Computer altern gut - mit Open Source

Die letzten 50 Jahre haben viele verschiedene Computerarchitekturen und -designs hervorgebracht.

Der größte Teil davon ist heute nicht mehr erhältlich. Sind diese Rechner und alle Programme dafür bald für immer verloren?

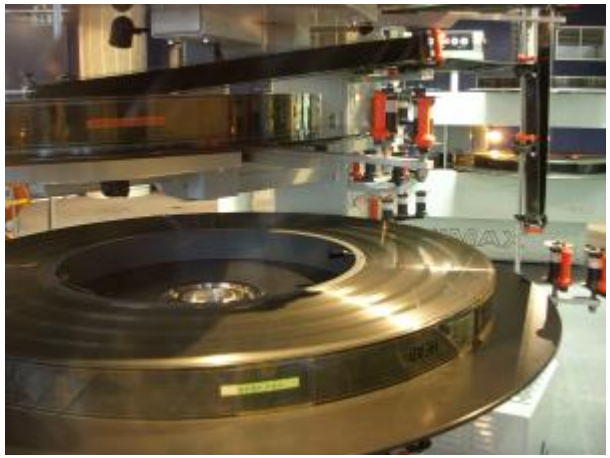
Am Beispiel älterer Spielekonsolen soll gezeigt werden, wie der Erhalt möglich ist und wie wichtig offene Soft- und Hardware für die Konservierung ist.

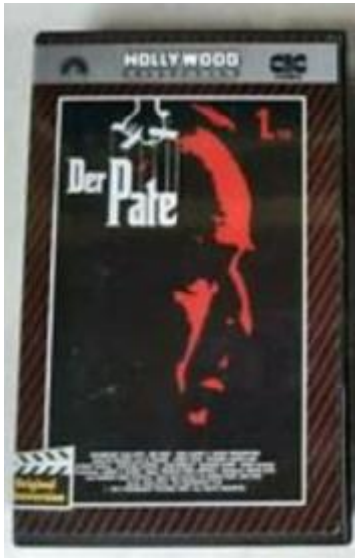
Computer altern gut - mit Open Source

1. Erhalten, aber warum?
2. Erhalten, aber wie?
3. Einsetzen, aber wo?
4. Erneuern, aber womit?
5. Ergebnis

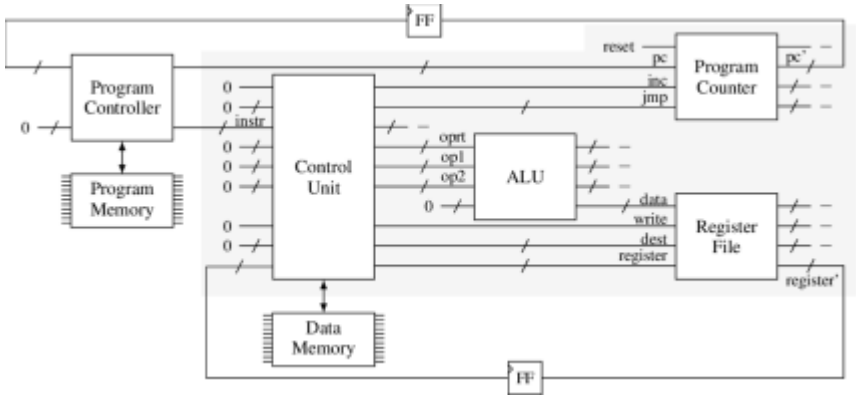
github.com/RobertPeip

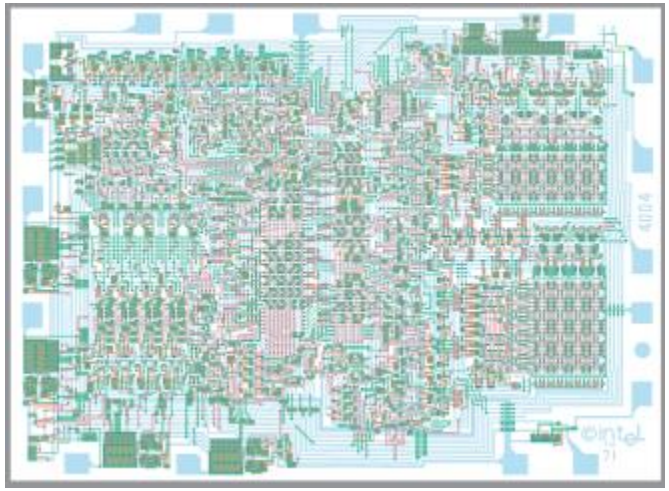
Erhaltenswert?



