

# GameBoy-Emulator

最后修改 2018/6/27

## 目录

<a href="#">使用方法</a>	1
<a href="#">项目简介</a>	2
<a href="#">BUG(feature)</a>	3
<a href="#">参考资料</a>	5
<a href="#">(附)加分项</a>	7

## 使用方法

打开程序后-文件-打开-选择 ROM 即可。

其他的选项都是假的。

点击关于可以看到款爷的头像。

Z-B

X-A

上下左右是方向键

Enter-Start

Space-Select

需要 vc++ 2017 或更高版本的运行库(release 中的程序采用了/MT, 可以直接运行)

工程加载需要 vs2017 或更高版本, 安装 win32 桌面开发的相关组件。

gbcore 文件夹:模拟器核心代码

win32 文件夹:win32 相关代码

testrom 文件夹: 一些测试 rom,分别是:显示一张图片; 显示一些精灵; 开机程序; 显示复杂的图像和精灵的组合, 响应按键; 任天堂官方测试, 测试简单的显示, 按键, 窗口, 音效; 小游戏; 大游戏(需要 MBC1)支持; CPU 测试。

## 项目简介

本项目实现的基础功能：

- 实现 Gameboy Z80 CPU 模拟（支持所有或大部分指令集）
- 实现时钟模拟
- 实现内存模拟
- 支持基本图形操作
- 支持对游戏进行交互操作（即输入）
- 支持载入 ROM
- 可以基本玩一款 GB 游戏

基础功能以外的可选功能：

- Background Window Graphic
- Sprite
- 调色板
- 卡带（MBC1,2,3）
- Serial IO

本项目由以下主要部分组成：

### CPU

模拟器的计算核心，负责执行指令，计数时间，响应中断

### MMU

管理所有的存储器（事实上，更好的处理方式是“管理所有的存储器通信”，而不是“管理所有存储器”，后者会为进一步拓展带来困难），管理手柄，串行 IO，以及卡带部分。

### Timer

负责计算时间，需要时发生中断，保持组件同步（逻辑上是这样，事实上没有可让他同步的）

### GPU

负责生成图像，需要时发送中断

### Window

负责实际显示图像，与玩家进行交互

joypad(直接写入 MMU，只包括几个元素和三个函数)

负责模拟手柄电路芯片的行为

### Cart(嵌入于 MMU)

卡带的抽象实现，目前支持 MBC123，应该满足绝大部分游戏的运行需要。

### MBC(嵌入于 Cart)

负责卡带内容读取的管理，MBC3 还附带了一个外部时钟。

### SerialIO(嵌入于 MMU)

负责串行接口通讯的模拟，目前没有实装通讯功能，假定一直处于脱机。

## Known BUG

目前本项目仍然存在一些 BUG:

- LCD 显示位有时无法正确复位, 比如波斯王子, 哆啦 A 梦 惑星传说  
具体的表现是在游戏主动关闭 LCD 显示后无法将其再度开启, 原因不明。
- 某些游戏中采用的特殊显示效果无效(大概算是修复了一半吧....)
- 窗口的显示有时会带有一些奇怪的效果, 比如缺失一条线或者是胡乱显示一条线。
- 特殊的专用测试 ROM 仍然大部分无法通过
- LYC 中断的进行有问题
- 似乎有其他的 SerialIO 工作模式(??? )
- 时钟由于奇特原因无法完美合拍, 为什么 CPU 执行指令的时间还和顺序有关了? Z80 没有流水线吧?

ie: 在 bgb 模拟器上的数据:

运行程序: DMG\_BOOT

	据文档应消耗	实消耗
ld	12	4
cp	8	8
jr(跳回 ld)	12	20
总计	32	32

PC=0x101 时

bgb: DIV=AB line=00 cnt=28

Mr.Buck GB:DIV=EF

offset ~ 40664 clk~0.01067 s

但是貌似没有严重影响, 连续玩了一个半小时的星之卡比 2 工作良好。

- 另注: 所有的存档文件都扔在了 ROM 同目录下。

本项目的一些缺陷:

- 没有在硬件上做到完全仿真, 仍使用了不少近似实现, 并且有不少细节尚未完成。
- fps 并对于不同的游戏并不能稳定在同一个值上, 视游戏的复杂程度, 平均帧率会在 60-80 之间波动, 因此对于那些自己统计你游戏时间的游戏, 这个值绝大多数情况下都会虚高。
- 由于使用的单线程实现, 假如游戏进入死循环(例如使用了奇怪的 ROM, 或者是发现了新的恶性 BUG), 整个程序都会卡死。
- MBC3 的外部时钟尚未完全测试, 对于所有情况下的工作正常尚不能保证(但看上去没问题)

项目编写过程

难点:

0. 代码体系构建

1 对于每一硬件部分作用的理解

2 中断的概念

3 DEBUG 相对繁琐困难, 作为辅助的 NOGMB 在我的电脑上经常崩溃, CPU 部分由于指令太

多，没有掌握编写技巧，也较容易写错，而且错误不易发觉。

4 对于类的封装，与显示部分实现对接

5（并没有实现）音频输出部分组件繁多，依赖关系复杂。

踩过的坑：

1.项目里附带的文档在 `cpu` 指令关键部分有遗漏和错误，比如 `BIT` 系列指令事实上有 128 条，然而只列出了 16 条。在个别描述上语义不清，带来了一定的理解困难。

