How do you write an emulator anyway?
Whoami

- Anisse Astier
- 🐦@Aissen@treehouse.systems
- https://anisse.astier.eu
- Writing gears, a Game Gear emulator
- Not an emulation expert
- Presented Z80 : the last secrets in 2022 and WASM101 yesterday
Why this presentation

• Will :
  – Give pointers
  – Help you start

• Not :
  – Exhaustive
  – Why

• Focus on simpler (8bits) platforms
What is an emulator?

- A software program
- To run software from another computer
What is an emulator?
What is an emulator?

• Spectrum of emulation and accuracy
  – One software
  – All software
  – Clock accurate
A crazy example
Where to start?

- Pick a target
- Pick a host platform
- Make sure you have time
Where to start?

• Start simple: one CPU instruction
  – Debug code, minimal disassembler: can you recognize the instruction?
  – Execution: run it, model CPU state and change it
  – Test: verify state change

• Learn CPU concepts as you go
Interesting CPU concepts

- State: registers
- Instruction
  - Assembly
  - Encoding
- Interrupts
- Memory access
STRUCTURE
Emulator structure

• Examples of 8-bit consoles
  - NES, SMS, GB, Atari 2600, Game Gear

• Various parts :
  - CPU
  - Display
  - Sound
  - Memory
  - ROM mapping
  - Input
Structure: tricks

- Devices: simple code boundary
- CPU: you are allowed to optimize a bit in advance (no allocation, jump tables)
- Vertical slice: common advice
  - Do what works for you
Structure: tricks

- You will need a debugger
- Build tooling early
- Or use already existing
Finding documentation

- Game boy: gbdev.io, Gekkio’s CTR

Game Boy: Complete Technical Reference

gekkio

https://gekkio.fi

January 7, 2024
Revision 135
Finding documentation

- SMS
- Game Gear
Finding documentation

Hardware Reference Manual
for the
SEGA Game Gear Console

The Undocumented Z80 Documented
Sean Young
Version 0.91, 18th September, 2005

Z80 Microprocessors

Z80 CPU

User Manual
UM008011-0816
Finding documentation

- Do web searches with technical terms. Ex: search for `<sound chip name>` (SN76498/PSG) instead of `<console sound>`
Device I/O

- Used to be almost as simple as writing to a memory address
- Dedicated instructions

Figure 7. Input or Output Cycles
Displaying something and making sound

• Understand your host platform first

• Hello world of video
  – Display a pixel buffer, change it 60 times a second

• Hello world of sound
  – Play a sine or square wave

• Nothing emulator specific
Graphics emulation

- Needs hardware understanding
- VDP example
  - VRAM/Regs I/O
  - Two planes: BG and sprites
  - BG is on torus, scrollable
Graphics emulation

- VDP example (cont)
  - Sprites are limited
  - Color encoding
  - Sprite pixel encoding
- Sync strategy: line
Sound emulation

- Needs hardware understanding

- PSG example
  - Regs I/O
  - Channels
  - Tones
    - Counter
  - Noise
    - Random/LFSR
    - Periodic
Sound emulation

• Start simple
  - Square wave
• Example: counter, not frequency

• Example: playing samples
• Sync strategy: cycles
Emulator testing: CPU

- Unit test components using existing test test suites.
- Examples for Z80
  - Zexall
  - Z80test
  - Fuse test suite
Emulator testing: audio

• Sound:
  − Listen to music, needs a good ear
  − FFTs
  − Can you hear samples (SEGA sound)
Emulator testing: roms and display

- GG test suite, SMS test suite, inspired by 240p
Emulator testing: roms and display

- Game boy examples:
  - dmg-acid2 ppu test
  - Blargg’s gb tests
  - Mooneye test suite
Emulator testing: frame generation

• Integration testing: automate frame generation
  – Everything is deterministic
• Compare with other (good) emulators
• Use real hardware
  – Tip: use flash carts
• Try lots of software/games/roms
Emulator testing: frame generation
Summary

- Pick platforms
- Start small
- Read lots of documentation
- Test
- Write and talk about it
Thank you! Questions?
Links

• My emulator, gears: https://github.com/anisse/gears
• gears in the browser: https://anisse.github.io/gears
• Romhack to generate level screens: https://gist.github.com/anisse/c6e4101236708890381414f48804201b
• Linux on scratch: https://turbowarp.org/892602496/
• https://github.com/gbdev/awesome-gbdev#emulator-development
• SMS test suite https://github.com/sverx/SMSTestSuite
• GG test suite https://github.com/sverx/GGTestSuite/
• Game boy test roms compilation: https://github.com/c-sp/gameboy-test-roms
Sources

- Screenshots from Sameboy https://sameboy.github.io/
- BGB: https://bgb.bircd.org/shots.html
- dmg-acid2 image: https://github.com/mattcurrie/dmg-acid2
- Z80test result on specemu: https://www.spectrumcomputing.co.uk/forums/viewtopic.php?t=752