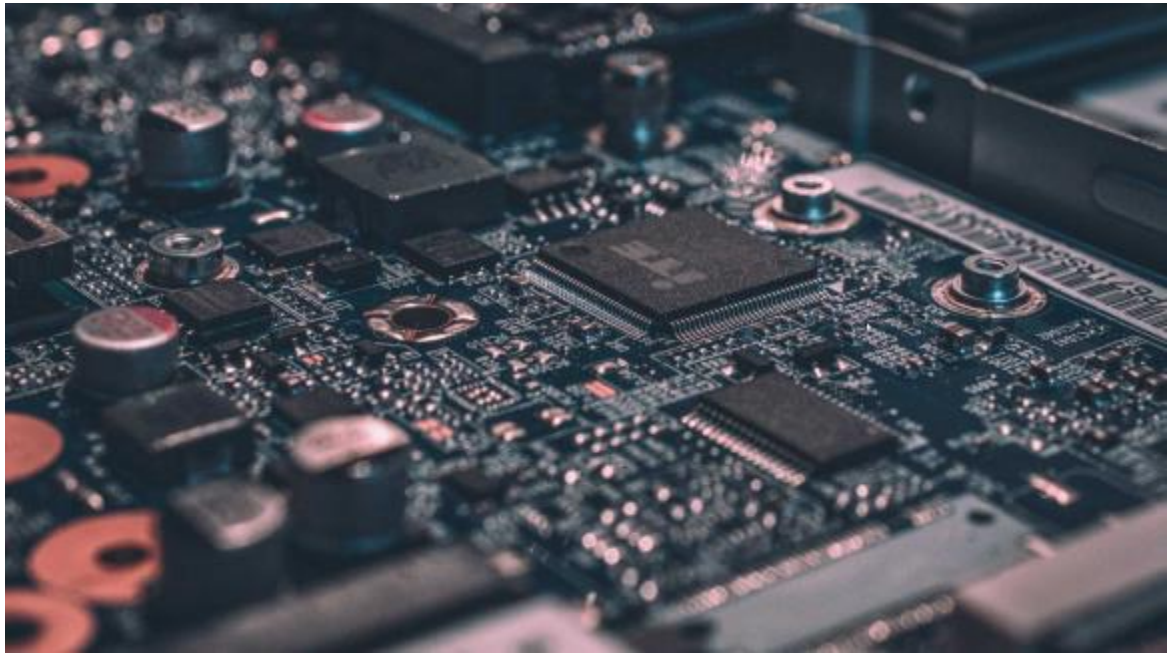


Erhaltenswert?





Technik vs. Recht



Software

Information

Dieser Inhalt ist nicht mehr im Nintendo eShop erhältlich.

Eigenschaften von Tetris (World) (Rev A).gb

Allgemein | Sicherheit | Details | Vorgängerversionen

Tetris (World) (Rev A).gb

Dateityp:	GB-Datei (.gb)
Öffnen mit:	Unbekannte Anwendung — Ändern...
Geändert:	Dienstag, 24. Dezember 1996, 22:32:00
Größe:	32.0 KB (32.768 Bytes)

