

## Voorwoord

CODES: D:Dante, L:Lucas, Ke:kerst, Kr:Krokus, Pa:Pasen

## Inhoud

Voorwoord.....	1
Inleiding.....	4
Projectbeschrijving (D,Ke).....	4
Hardware CPU (L).....	4
A-register (Kr) .....	4
B-register (Kr).....	4
ALU ((Ke).....	4
RAM (Kr) .....	4
ROM (Kr) .....	4
Program counter (Ke).....	4
Instruction decoder (Ke) .....	4
Hardware I/O (L) .....	4
NES-controller + interface (Pa).....	4
PPU (D,Ke).....	4
GPU (Pa).....	4
Software CPU (D) .....	5
Website (Pa).....	5
Simulatie van de game (Kr).....	5
Webpage werking.....	5
Programma simulatie.....	5
Programma van de game (Pa) .....	5
HEX werking .....	5
Assembly .....	5
Zelf een IDE maken.....	5
Game design .....	5
Software I/O (D).....	5
GPU (Kr) .....	5
PPU .....	5
pixelart.....	5
Snelheid optimalisatie.....	5
Bijlage .....	6

Schema's..... 6  
Bronnen ..... 6  
Programma's..... 6

Inleiding

Projectbeschrijving (D,Ke)

Hardware CPU (L)

A-register (Kr)

B-register (Kr)

ALU ((Ke)

De ALU of arithmetic logic unit is het deel van een centrale verwerkings eenheid of CPU die alle berekening uitvoert. In een CVE met meerdere kernen zit er een ALU in elke kern. Een ALU rekt op basis van logische poorten door middel van comparatoren en optel circuits. Zo'n optel circuit laat de ALU toe om twee binaire waarden bij elkaar op te tellen, aftrekken gebuert via de 2-complements methode.

Een opteller bestaat in twee versies namelijk een halve en volledige opteller. De halve opteller heeft twee ingangen: bit  $A_x$  en bit  $B_x$  en twee uitgangen namelijk de som  $S_x$  en de rest  $C_{out}$ . De  $x$  duidt de positie van de bit aan in een binaire getal. Eeb volledige opteller neemt daar een extra ingang bij nammelijk een rest in  $C_{in}$ . Het optellen van twee bits kan als volgt worden samengevat:  $0+0=0$ ,  $0+1=1+0=1$ ,  $1+1=10$ . In praktijk houdt dat in dat  $C_{out}$  0 blijft totdat bijde ingangen  $A_x$  en  $B_x$  1 zijn

RAM (Kr)

ROM (Kr)

Program counter (Ke)

Instruction decoder (Ke)

Hardware I/O (L)

NES-controller + interface (Pa)

PPU (D,Ke)

GPU (Pa)

Software CPU (D)

Website (Pa)



Simulatie van de game (Kr)

Webpage werking

Programma simulatie

Programma van de game (Pa)

HEX werking

Assembly

Zelf een IDE maken

Game design

*Programma*

*ART*

*Gameplay loop*

Software I/O (D)

GPU (Kr)

PPU

pixelart

Snelheid optimalisatie

## Bijlage

### Schema's

Afbeeldingen (bijlage)

### Bronnen

Boek: Digitale Technieken 3<sup>de</sup> graad, die keure, H. Saeyes/ H. Vandenheede | ISBN 9057513099

<https://eater.net/8bit/schematics>

<https://stackoverflow.com/>

### Programma's

Whimsical: <https://whimsical.com/vties-BPHo6qPMGcZfC1St86wmkk>

Logisim: <https://github.com/logisim-evolution/logisim-evolution/releases>

### Chips

EEPROM: <https://nl.rs-online.com/web/p/eprom/1276552>