Embedding Microcode in the Sourcecode of the Programming Language O

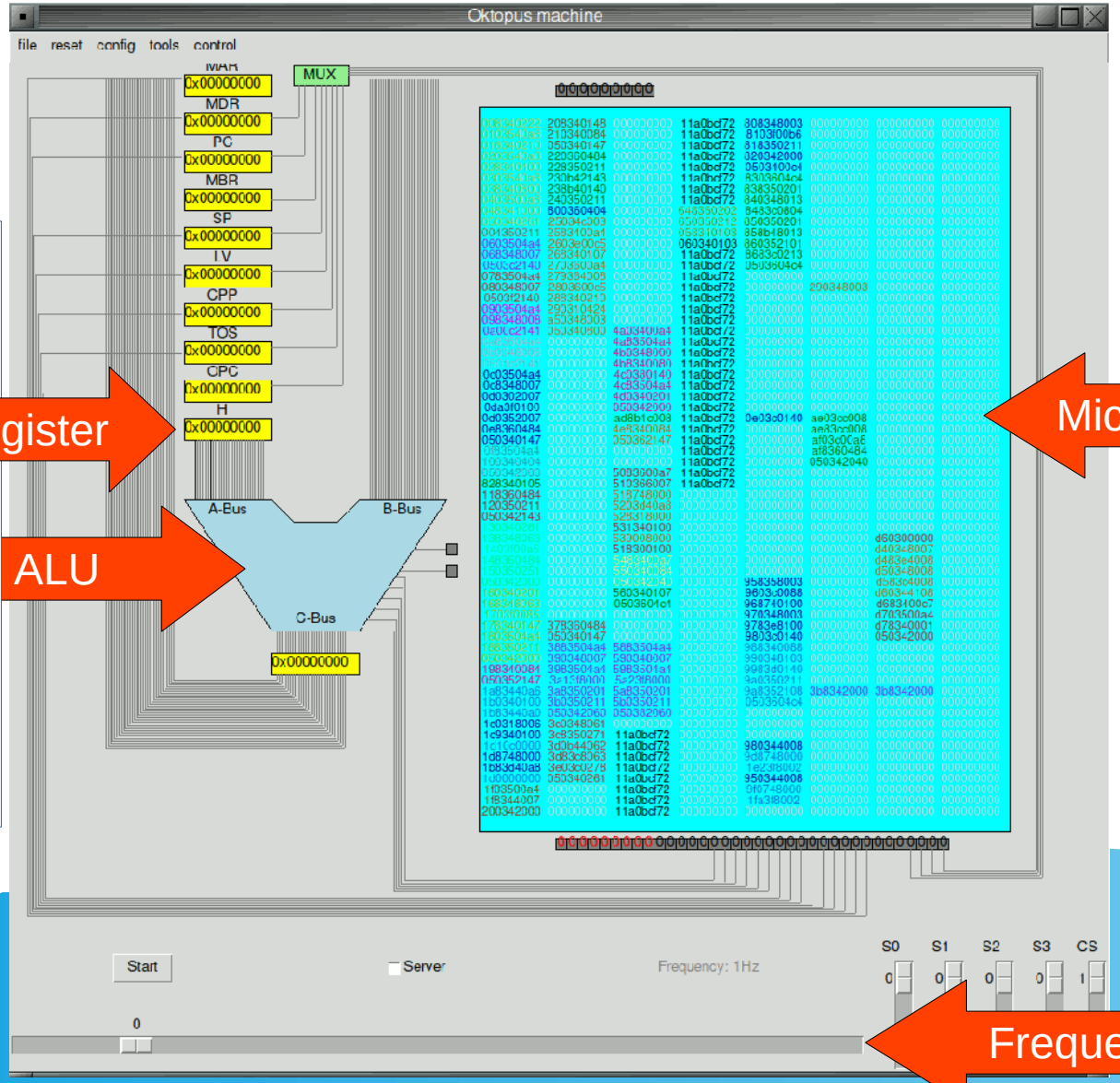
Prof. Dr. Thomas Pucklitzsch
DHSN Saxony/Germany