Gerät

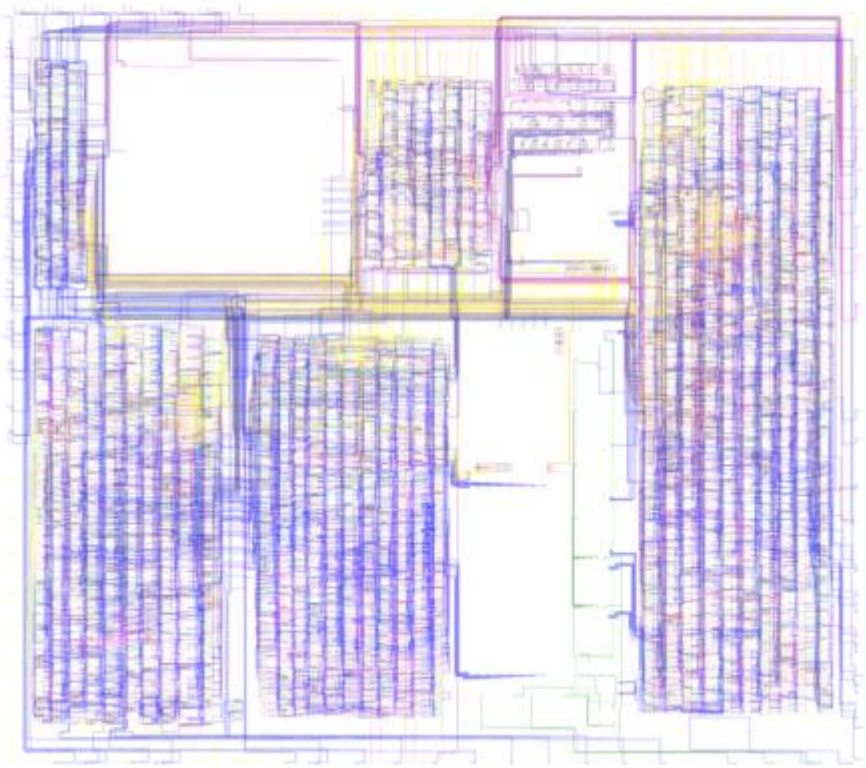

Nintendo GAME BOY
SELECT
START
TETRIS


Buttons and screws
included

Interne Schaltung

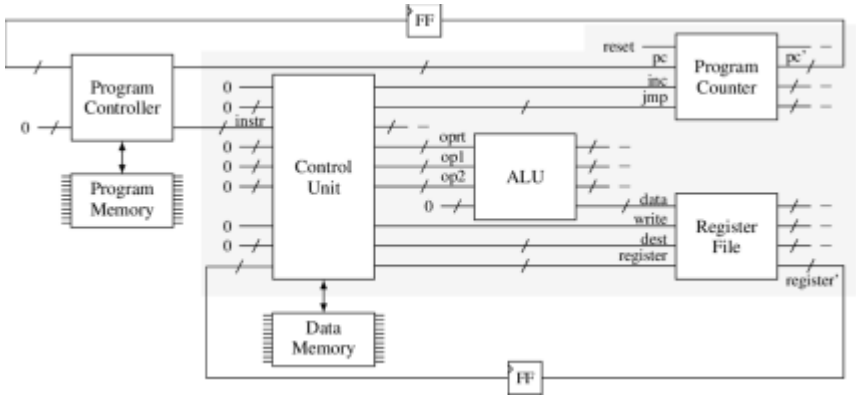
DMG-CPU-Inside

This repo contains an annotated overlay for the Nintendo Game Boy DMG-CPU-B chip die and the extracted schematics.



furrtek Furrtek

Funktionsweise



gameboy emulator

2,116 repository results

Languages

C++	489
C	371
Rust	327
JavaScript	128
C#	110
Go	108
Java	102
Python	81
Assembly	49
TypeScript	42

Emulation

“Als Emulator wird in der Computertechnik ein System bezeichnet, das ein anderes in bestimmten Teilaspekten nachbildet.”

