

# FPG-8: A Chip-8/S-Chip implementation in Verilog

---

Guilherme Tebaldi Meira



**Worcester Polytechnic Institute**  
Department of Electrical and Computer Engineering

# Chip-8

- Simple interpreted language
- Created in the mid-1970's for the COSMAC VIP and Telmac 1800
- Focused on the creation of simple games
- “Hello world” of emulators



# Chip-8 Specifications

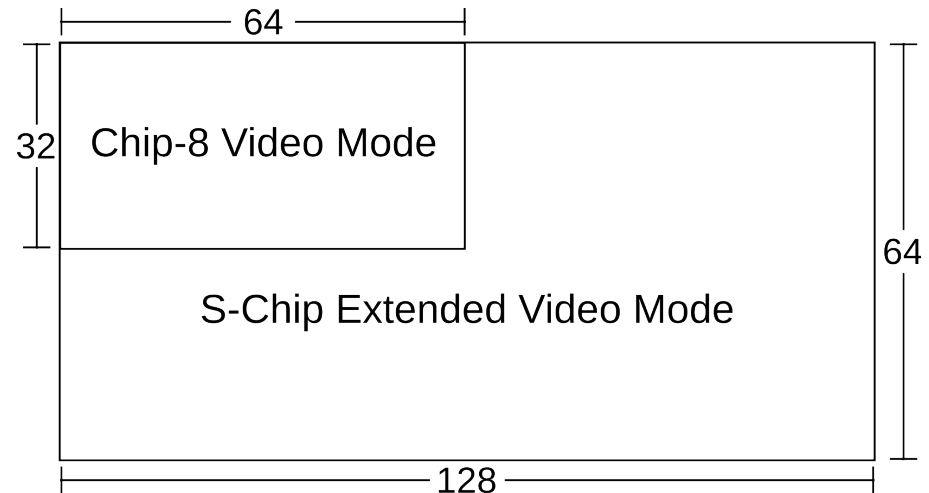
- 4kB of RAM
- 16 general purpose registers (V0-VF)
- Stack for function calls
- 60Hz Timers
  - Delay timer
  - Sound timer
- Hexadecimal keyboard
- 64x32 monochromatic display
- Sprites
  - 8 pixels wide
  - 1 to 10 pixels tall
- 35 opcodes