Prof. Dr. Mathias Sporer
DHSN Saxony/Germany

Anja Pucklitzsch
DHSN Saxony/Germany

The Octopus-Machine

Implentation
* Python
* C (faster)
Visualisazion
* Signals
* Registers
* Control Unit
Execution
*works like Hardware
Memory
*8KB

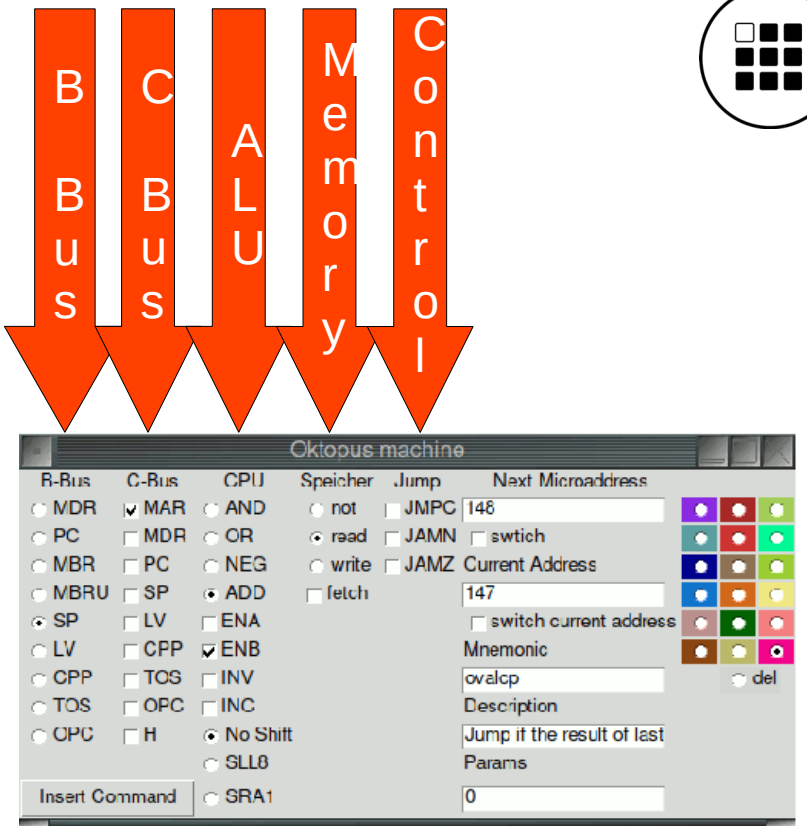


DUALE HOCHSCHULE SACHSEN

Microcode

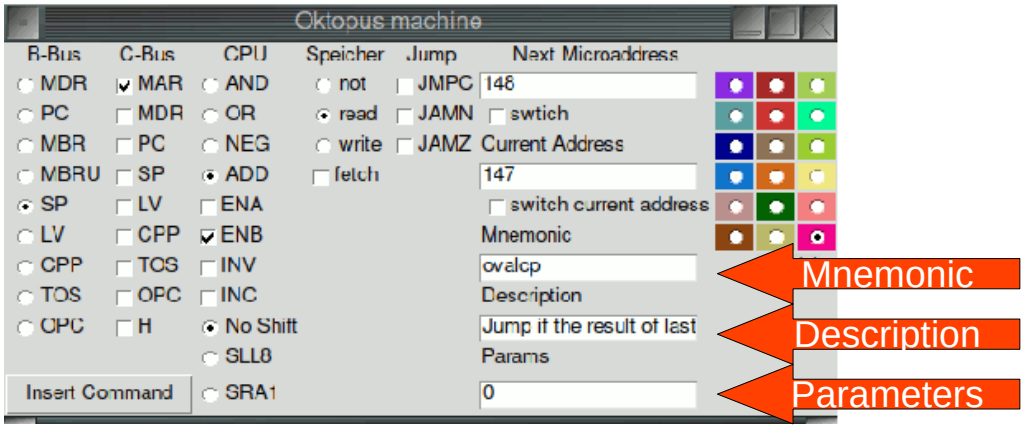
You can create your own Microcommands easy with a simple Form.

