Olivier Mattmann, COLIN FINGERLIN, MATIAS carballo González

[Company address]

NES-EMULATOR

CAOS-PROJEKT

# INHALTSVERZEICHNIS

[INHALTSVERZEICHNIS 1](#_Toc54905779)

[HARDWARE 2](#_Toc54905780)

[CPU 2](#_Toc54905781)

[Beschreibung 2](#_Toc54905782)

[APU 3](#_Toc54905783)

[Beschreibung 3](#_Toc54905784)

[PPU 3](#_Toc54905785)

[Memory 3](#_Toc54905786)

[Bus 3](#_Toc54905787)

# HARDWARE

Die Konsole besteht aus der CPU, der APU (Audio Processing Unit), der PPU (Picture Processing Unit), insgesamt 4kB RAM, und natürlich die ROM auf der Cartridge.

## CPU

### Beschreibung

Bei der CPU handelt es sich um ein 6502 Prozessor, der mit einer **Clockspeed von 1,79MHz** läuft. Die CPU besitzt 6 Register und 8 Flags, die unten beschrieben werden.

|  |  |  |
| --- | --- | --- |
| **Register** | **Grösse** | **Beschreibung** |
| PC (Programm Counter) | 16 Bit | Zeigt auf die Adresse der nächsten Instruktion |
| S (Stack Pointer) | 8 Bit | Zeigt auf die nächste leere Adresse in der Stack Memory (addiert zur ersten Adresse der Stack memory) |
| P (Processor status) | 8 Bit | Jedes Bit repräsentiert eine Flag (Wird unten gezeigt) |
| A (Accumulator) | 8 Bit | Hauptregister für arithmetische und logische Operationen |
| X (Index Register X) | 8 Bit | Hauptregister für die Datenaddresierung |
| Y (Index Register Y) | 8 Bit | Hat wenige Operationen |

|  |
| --- |
| Flags |
| N (Negative) |
| V (Overflow) |
| 1 (immer 1) |
| B (Break) |
| D (Decimal Mode) |
| I (Interrupt Disable) |
| Z (Zero) |
| C (Carry) |

Es müssten dazu die Instruktionen emuliert werden. Dafür benutzen wir verschiedene offizielle und nicht offizielle Sammlungen von Instruktionen und Opcodes.[[1]](#footnote-1)

#### Addressierungsmodi[[2]](#footnote-2)

#### Memory map[[3]](#footnote-3)

|  |  |  |
| --- | --- | --- |
| Address range | Size | Device |
| $0000-$07FF | $0800 | 2KB internal RAM |
| $0800-$0FFF | $0800 | [Mirrors](http://wiki.nesdev.com/w/index.php/Mirroring) of $0000-$07FF |
| $1000-$17FF | $0800 |
| $1800-$1FFF | $0800 |
| $2000-$2007 | $0008 | [NES PPU](http://wiki.nesdev.com/w/index.php/PPU_registers) registers |
| $2008-$3FFF | $1FF8 | Mirrors of $2000-2007 (repeats every 8 bytes) |
| $4000-$4017 | $0018 | [NES APU](http://wiki.nesdev.com/w/index.php/APU) and [I/O registers](http://wiki.nesdev.com/w/index.php/2A03) |
| $4018-$401F | $0008 | APU and I/O functionality that is normally disabled. See [CPU Test Mode](http://wiki.nesdev.com/w/index.php/CPU_Test_Mode). |
| $4020-$FFFF | $BFE0 | Cartridge space: PRG ROM, PRG RAM, and [mapper](http://wiki.nesdev.com/w/index.php/Mapper) registers (See Note) |

We use memory in the form of a char Array because it makes it easy to address using the cartridge addressdata.

#### Interrupts[[4]](#footnote-4)

#### Pins[[5]](#footnote-5)

## APU

### Beschreibung

Die APU kümmert sich um das Processing vom Audio.

## PPU

## Memory

## Bus

# Implementation Plan

## CPU

### Opcodes

Possibility to add new Opcodes to the structure should be guaranted.

1 Byte Opcode (Hardware = Position in Opcode Matrix)

For each opcode different addressing modes, Number of Parameters and clock cycles[[6]](#footnote-6)

### Registers

## Memory

Ideas:

* One Global char Array, Index as address
* Different Arrays depending on component

## Bus

Read/Write functions for 8 Bit data with 16 Bit addresses as parameter.

1. http://wiki.nesdev.com/w/index.php/CPU\_unofficial\_opcodes [↑](#footnote-ref-1)
2. http://wiki.nesdev.com/w/index.php/CPU\_addressing\_modes [↑](#footnote-ref-2)
3. http://wiki.nesdev.com/w/index.php/CPU\_memory\_map [↑](#footnote-ref-3)
4. http://wiki.nesdev.com/w/index.php/CPU\_interrupts [↑](#footnote-ref-4)
5. http://wiki.nesdev.com/w/index.php/CPU\_pin\_out\_and\_signal\_description [↑](#footnote-ref-5)
6. http://www.obelisk.me.uk/6502/instructions.html [↑](#footnote-ref-6)