1	2	3	C
4	5	6	D
7	8	9	E
A	0	B	F

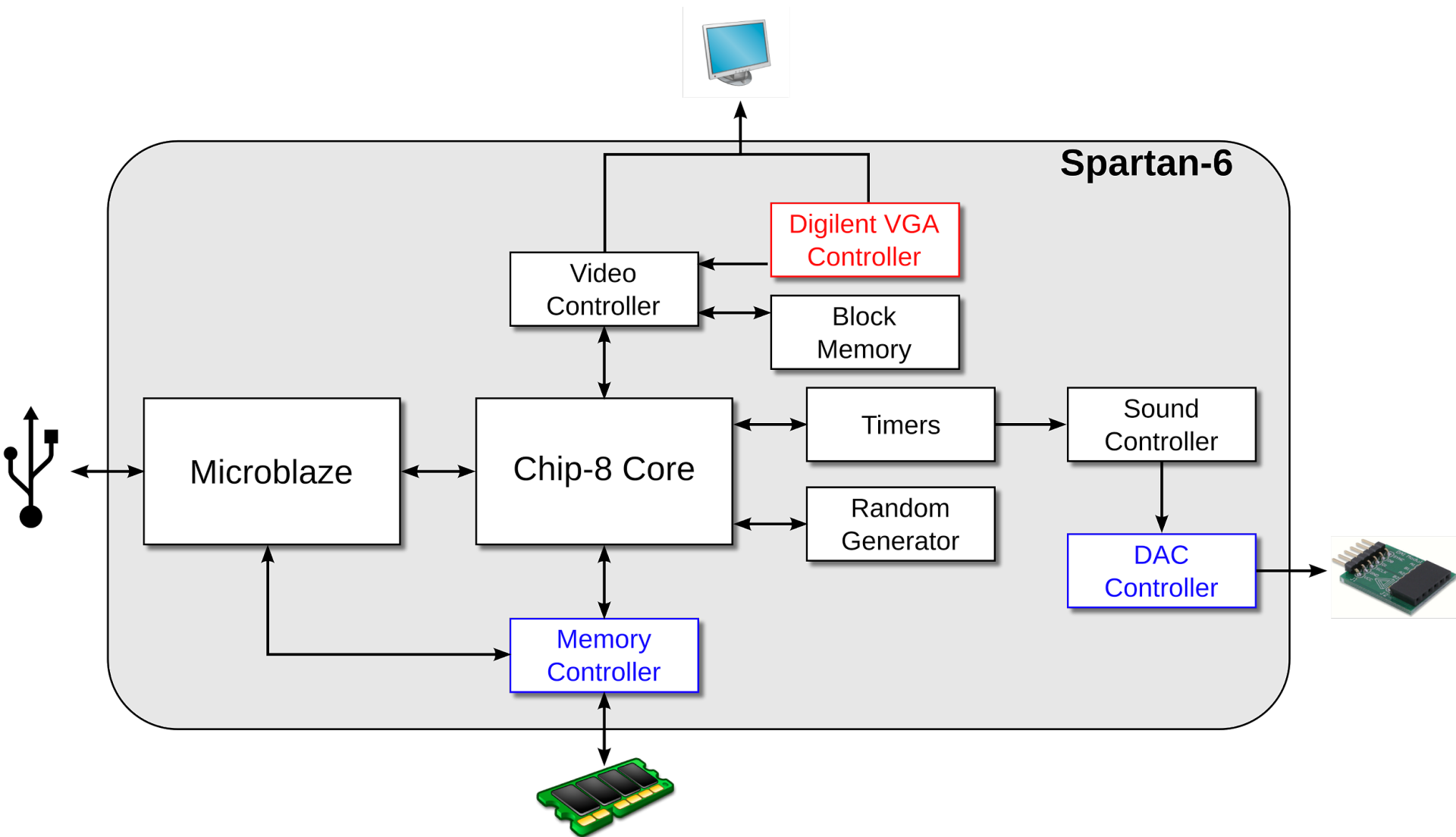


# S-Chip Extension

- 128x64 video mode
- 16x16 sprites
- 10 new opcodes
  - Screen manipulation (scroll up, left and right)
  - Switch between video modes
  - Interaction with HP-48 (not implemented)
  - Exit