* Source Register
* ALU - Functions
* Destination Register
* Memory Commands
* Control Commands (Jump)



Generating Assembler

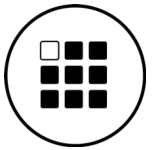
Because you can create your own Microcode, every Instance of the Octopus Maschine can be different. So you need a different Assembler for every Microprogramm. You can give for every Microcode a Mnemonic and the number of Parameter and you can generate a Assembler written in Python.



DUALE HOCHSCHULE SACHSEN

IARIA

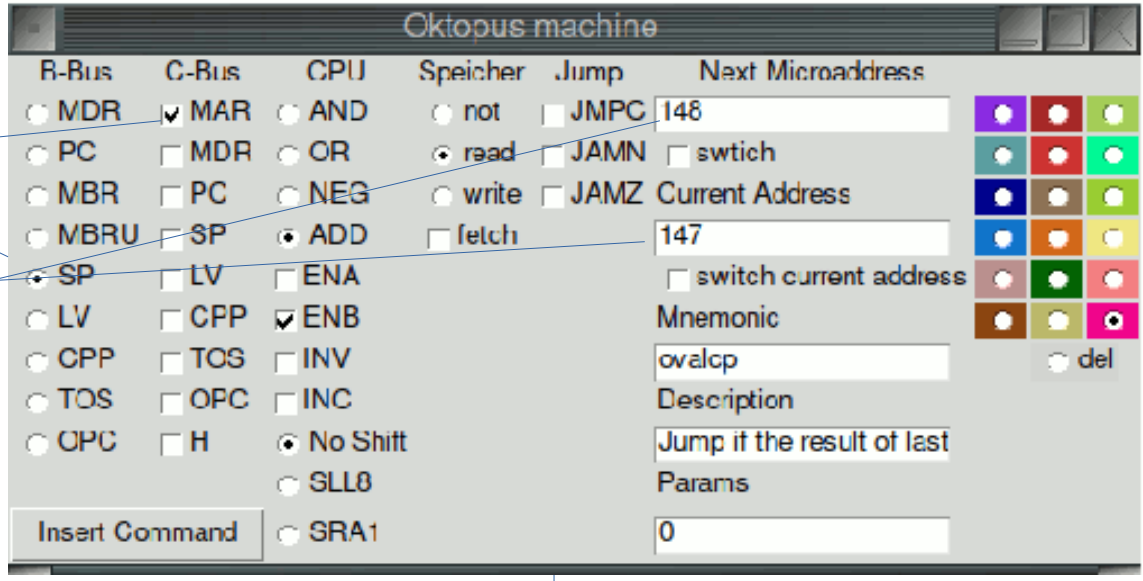
The Programming Language O



Requirements
* based on Extendable Markup Language (XML)
* simple parser like the parser of Lisp
* using namspaces for different versions of Microcode

