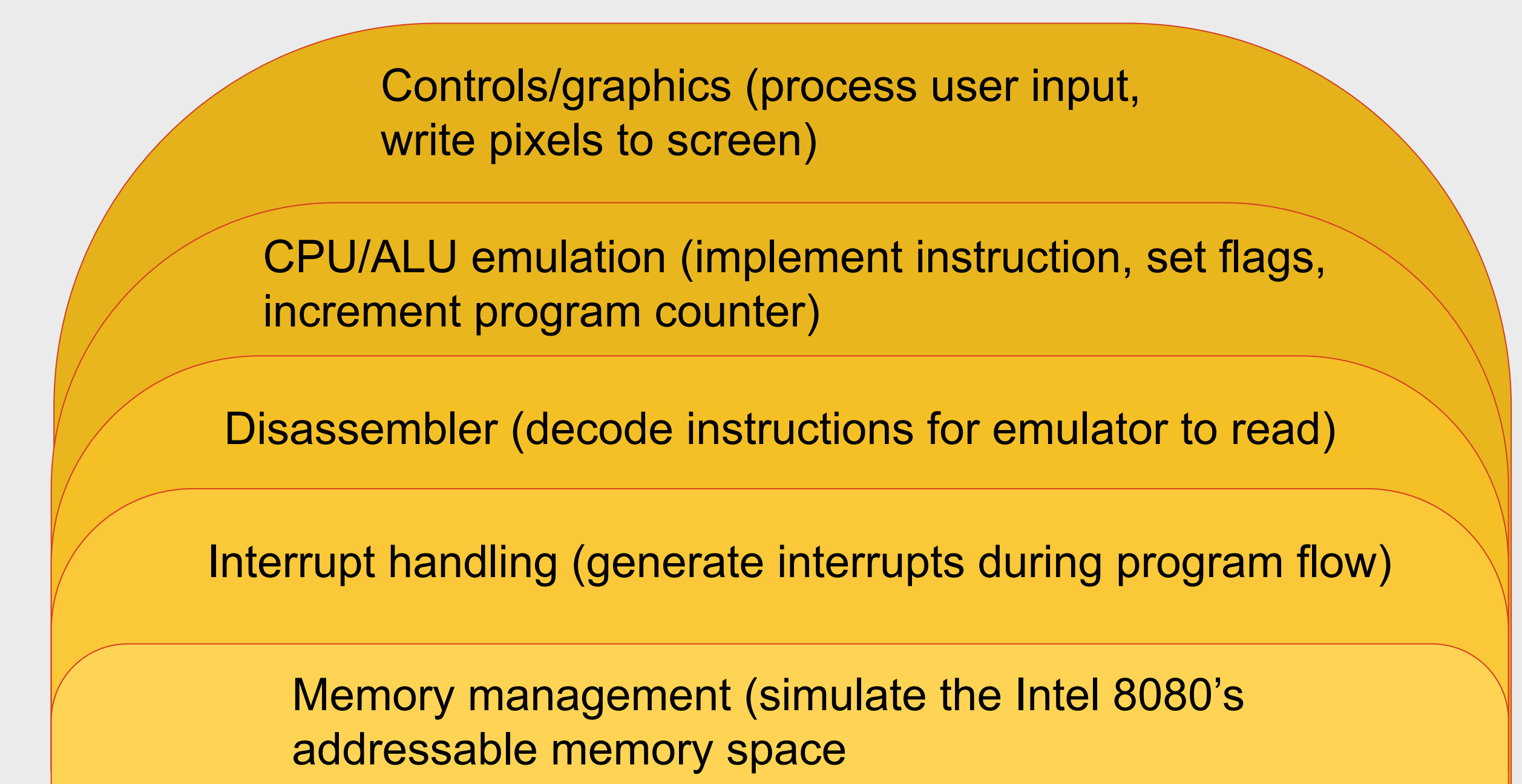
- This project gave us the opportunity to learn more about computer architecture, processor operations, and the different layers of building an emulator.
- We used the C language and the SDL Library to develop the program.
- We also were exposed to assembly language instructions when disassembling the Space Invaders ROM and during the process of debugging.
- The SDL Library was used to detect keyboard input from the user, and also draw graphics on the screen by reading from the game's frame buffer.
- We divided the emulator code into several modules, including the CPU, ALU, memory, graphics, and controls, in order to keep the code more readable and maintainable.
- This project has been a great experience in learning the lower-level side of software development and debugging.
- Repo: <https://github.com/h-boehm/i8080-emulator.git>



### Game Loop



### Emulation Layers



## IMPLEMENTATION PROCESS

- We started the project by learning how to disassemble the Space Invaders ROM file. This allowed us to convert the binary instructions into a slightly more human-readable assembly language format, and it also gave us a starting point to begin working on the CPU emulation.
- Next, we started working on the memory module of the program, which enabled us to load in the Space Invaders binary file into memory. The program is divided up into 4 files that are 2048 bytes each.
- Using the disassembler code as a basis, we started to implement the CPU instructions required to run the Space Invaders game. These instructions were divided into arithmetic, logical, branching, input/output, and stack related subgroups, performing operations such as addition and subtraction, AND/OR/XOR, jumps, and push/pop values from the stack.
- In order to run Space Invaders, we needed to handle CPU interrupts that are generated by video hardware in the real Space Invaders arcade machine. These interrupts are sent in the middle of a frame and at the end of a frame, and occur 60 times per second (60 Hz). The program will not run unless the interrupts occur.
- We used the SDL library in order to draw graphics and take keyboard input. The program will read from the game's frame buffer, which contains the graphics data, and set a pixel value on the screen to black or white. To read from the keyboard, the program will poll for any user input events, detect which key is pressed, and then pass the event to the CPU.

## FEATURES

- Contains several layers that handle memory and interrupts, decode the ROM instructions, emulate the processor architecture, and simulate the hardware.
- Control and graphics peripherals enable user interaction and a visual display.
- Easy-to-build program that compiles with make - running the executable will bring up the game window for the user to play.

## OUTCOMES

- Retro games are a valuable part of tech history, and emulators bring those experiences back to life for today's generation
- Individuals who played the game during the 70s-80s can revisit those memories
- Building this program was a great way to learn about processor architecture in depth