Wikipedia

Software Emulator

Implementierung der **Funktionsweise** aller Einzelkomponenten in Programmcode

Pseudoparallele Ausführung aller Einzelkomponenten

Für ausgeführte Software nicht unterscheidbar von der Ausführung auf dem ursprünglichem Gerät

Für den Nutzer nicht unterscheidbar von der Ausführung auf dem ursprünglichem Gerät

Hardware Emulator

Implementierung der **Funktionsweise** oder der **Schaltung** aller Einzelkomponenten in Hardwarebeschreibungssprache

Parallele Ausführung aller Einzelkomponenten

Für ausgeführte Software nicht unterscheidbar von der Ausführung auf dem ursprünglichem Gerät

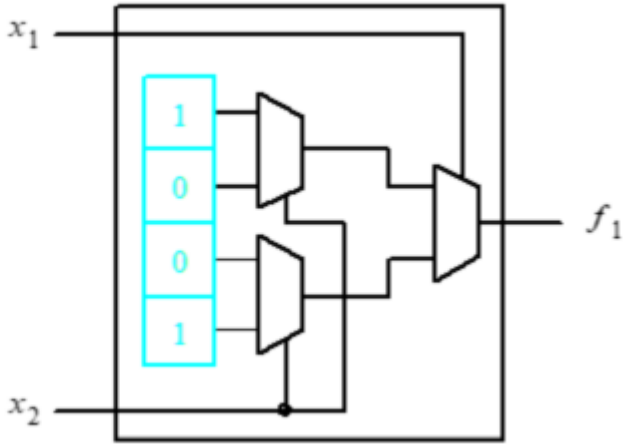
Für den Nutzer nicht unterscheidbar von der Ausführung auf dem ursprünglichem Gerät

FPGA

Field Programmable Gate Array

- 1.000 – 5.000.000 Look up Tables
- können jede digitale Funktion nachbilden
- Berechnung erfolgt parallel
- beliebig verschaltbar
- RAM basiert: beliebig oft neu verdrahtbar

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1



Warum überhaupt in Hardware?

Original läuft auch in Hardware

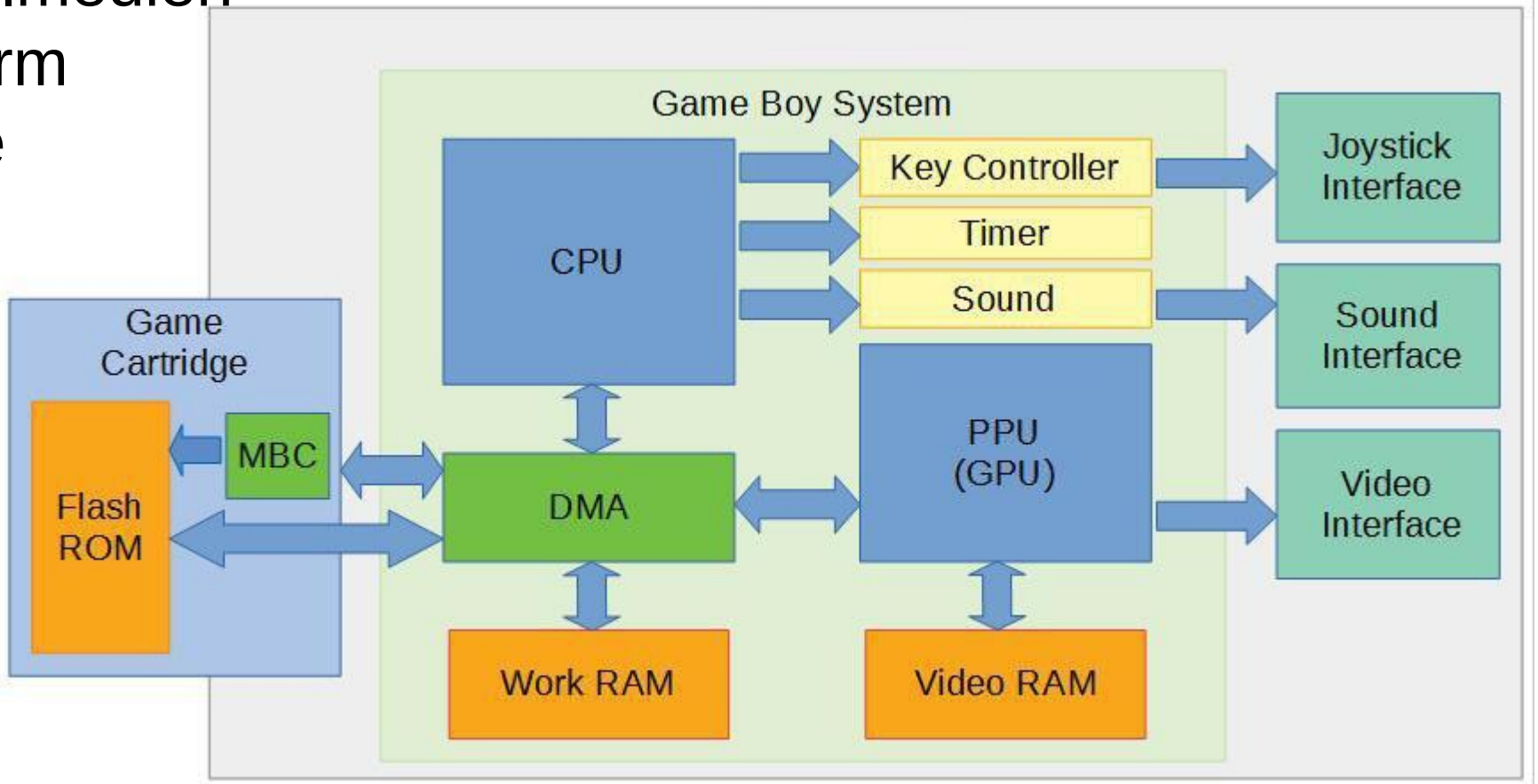
Parallel Ausführung – keine Kompromisse nötig

