CPU Role Play - Program „Fibonacci Numbers“

Starting Configuration

The **Program** is stored in memory starting at memory cell 0 (start)

The **Data** is stored in memory starting at memory cell 100:

The memory cell with address 100 contains the number of planned iterations (e.g. the number 3)
The memory cell with address 101 contains the number 1
The memory cell with address 102 contains the number 1

Program (Assembly Language)

```
# Get the number of planned iterations from memory cell hundred and store it
# in register D
LOAD (100) ->D

#Jump label (remember this memory address for later)
loop:

# Get the last two F-numbers from memory (cells 101 and 102)
LOAD (101) -> A
LOAD (102) -> B

# Add the registers A and B, store the result in register C
ADD A,B -> C

# Overwrite the first fibonacci number with the second
LOAD (102) -> A
LOAD A -> (101)

# store register C as second F-number
LOAD C-> (102)

# Decrease number of remaining iterations by one („decrement“)
DEC D

# if register D is not zero yet-> jump to address „loop“
JMPDNZ loop:

# Output the result (e.g. on a display)
OUT (102)
```

Program (Maschine Language)

Adress (Bin)	Adress (Dec)	Content (Bin)	Content (Dec)
0000 0000	0	0000 0010	2
0000 0001	1	0110 0100	100
0000 0010	2	0000 0000	0
0000 0011	3	0110 0101	101
0000 0100	4	0000 0001	1
0000 0101	5	0110 0110	102
0000 0110	6	0000 0005	5
0000 0111	7	0000 0000	0
0000 1000	8	0110 0110	102
0000 1001	9	0000 0011	3
0000 1010	10	0110 0101	101
0000 1011	11	0000 0100	4
0000 1100	12	0110 0110	102
0000 1101	13	0000 0110	6
0000 1110	14	0000 0111	7
0000 1111	15	0000 0010	2
0001 0000	16	0000 1000	8
0001 0001	17	0110 0110	102

Data (Initial State)

0110 0100	100	0000 0011	3	
0110 0101	101	0000 0001	1	first Fibonacci number
0110 0110	102	0000 0001	1	second Fibonacci number