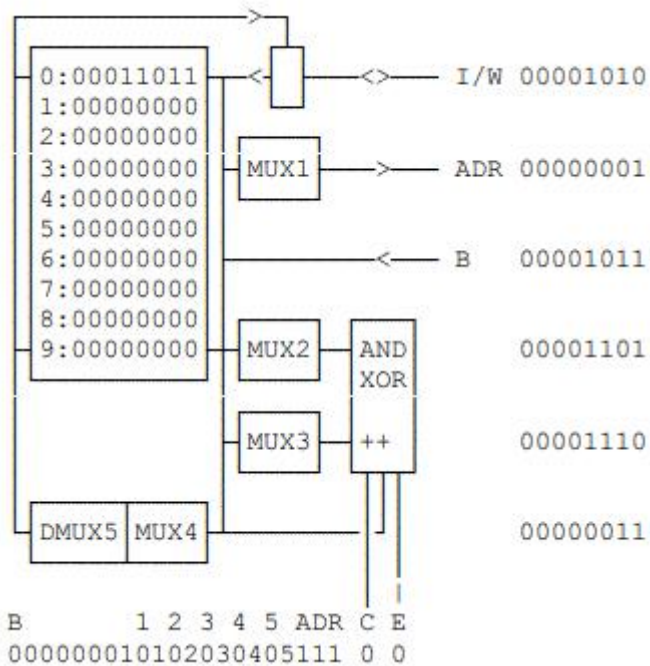


CPU-Modell - Mikrokode



B	1	2	3	4	5	ADR
00000001	01	0203	0405	001		
00000010	00	0000	0000	002		
00000011	00	0102	1403	002		
00000100	00	1010	1401	001		

Die Leitungen sind innerhalb der CPU Leitungsbündel, die alle Quellen allen Zielen zur Verfügung stellen. Daher lassen sich gleichzeitig Register R1 über MUX1 mit der RAM-Adressleitung, Register R2 mit MUX2, Register R3 mit MUX3 und Register R4 mit MUX4 verbinden (vgl oben erste Zeile).

Zusätzlich zu den Registernummern 0 - 9 kann man bei den MUXen noch die Zahlen 10 bis 14 zur Steuerung verwenden. Deren Wirkung ist der unteren Tabelle zu entnehmen.

0...9	Register R0 .. R9
I/W	Datenleitung / Flag RAM-Schreiben (1) / -Lesen (0)
B	Zahl, aus Mikrobefehl ins Rechenwerk eingekoppelt (Dualzahl)
C	Carry-Flag (CF): Übertrag aus der Addition
E	Equal-Flag (EF): 1 bei Gleichheit Operanden, 0 bei Ungleichheit
ADR	1) RAM-Adresse (MUX1) 2) Adressenteil des nächsten Mikrobefehls
MUX1	Wahl der RAM-Adresse (Dezimalzahl)
MUX2 , MUX3	Wahl der Operanden (Dezimalzahl)
MUX4	Wahl des Ergebnisses (s.u. - Dezimalzahl)
DMUX5	Anwahl der Register bzw. der Datenleitung (s.u. - Dezimalzahl)

Steuerung der MUXe und des DMUX erfolgt durch Zahlen:

```

0..9      Register
10        Datenleitung vom bzw. in das RAM
11        B-Leitung
12        AND-Ergebnis (bitweise)
13        XOR-Ergebnis (bitweise)
14        Additionsergebnis

```

Die Adresse des nächsten Mikrobefehls ergibt sich aus der Formel:

$$4 \cdot \text{ADR} + 2 \cdot \text{C} + \text{E} \quad (\text{ADR: Adressanteil des aktuellen Mikrobefehls})$$

Die Operationen AND, XOR, ADD werden immer ausgeführt und ausgewertet!

Interpreter Maschinenkode -> Mikrokode (ULog-8088.cpu)

Register:

von außen zugreifbar:

0:	AX	Akkumulator
1:	BX	Rechenregister
2:	CX	"
3:	DX	"
4:	ST	Statusregister (Bit 0: Zero, 1: Carry, 2: Overflow)
5:	PC	Befehlsadresse
6:	SP	Stackende

von außen nicht zugreifbar:

7:	BE	Befehl	und weitere Funktionen
8:	AD	RAM-Adresse	"
9:	RE	Rechenregister	"

Befehle:

00:	NOP	keine Wirkung
01:	MOV AX,BX	BX -> AX
02:	MOV AX,CX	CX -> AX
03:	MOV AX,DX	DX -> AX
04:	MOV BX,AX	AX -> BX
05:	MOV CX,AX	AX -> CX
06:	MOV DX,AX	AX -> DX
07:	ADD AX,BX	AX+BX -> AX
:	:	:
0A:	AND AX,BX	AX&BX -> AX
:	:	:
0D:	OR AX,BX	AX BX -> AX
:	:	:
10:	SBC AX,BX	AX-BX -> AX
:	:	:
13:	CMP AX,BX	(AX-BX=0) -> ST
:	:	:
16:	NOT AX	NOT(AX) -> AX
17:	NOT BX	NOT(BX) -> BX
:	:	:
1A:	SHL AX	2*AX -> AX
:	:	:
21:	MOV AX,[BX]	[BX] -> AX
:	:	:
27:	ADD AX,[BX]	AX+[BX] -> AX
:	:	:
2A:	AND AX,[BX]	AX&[BX] -> AX
:	:	:
30:	JMP AX	AX -> PC
:	:	:
34:	JSR AX	SP+1 -> SP, PC -> [SP], AX -> PC
:	:	:
38:	JZ AX	Bit 0 in ST gesetzt => AX -> PC
:	:	:
3C:	JC AX	Bit 1 in ST gesetzt => AX -> PC
:	:	:
40:	JNZ AX	Bit 0 in ST nicht gesetzt => AX -> PC
:	:	:

81:	MOV AX,Zahl	Zahl -> AX
:	:	:
A1:	MOV AX,[Zahl]	[Zahl] -> AX
:	:	:
B1:	MOV [Zahl],AX	AX -> [Zahl]

Maschinencode-Interpreter:

Funktionsweise:

Nach der Initialisierung (Adresse 0), bei der der Stackpointer SP auf &HFF gesetzt wird, holt sich der Rechner in einer Schleife (ab Adresse 4) jeweils ein Byte, das Befehlsbyte aus dem RAM (Adresse 4-7) und untersucht das höchste Bit (Adresse 16,17):

Ist dieses 0, so stellt das erste Byte einen 1-Byte-Befehl dar.

Ist hingegen das höchste Bit 1, so muss ein zweites Byte gelesen werden, das eine Zahl, das Zahlenbyte enthält. Das Zahlenbyte wird direkt als Zahl genutzt, wenn das Befehlsbyte zwischen 81 und 9F liegt.

Liegt das Befehlsbyte zwischen A1 und AF, so ist das Zahlenbyte eine RAM-Adresse, aus der ein weiteres Byte gelesen wird, das dann durch das Befehlsbyte als Zahl verarbeitet wird.

Liegt das Befehlsbyte zwischen B1 und BF, so ist das Zahlenbyte eine RAM-Adresse, in der später eine Zahl gespeichert wird.

In der Adresse 21 wird dann der NOP-Befehl in Mikrokode umgesetzt, in der Adresse 25 der Befehl MOV AX,BX. Es folgt der Rücksprung zu einer der Adressen 4-7.

Hinweis: Die Mikro-Befehle lassen die Bestimmung des CF- und EF-Flags zu, so dass hierdurch die möglichen Folgeadressen eingeschränkt und Verzweigungen realisiert werden können.

Interpreteranfang:

Es sind nicht alle Mikrokodezeilen aufgeschrieben, fehlende Zeilen sind identisch mit der letzten hier vermerkten Zeile.

Mikrobefehl:	Wirkung:	Bedingung:	für Folgeadresse:
B M1M2M3M4D5ADR 0: 1111111100000001106001	11111111 -> SP		4-7 -> ADR
4: 000000010511051405002	PC -> Adr.Bus, PC+1 -> PC		8-9 -> ADR
8: 1000000000011101007003	Dat.Bus -> R7, 10000000+Dat.Bus -> CF	CF=1: 14-15 -> ADR CF=0: 12-13 -> ADR	
12: 0000000000011070000005	00000000==R7 -> EF	EF=1: 21 -> ADR EF=0: 20 -> ADR	
14: 000000010511051405004	PC -> Adr.Bus, PC+1 -> PC		16-17 -> ADR
16: 100000010011071008016	10000001==R7 -> EF	EF=1: 65,67 -> ADR EF=0: 64,66 -> ADR	
20: 000000010011070000006	00000001==R7 -> EF	EF=1: 25 -> ADR EF=0: 24 -> ADR	
21: 0000000000000000000001	- (NOP)		4-7 -> ADR
24: 00000001000110700000007	00000010=R7 -> EF	EF=1: 29 -> ADR 28 -> ADR	
25: 0000000000000000100001	BX -> AX (MOV AX,BX)		4-7 -> ADR

Das Programm setzt sich analog zu den Zeilen 21 usw. für die restlichen realisierten Maschinenbefehle fort. Da hier keine neuen Gesichtspunkte auftreten, ist der Rest des Programmes hier nicht mehr dargestellt.