Geringe Latenzen (Ein- und Ausgabe)

Geringe Stromaufnahme

Komponenten

- Prozessor (CPU)
- Grafik (PPU)
- Sound
- Arbeitsspeicher (RAM)
- Wechselmedien
- Bildschirm
- Eingabe



CPU in Software

Interpreter

Nachbau jeder Instruktion durch Software Beschreibung

Instruction	Instruction Code	Operation
NOP	0000 0000	No operation
LDAC	0000 0001 Γ	$AC \leftarrow M[\Gamma]$
STAC	0000 0010 Γ	$M[\Gamma] \leftarrow AC$
MVAC	0000 0011	$R \leftarrow AC$
MOVR	0000 0100	$AC \leftarrow R$
JUMP	0000 0101 Γ	GOTO Γ
JMPZ	0000 0110 Γ	IF (Z=1) THEN GOTO Γ
JPNZ	0000 0111 Γ	IF (Z=0) THEN GOTO Γ
ADD	0000 1000	$AC \leftarrow AC + R$, IF $(AC + R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
SUB	0000 1001	$AC \leftarrow AC - R$, IF $(AC - R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
INAC	0000 1010	$AC \leftarrow AC + 1$, IF $(AC + 1 = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
CLAC	0000 1011	$AC \leftarrow 0$, $Z \leftarrow 1$
AND	0000 1100	$AC \leftarrow AC \wedge R$, IF $(AC \wedge R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
OR	0000 1101	$AC \leftarrow AC \vee R$, IF $(AC \vee R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
XOR	0000 1110	$AC \leftarrow AC \oplus R$, IF $(AC \oplus R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
NOT	0000 1111	$AC \leftarrow AC'$, IF $(AC' = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$

Recompiler (dynamisch)

Übersetzung jeder Quell-Instruktion in eine oder mehrere Ziel-Instruktionen

Normal instructions				Compressed	
00000	SUB	10000	CMP	000	SUB
00001	AND	10001	TEST	001	AND
00010	ADD	10010	LW	010	ADD
00011	OR	10011	SW	011	CMP
00100	XOR	10100	LH	100	LW
00101	LSR	10101	SH	101	SW
00110	LSL	10110	LB	110	LDI
00111	ASR	10111	SB	111	MOV
01000	BREV	11000	LDI	Reserved for FPU	
01001	LDILO	11001			
01010	MPYUHI	Special Insn		11010	FPADD
01011	MPYSHI			11011	FPSUB
01100	MPY	11100	BREAK	11100	FPMPY
01101	MOV	11101	LOCK	11101	FPDIV
01110	DIVU	11110	SIM	11110	FPI2F
01111	DIVS	11111	NOOP	11111	FPF2I

