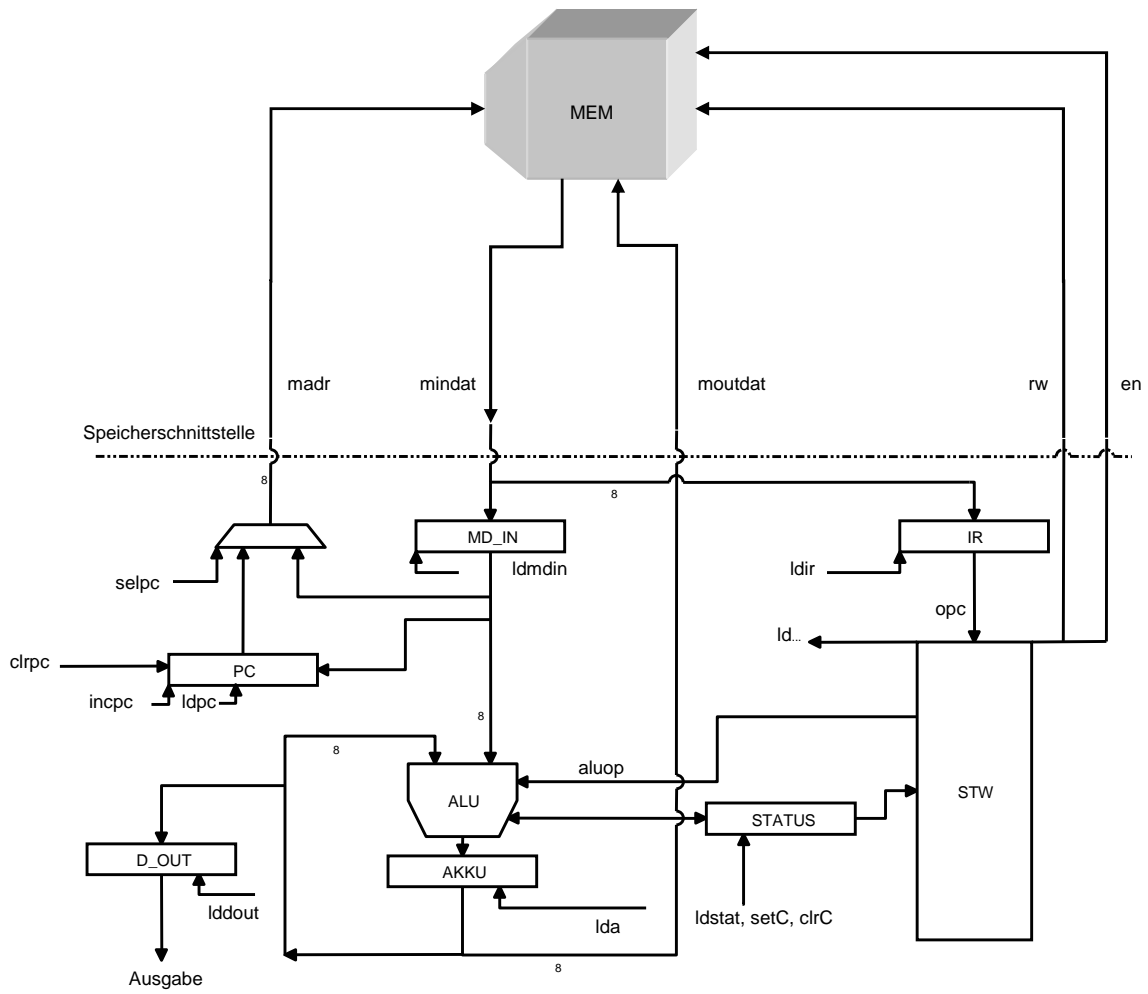


# Ein einfacher Modellrechner



---

---

## Befehlsliste

---

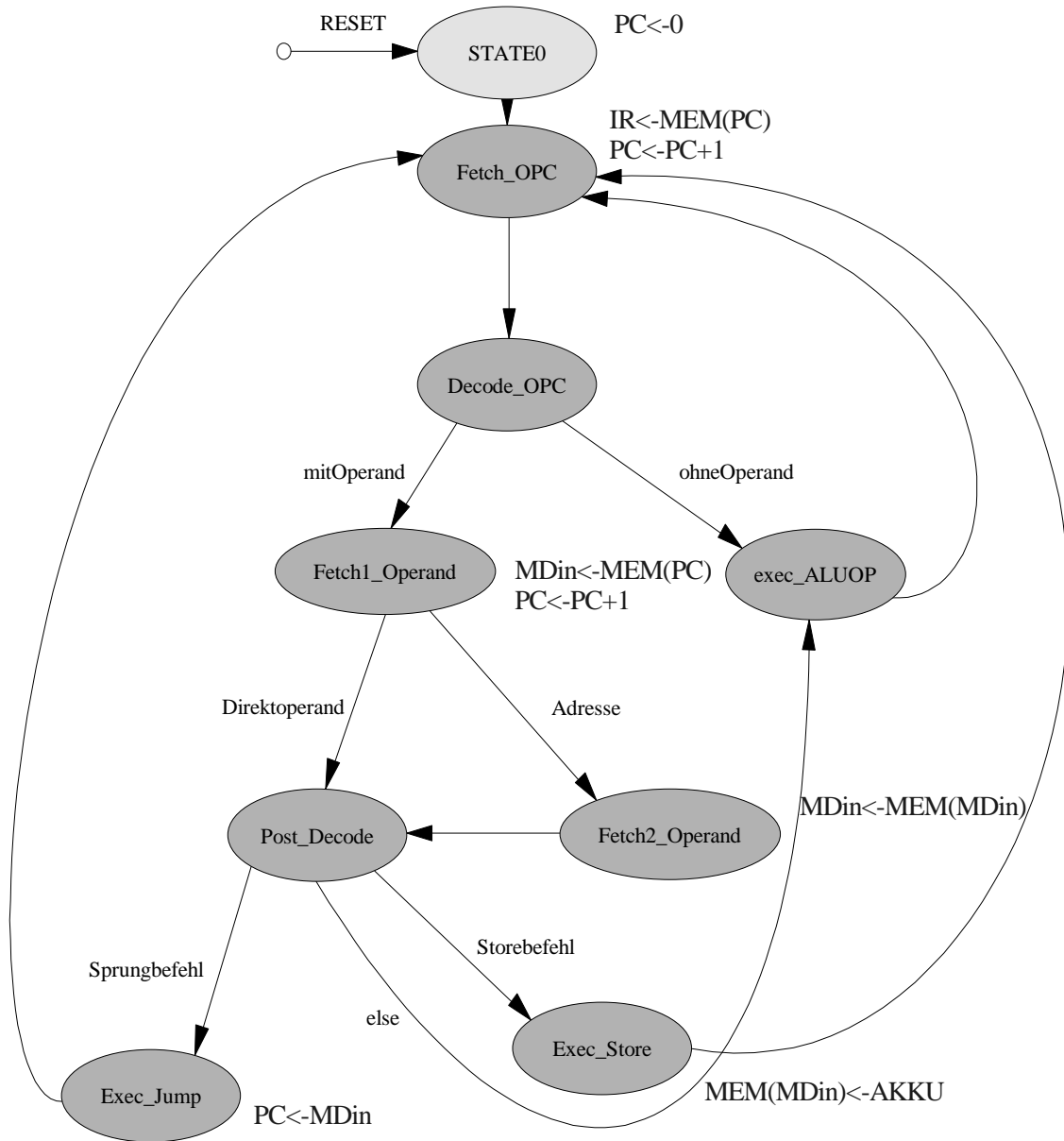
Die Unterscheidung, ob im 2. Byte eine Operandenadresse oder der Wert des Operanden steht, ist im niederwertigsten Bit des 1. Befehlsbytes vermerkt. Die in der Spalte *MNEMONIC* benutzten Operatoren haben folgende Bedeutung. *C* bezeichnet eine Konstante, *N* eine Speicheradresse und *L* ein Label.