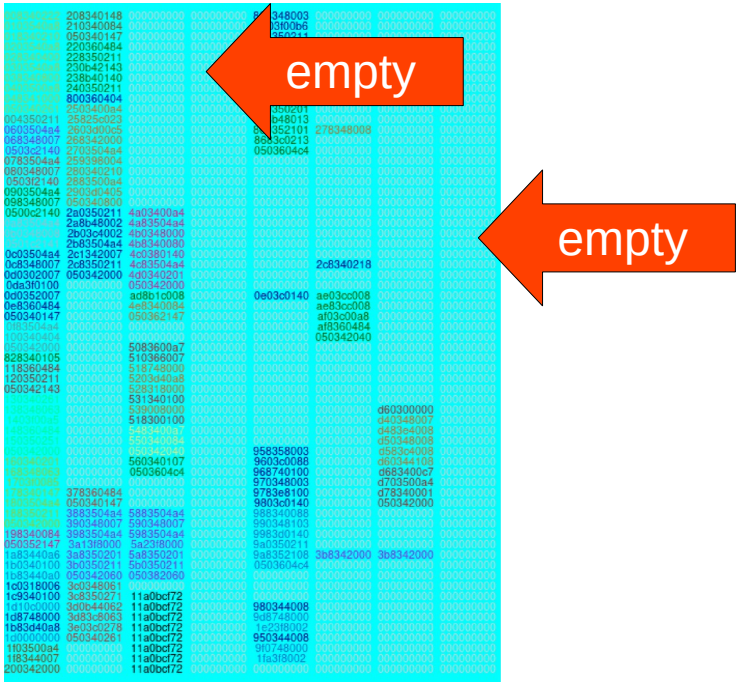
Insert Microcode in O

```
<o>
 <add_microcommand>
   <name>new_microcommand</name>
   <b-bus>MDR</b-bus>
   <c-bus>TOS</c-bus>
   <inc>1</inc>
    …
    <addr>62</addr>
    <naddr>63</addr>
    <side>right</side>
    <nside>left</nside>
  </add_microcommand>
</o>
```



Realiziation in Hardware

* tow ranges of microprogrammemory are reserved
* jumps are possible
* modification of the existing microprogram is not possible

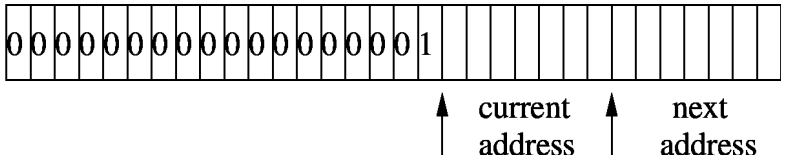


Microprogram to extend Microcode

* every microcommand contains the address of the next one
* a microcommand need 36 bit
* a address range of memory is mapped to the microprogrammemory
* the address contains the microprogramaddress and the address of the next microcommand

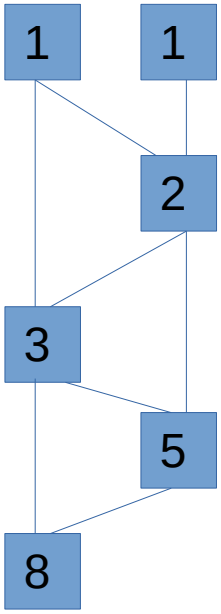
Ossamblercommands:

- omic – three bits for the microcommand and two bits for the addresses
- define – insert the new mnemonc into the list of the Ossambler



2^{15}-2^{16}

Example Fibbonacci numbers



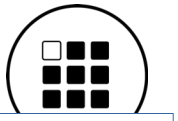

```
oload var2 (6)
odup (2)
odup (2)
:loop
oadd (3)
odup (2)
odup (2)
ostore var2 (7)
oload var1 (6)
oadd (3)
odup (2)
odup (2)
ostore var1 (7)
oload var2 (6)
ogoto loop (6)
```

46 cycles

```
H=MDR=1, wr
MAR = MDR=PC=1
MDR=MDR+H, wr
PC=MAR=PC+1
OPC=MDR+H
H=MDR
MDR=OPC, wr, goto 3
```

5 cycles

Conclusion



* the compiler and the hardware must support inserting Microcode
* changing Microcode is hard for real hardware
* embedding Microcode into sourcecode is easy
* it hast advantages for the performance

References

[1] B. Kollenda et al., “An exploratory analysis of microcode as a building block for system defenses,” 2020.

[2] P. Borrello, C. Easdon, M. Schwarzl, R. Czerny, and M.Schwarz, “Customprocessingunit: Reverse engineering and customization of intel microcode,” 2023.

[3] T. Pucklitzsch and M. Sporer, O - a new approach for a very simple language for distributed computation, 2025.

[4] A. S. Tanenbaum and T. Austin, Computerarchitektur - von der digitalen logik zum Parallelrechner, 2014.