CPU in Hardware

Verhalten

Schaltung

Instruction	Instruction Code	Operation
NOP	0000 0000	No operation
LDAC	0000 0001 Γ	$AC \leftarrow M[\Gamma]$
STAC	0000 0010 Γ	$M[\Gamma] \leftarrow AC$
MVAC	0000 0011	$R \leftarrow AC$
MOVR	0000 0100	$AC \leftarrow R$
JUMP	0000 0101 Γ	GOTO Γ
JMPZ	0000 0110 Γ	IF (Z=1) THEN GOTO Γ
JPNZ	0000 0111 Γ	IF (Z=0) THEN GOTO Γ
ADD	0000 1000	$AC \leftarrow AC + R$, IF $(AC + R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
SUB	0000 1001	$AC \leftarrow AC - R$, IF $(AC - R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
INAC	0000 1010	$AC \leftarrow AC + 1$, IF $(AC + 1 = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
CLAC	0000 1011	$AC \leftarrow 0$, $Z \leftarrow 1$
AND	0000 1100	$AC \leftarrow AC \wedge R$, IF $(AC \wedge R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
OR	0000 1101	$AC \leftarrow AC \vee R$, IF $(AC \vee R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
XOR	0000 1110	$AC \leftarrow AC \oplus R$, IF $(AC \oplus R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
NOT	0000 1111	$AC \leftarrow AC'$, IF $(AC' = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$

menschenlesbar
leichter wartbar

Netzliste
an Architektur angepasst

Instruction	Instruction Code	Operation
NOP	0000 0000	No operation
LDAC	0000 0001 Γ	$AC \leftarrow M[\Gamma]$
STAC	0000 0010 Γ	$M[\Gamma] \leftarrow AC$
MVAC	0000 0011	$R \leftarrow AC$
MOVR	0000 0100	$AC \leftarrow R$
JUMP	0000 0101 Γ	GOTO Γ
JMPZ	0000 0110 Γ	IF (Z=1) THEN GOTO Γ
JPNZ	0000 0111 Γ	IF (Z=0) THEN GOTO Γ
ADD	0000 1000	$AC \leftarrow AC + R$, IF $(AC + R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
SUB	0000 1001	$AC \leftarrow AC - R$, IF $(AC - R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
INAC	0000 1010	$AC \leftarrow AC + 1$, IF $(AC + 1 = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
CLAC	0000 1011	$AC \leftarrow 0$, $Z \leftarrow 1$
AND	0000 1100	$AC \leftarrow AC \wedge R$, IF $(AC \wedge R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
OR	0000 1101	$AC \leftarrow AC \vee R$, IF $(AC \vee R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
XOR	0000 1110	$AC \leftarrow AC \oplus R$, IF $(AC \oplus R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
NOT	0000 1111	$AC \leftarrow AC'$, IF $(AC' = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$

Quellen - Hersteller

zilog

Embedded in Life

An IXYS Company

Z80 Microprocessors

Z80 CPU

User Manual

UM008011-0816

www.zilog.com

Quellen – Offene Dokumentation

```
=================================================================
      Everything You Always Wanted To Know About GAMEBOY *
=================================================================

                    * but were afraid to ask

       Pan of -ATX- Document Updated by contributions from:
    Marat Fayzullin, Pascal Felber, Paul Robson, Martin Korth
            CPU, SGB, CGB, AUX specs by Martin Korth

                 Last updated 10/2001 by nocash
              Previously updated 4-Mar-98 by kOOPa
```