# Implementation



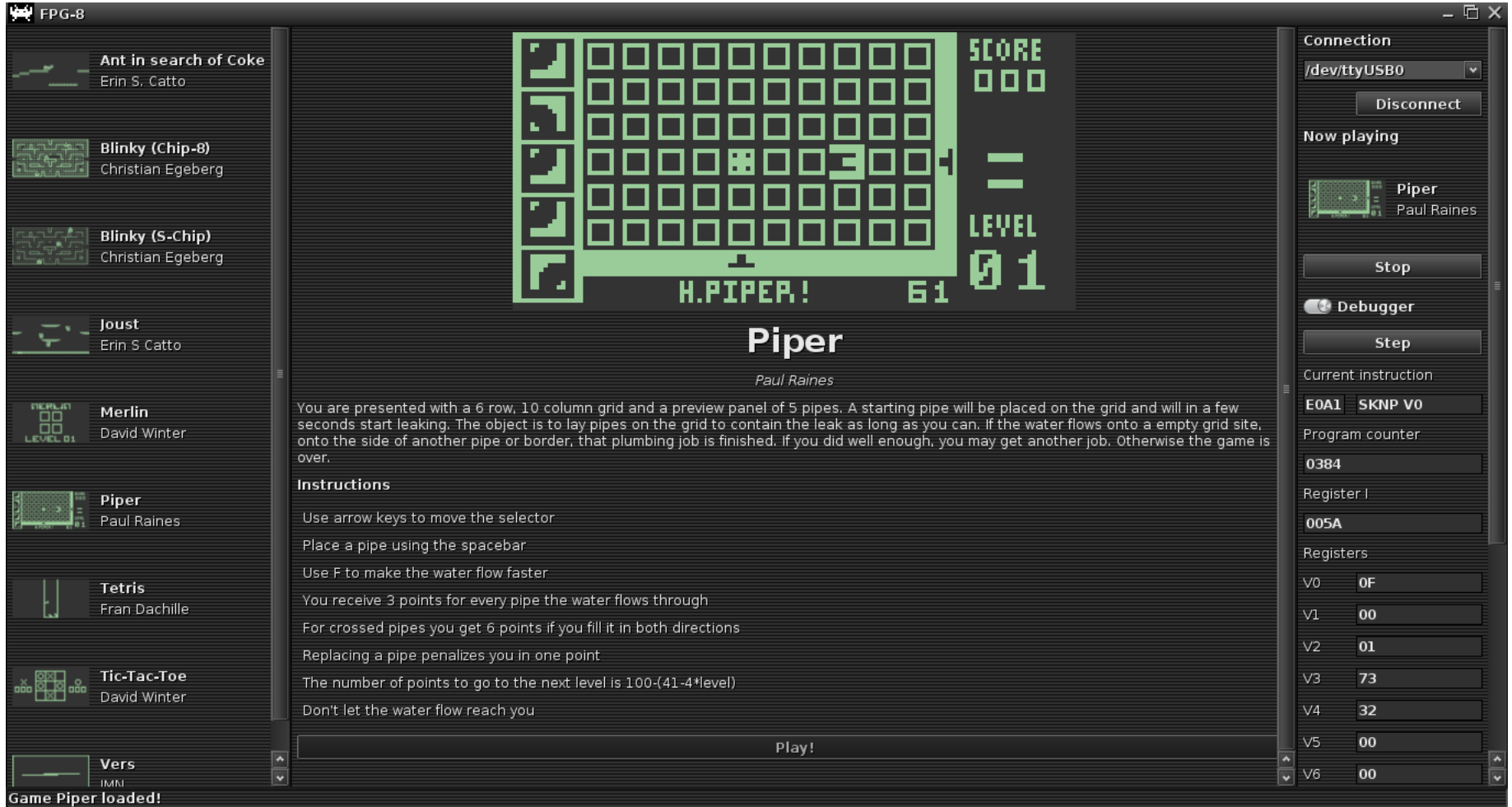
# Serial Interface Protocol

---

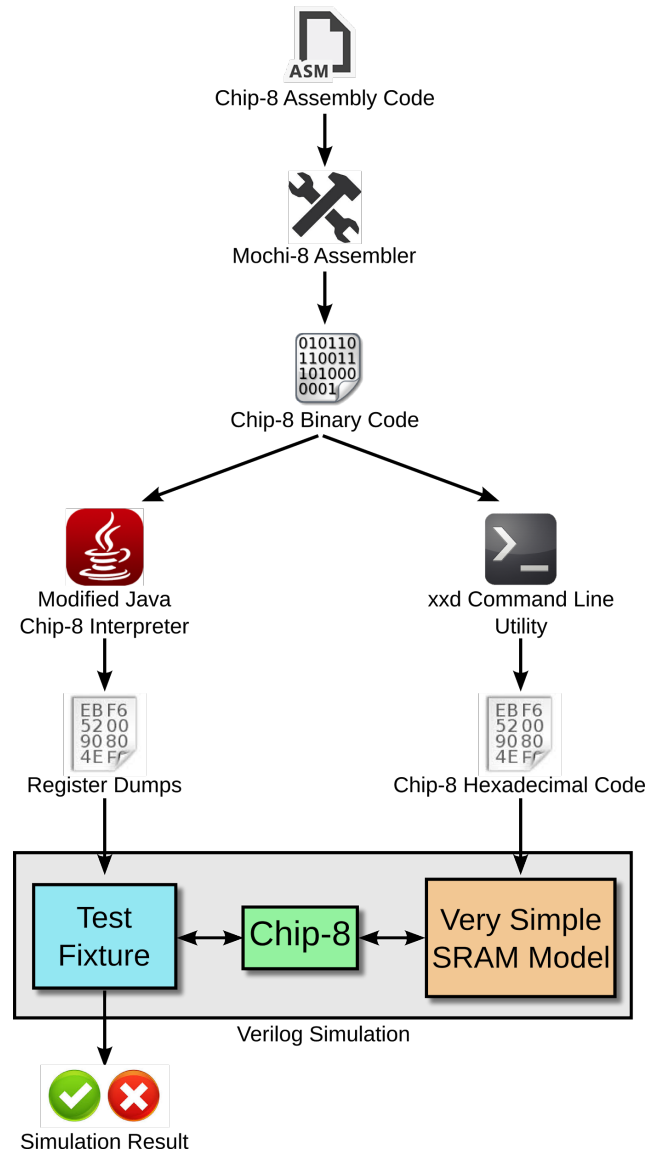
- Load a game into the SRAM
- Start/stop the processor
- Debug mode
  - Enable/disable debug mode
  - Step to the next instruction
  - Read registers
- Provide input
  - Key pressed/key released commands
  - Very inconvenient