2.附带的 `js` 版本教程中对 `timer` 的代码部分有错误且不易理解，`timer` 写错后产生的错误十分严重，但原因不易发现。（比如俄罗斯方块只出长条）`joypad` 部分也有错误。

2.5.误判过不少错误原因，白白浪费了许多时间。

3.网上找的测试 ROM 有的竟然需要使用 `CGB` 专用的寄存器。。。

4.低位优先和栈的配合不当曾经引起过不少问题。

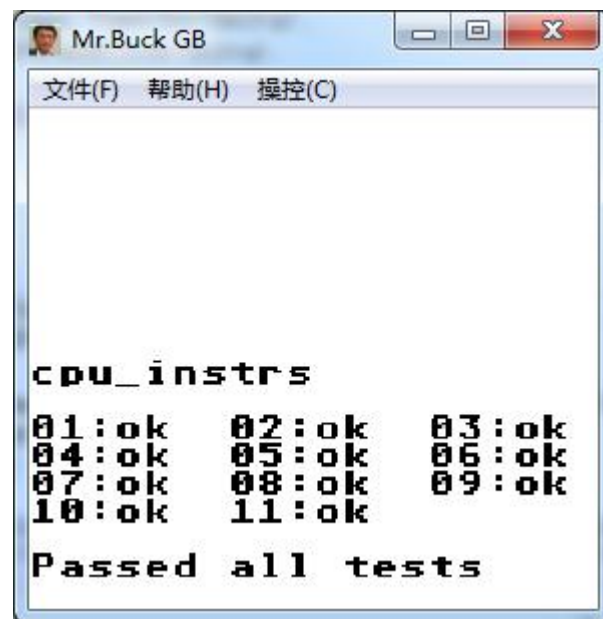
5.没有注意到 `CPU` 频率与其余硬件工作周期是完全对应的，绕过不少弯子。

6.`js` 版本的教程中，给的 `Boot` 是有错的，倒数第三行末尾，`0x4C` 应为 `0x3C`。//为什么要这么对我？我不值得你们去爱吗？

DEBUG 过程

逐指令输出寄存器信息，与 `BGB` 模拟器进行比对。

在大错误基本排除后运行测试 ROM，辅助调试。



## 参考资料

- GameBoy 手册

<http://marc.rawer.de/Gameboy/Docs/GBCPUman.pdf> 看下边这个更好

Gameboy “pandoc”

<http://bgb.bircd.org/pandocs.htm>

- Gameboy Emulation Tutorial (JavaScript)

<http://imrannazar.com/GameBoy-Emulation-in-JavaScript> 哥们凉了

timer 写错了 ( ? )。其他部分写的还不错，GPU 部分与其他人的有点出入，但是问题应该不大，没有 B&WGraphic 部分。

一个博客，包含了不少有用的信息

<http://blog.rekawek.eu/2017/02/09/coffee-gb/>

<https://github.com/trekawek/coffee-gb>

github 上的一个项目和它附的参考资料之一 用的 qt

<https://github.com/brNX/gameboy-emu>

<http://www.codeslinger.co.uk/pages/projects/gameboy/interrupts.html>

- NO\$GMB (Gameboy Emulator / Debugger)

<http://problemkaputt.de/gmb.htm> 不好用到飞起，还会崩溃

brb ,an advanced Gameboy Debugger

<http://gbdev.gg8.se/> 赞这个

- Start Gameboy Programming

<http://www.loirak.com/gameboy/gbatutor.php>

<http://www.devrs.com/gb/> 这个更好，但是内容对本项目其实基本没有帮助 ( ? )

GameBoyWiki，想查阅卡带和音频输出的详细信息来这里，还拥有一个文件服务器，里面可以找到专用的测试 ROM 文件，brb 调试器也可以在这里找到

<http://gbdev.gg8.se/wiki>

XnView，以及 ppm 图像格式的介绍

没有写 GUI 时的临时替代品，用来看看图像对不对

WinHex，二进制文件浏览编辑器

百度就有

测试过的 ROM:

可以正常玩的

坦克大战

俄罗斯方块

星之卡比 1,2

超级马里奥大陆 1,2

恶魔城系列

瓦里奥大陆 1

谍对谍

魂斗罗

游戏王

国际象棋

吞食天地

口袋妖怪红(汉化版简直神奇，不过也不要难为人家了....)

有问题但也一定程度上能玩的

波斯王子

没法玩的

非 mbc1，mbc2，mbc3 的游戏，在我的设置下会直接闪退。

哆啦 a 梦惑星传说 (LCD)

附:加分项(申请)

- 项目目录结构良好、清晰;
- 没有内存泄漏;
- (助教自行通过界面玩耍程序时) 程序很少崩溃;
- 其他你认为可以作为加分项的亮点:  
脸皮厚