Game Boy: Complete Technical Reference

gekkio
https://gekkio.fi

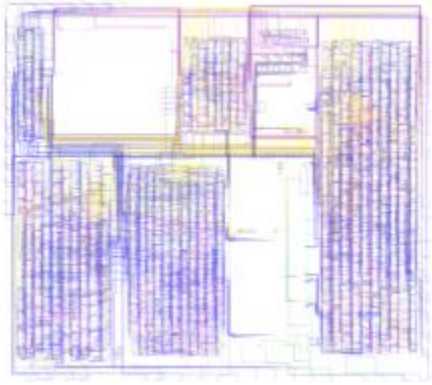
November 29, 2020

Revision 110

Game Boy™ CPU Manual

Sources by: Pan of Anthrox, GABY, Marat Fayzullin, Pascal Felber, Paul Robson, Martin Korth, kOOPa, Bowser

Quellen - Hardware



Quellen – Testprogramme

Mooneye Test Suite

Mooneye Test Suite is a suite of Game Boy test ROMs

MIT License

Languages

- Assembly 99.5%
- Other 0.5%

SameSuite

A suite of Game Boy tests, used for hardware research and emulator testing

MIT License

Languages

- Assembly 99.9%
- Makefile 0.1%

AGE test roms

a collection of Game Boy test roms to examine some Game Boy hardware edge cases

MIT License

Languages

- Assembly 89.4%
- PHP 8.9%
- Other 1.7%

Quellen – Emulatoren

Mooneye GB

A Game Boy research project and emulator written in Rust

GPL-3.0 License

Languages

- Rust 98.4%
- GLSL 1.4%
- Shell 0.2%

SameBoy

Game Boy and Game Boy Color emulator written in C

MIT License

Languages

- C 74.2%
- Objective-C 19.6%
- GLSL 2.5%
- Assembly 1.8%
- Makefile 1.7%
- Metal 0.1%
- Other 0.1%

higan

higan is a multi-system emulator, originally developed by Near, with an uncompromising focus on accuracy and code readability.

GPL-3.0 License

Languages

- GLSL 62.9%
- C++ 36.5%
- Other 0.6%

Quellen – Source Code

```
namespace higan::GameBoy {

#define SP r.sp.word
#define PC r.pc.word

#include "io.cpp"
#include "memory.cpp"
#include "timing.cpp"
#include "debugger.cpp"
#include "serialization.cpp"
CPU cpu;

auto CPU::load(Node::Object parent) -> void {
  wram.allocate(!Model::GameBoyColor() ? 8_KiB : 32_KiB);
  hram.allocate(128);
```

Erhalten

Quellcodes unter passender Lizenz stellen, z.b. GPL-3.0

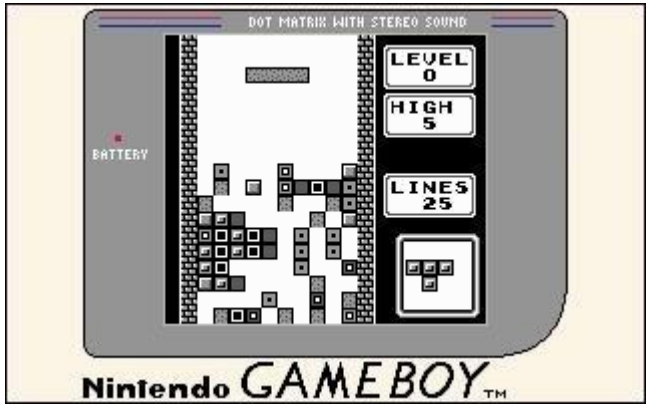
Veröffentlichen des Quellcodes, allgemein verfügbar

Versionsverwaltung benutzen

Alle Abhängigkeiten mindestens angeben,
wenn möglich auch zur Verfügung stellen

Buildanleitung erstellen

Als Software überall einsetzbar



In Hardware etwas eingeschränkter

Open Source zur Konservierung

Eine Möglichkeit nicht auf Wohlwollen und Existenz des Herstellers angewiesen zu sein

Möglichkeit auf neue Techniken reagieren zu können

Großteil Hobbyprojekte:
Basis für Nachfolger und Nachahmer schaffen

Kommerzielle Nutzung nicht ausgeschlossen
Beispiel:
PlayStation Classic nutzt OpenSource PCSX Emulator