# Java Application



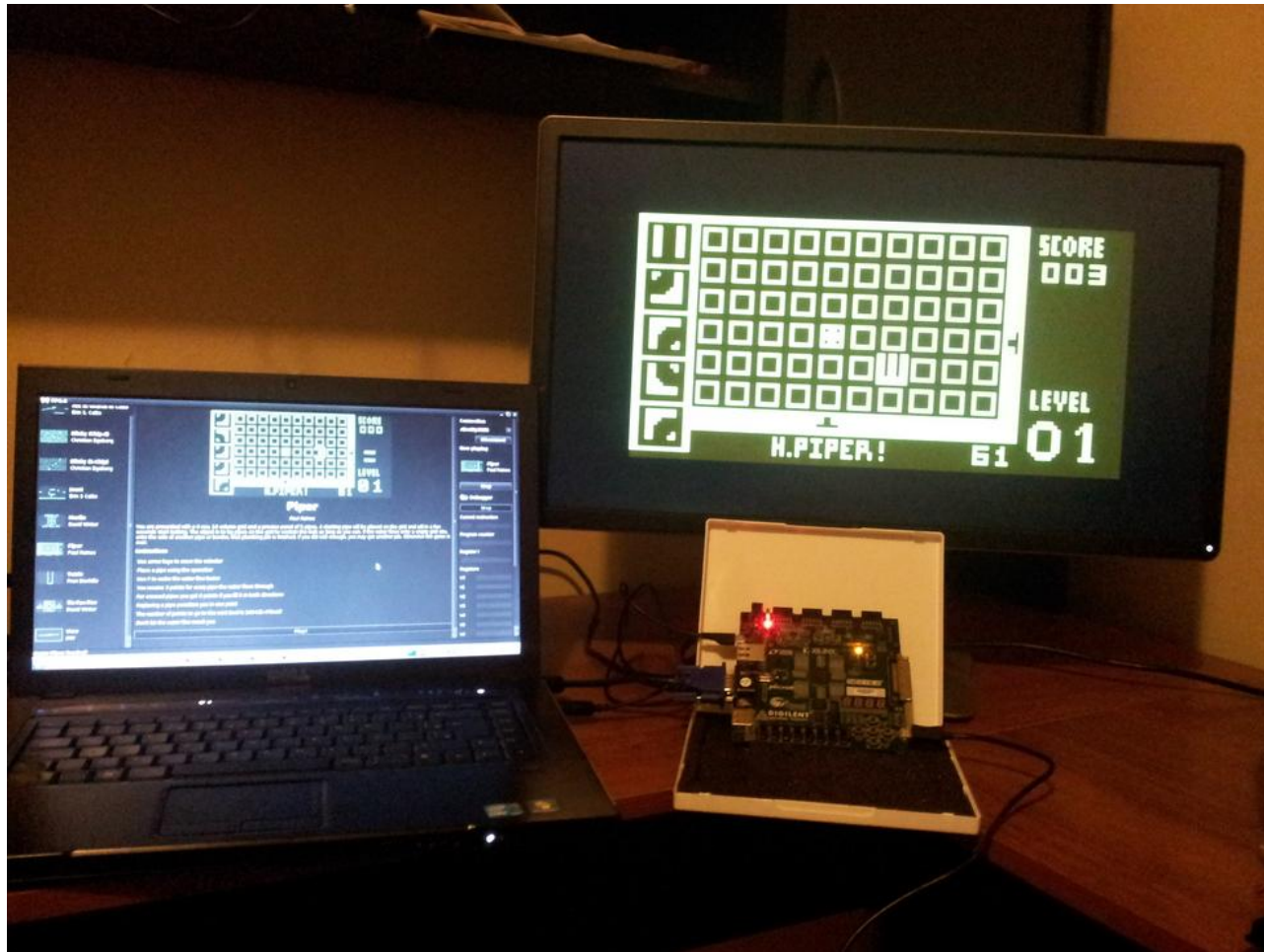
# Testing





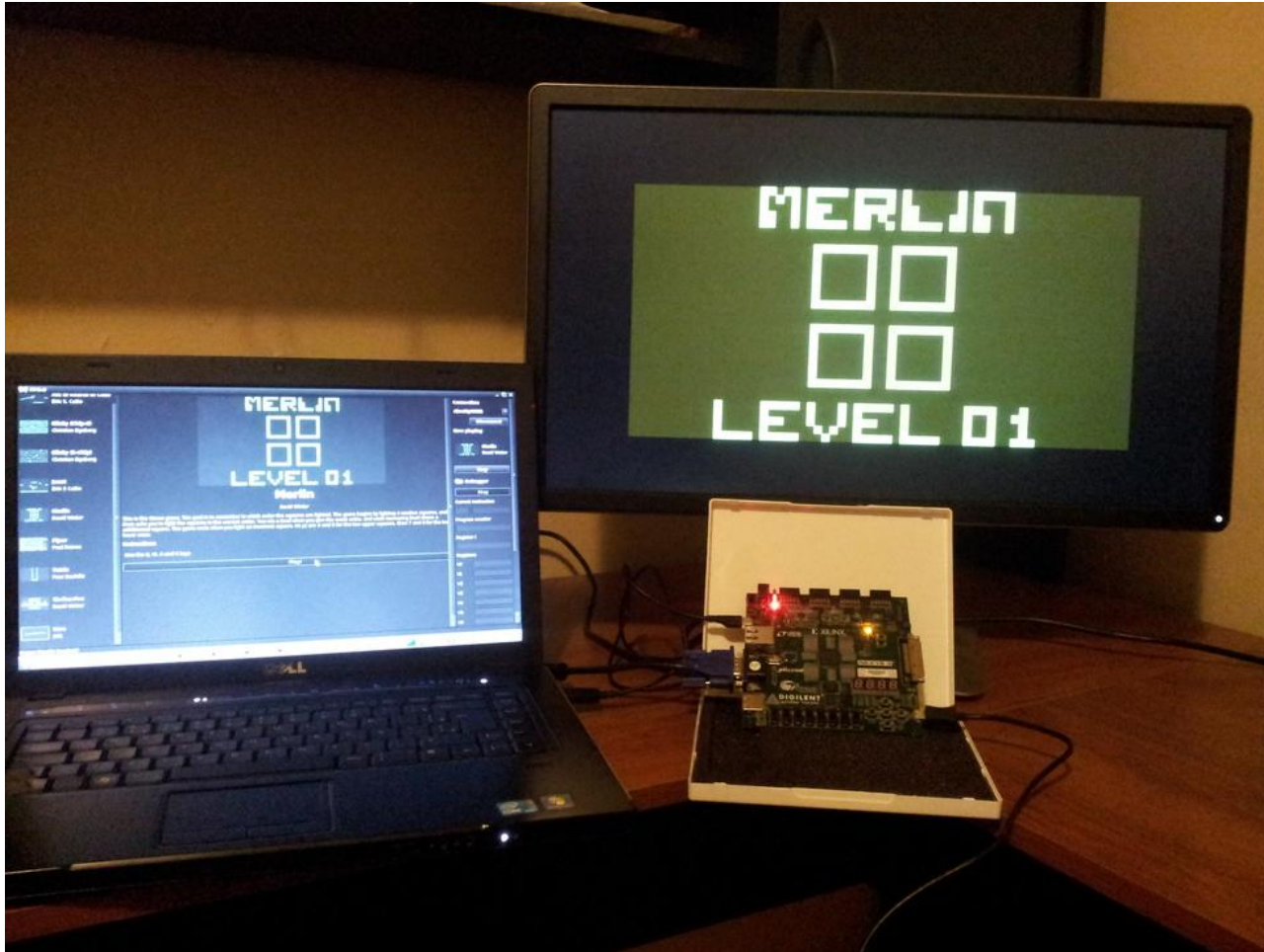