**Status:** Beim Statusbits bedeutet *!* = *wird verändert* und *+* = *wird nicht verändert*.

Der Eintrag – bedeutet, dass die entsprechenden Befehle keine Statusbits beeinflussen.

OPC	I	Befehl	MNEMONIC	Byte1	Byte2	Status OCNZ
0	0	lade AKKU mit Wert	LDA #C	0000 0000	WERT	00!!
0	1	lade AKKU mit dem Speicherinhalt von Adresse	LDA N	0000 0001	ADRESSE	00!!
1	X	AKKU bitweise negieren	NOT	0001 000x		00!!
2	0	AKKU mit Wert verunden	AND #C	0010 0000	WERT	00!!
2	1	AKKU mit dem Speicherinhalt von Adresse verunden	AND N	0010 0001	ADRESSE	00!!
3	0	AKKU mit Wert verodern	OR #C	0011 0000	WERT	00!!
3	1	AKKU mit dem Speicherinhalt von Adresse verodern	OR N	0011 0001	ADRESSE	00!!
4	0	Wert zum AKKU addieren	ADD #C	0100 0000	WERT	!!!!
4	1	Speicherinhalt von Adresse zum AKKU addieren	ADD N	0100 0001	ADRESSE	!!!!
5	X	AKKU nach rechts schieben	SHR	0101 000x		0!!!
6	X	AKKU nach rechts schieben Vorzeichen nachziehen	ASR	0110 000x		0!!!
7	X	AKKU nach links schieben	SHL	0111 000x		0!!!
8	X	Carrybit im Statusregister löschen	CLRC	1000 000x		+0++
9	X	Carrybit im Statusregister setzen	SETC	1001 000x		+1++
10	X	NO $\bar{O}$ keine Operation	NOP	1010 000x		–
11	X	Sprung zur Adresse Label	BRA L	1011 0000	LABEL	–
12	X	Sprung zur Adresse Label falls Z-Statusbit gesetzt ist	BRZ L	1100 0000	LABEL	– –
13	X	Schreibe Inhalt des AKKUs an Adresse in den Speicher	STA N	1101 0000	ADRESSE	– –
14	X	Schreibe Inhalt des AKKUs in das Ausgaberegister	OUT	1110 000x		– –

# Zustandsgraph des Steuerwerks



---

---

## Disassembliertes Programm

---

```
0 : 00; Lade AKKU mit Wert
1 : 01; Wert = 1
2 : e0; Gib Inhalt des AKKUs aus
3 : 70; shl
4 : e0; Gib Inhalt des AKKUs aus
5 : c0; Sprung nach Label, falls Z-Bit gesetzt
6 : 09; Label = 9
7 : b0; Sprung nach Label
8 : 03; Label = 3
9 : 00; Lade AKKU mit Wert
a : 80; Wert = 128
b : e0; Gib Inhalt des AKKUs aus
c : 50; lshr
d : e0; Gib Inhalt des AKKUs aus
e : c0; Sprung nach Label, falls Z-Bit gesetzt
f : 00; Label = 0
10 : b0; Sprung nach Label
11 : 0c; Label = 12
```

Zuerst wird das niederwertigste Bit gesetzt und in jeder Schleife ausgegeben und nach links geschiftet, wenn es über die Wortgrenze gewandert ist wird das höchstwertige Bit gesetzt und in einer neuen Schleife ausgegeben und läuft nach rechts durch. Danach springt das Programm wieder zum Anfang, so dass dieser Vorgang beliebig oft wiederholt wird.