# Final Result

- H. Piper



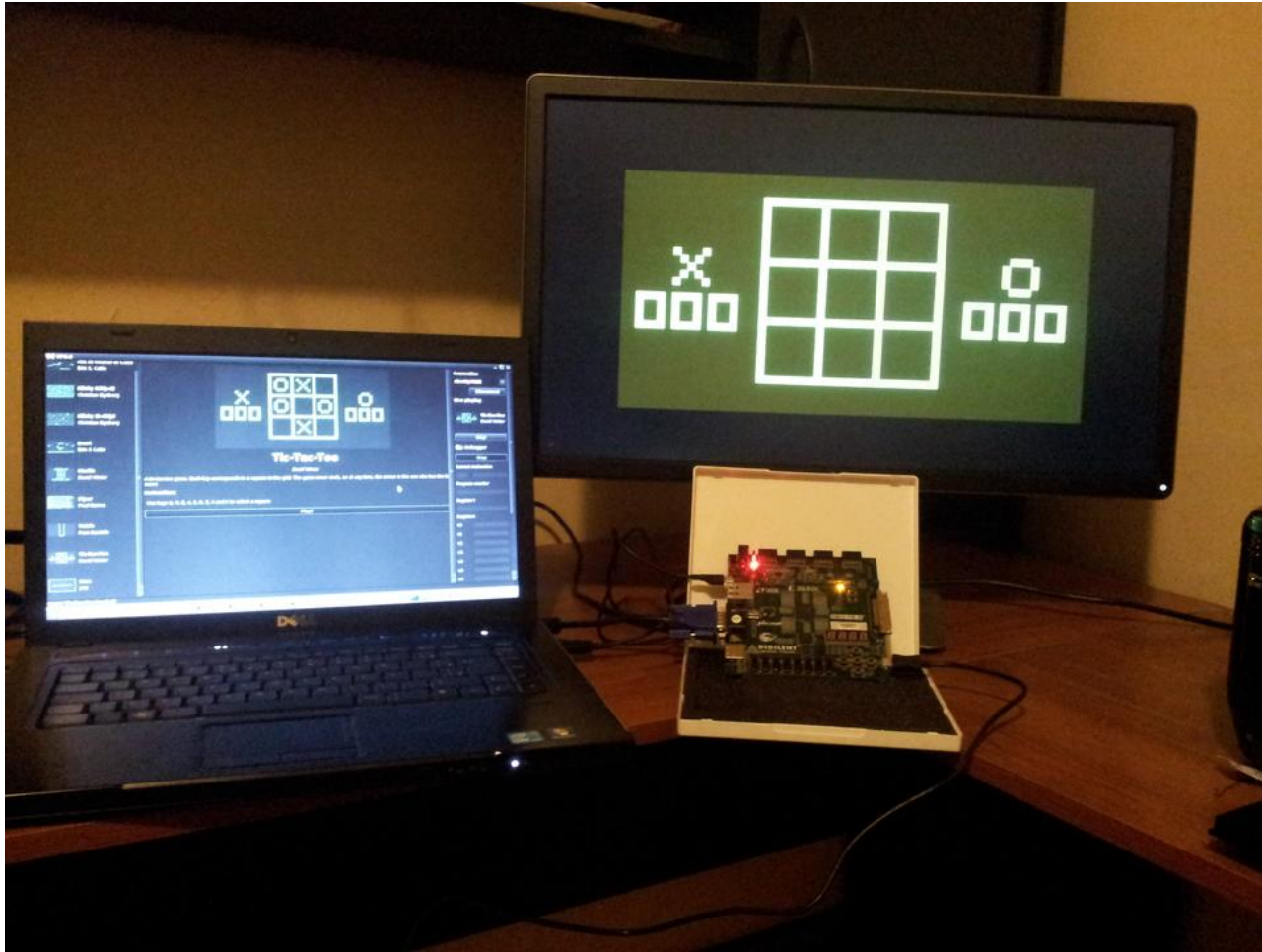
# Final Result

- Merlin



# Final Result

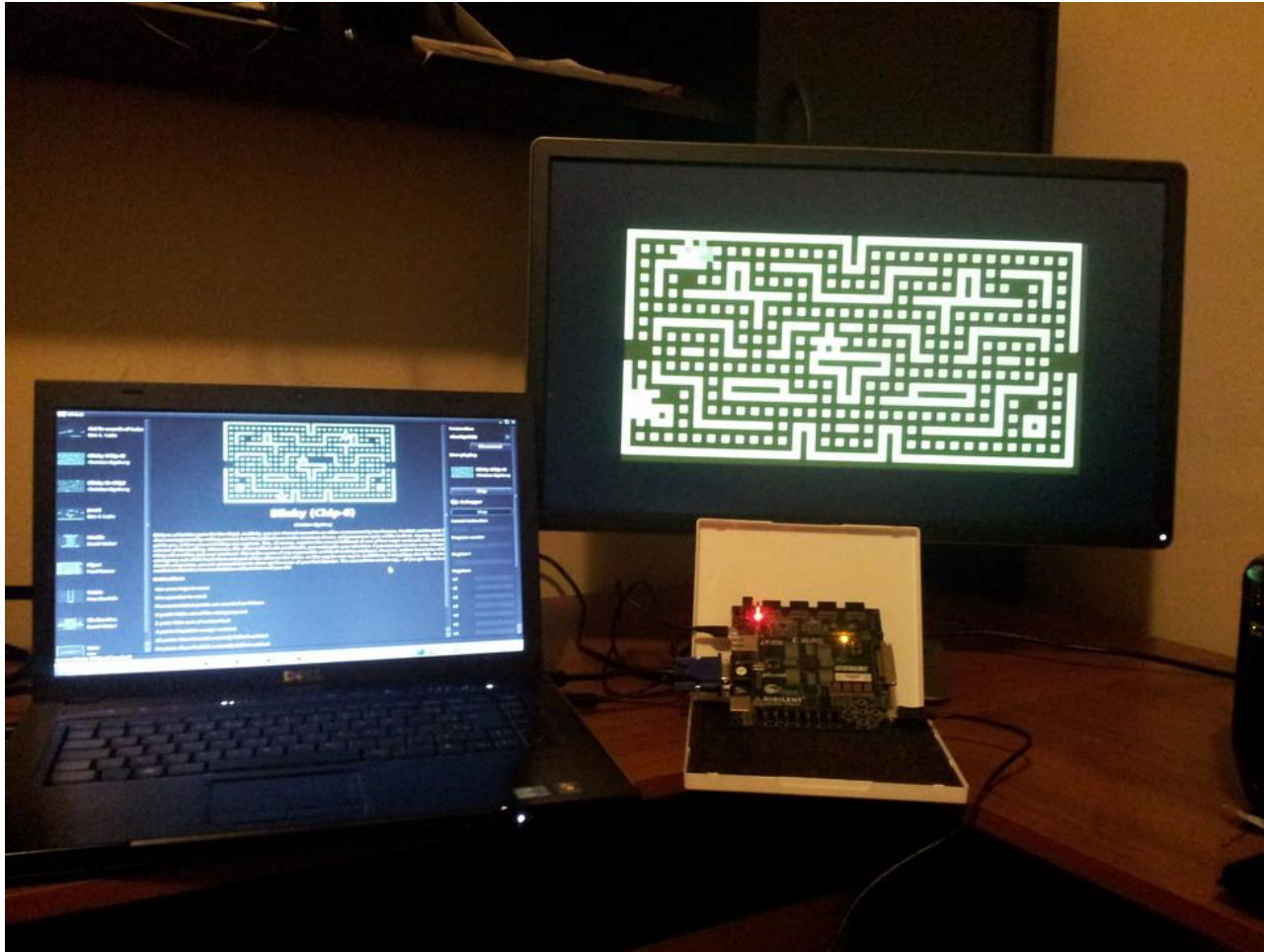
- Tic-Tac-Toe





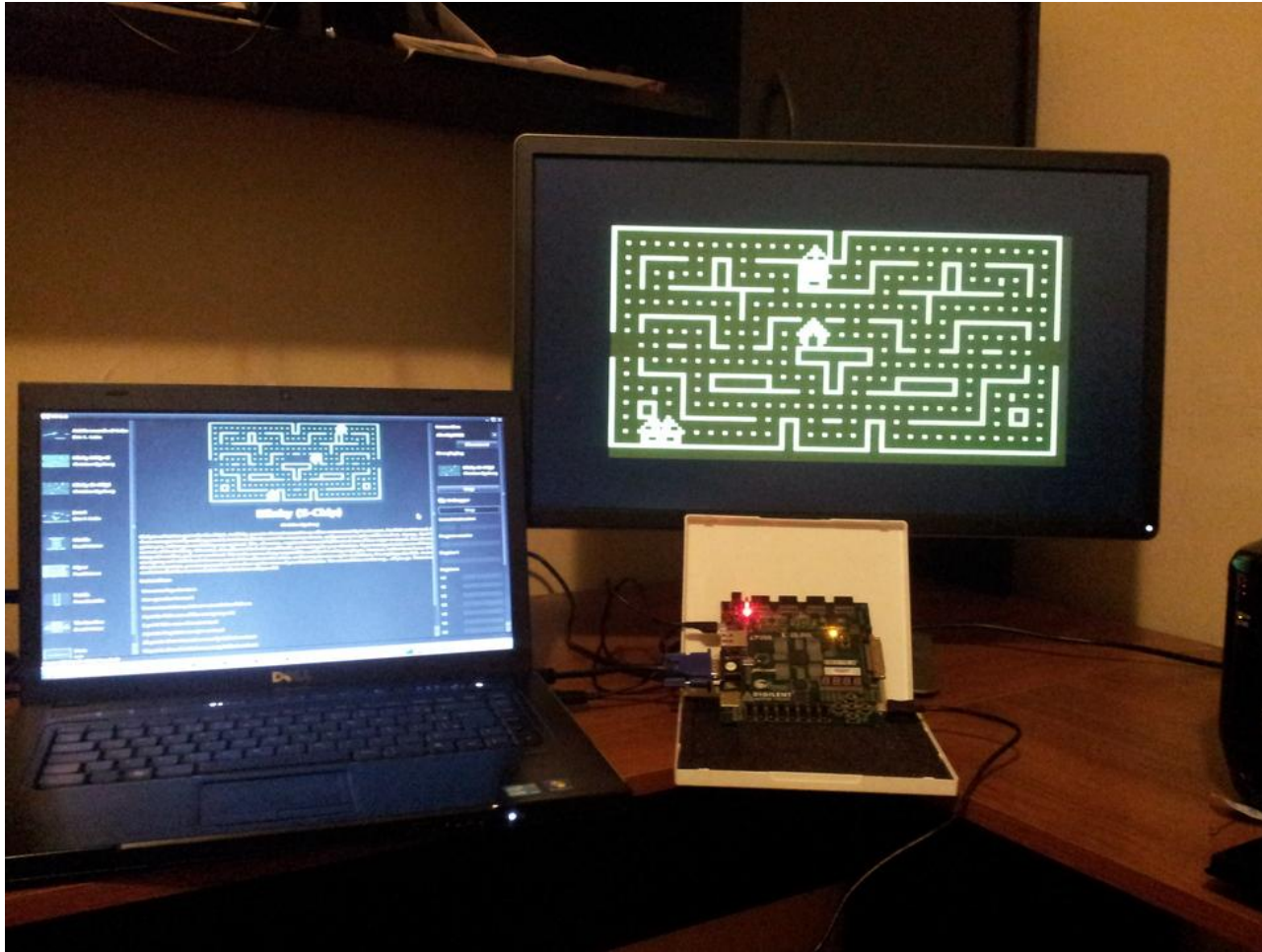
# Final Result

- Blinky (Chip-8)



# Final Result

- Blinky (S-Chip)



# Status

---

- Project is finished

**Fork me on GitHub!**  
[github.com/guimeira/fpg8](https://github.com/guimeira/fpg8)



# Questions?

---



THANK  
YOU!

