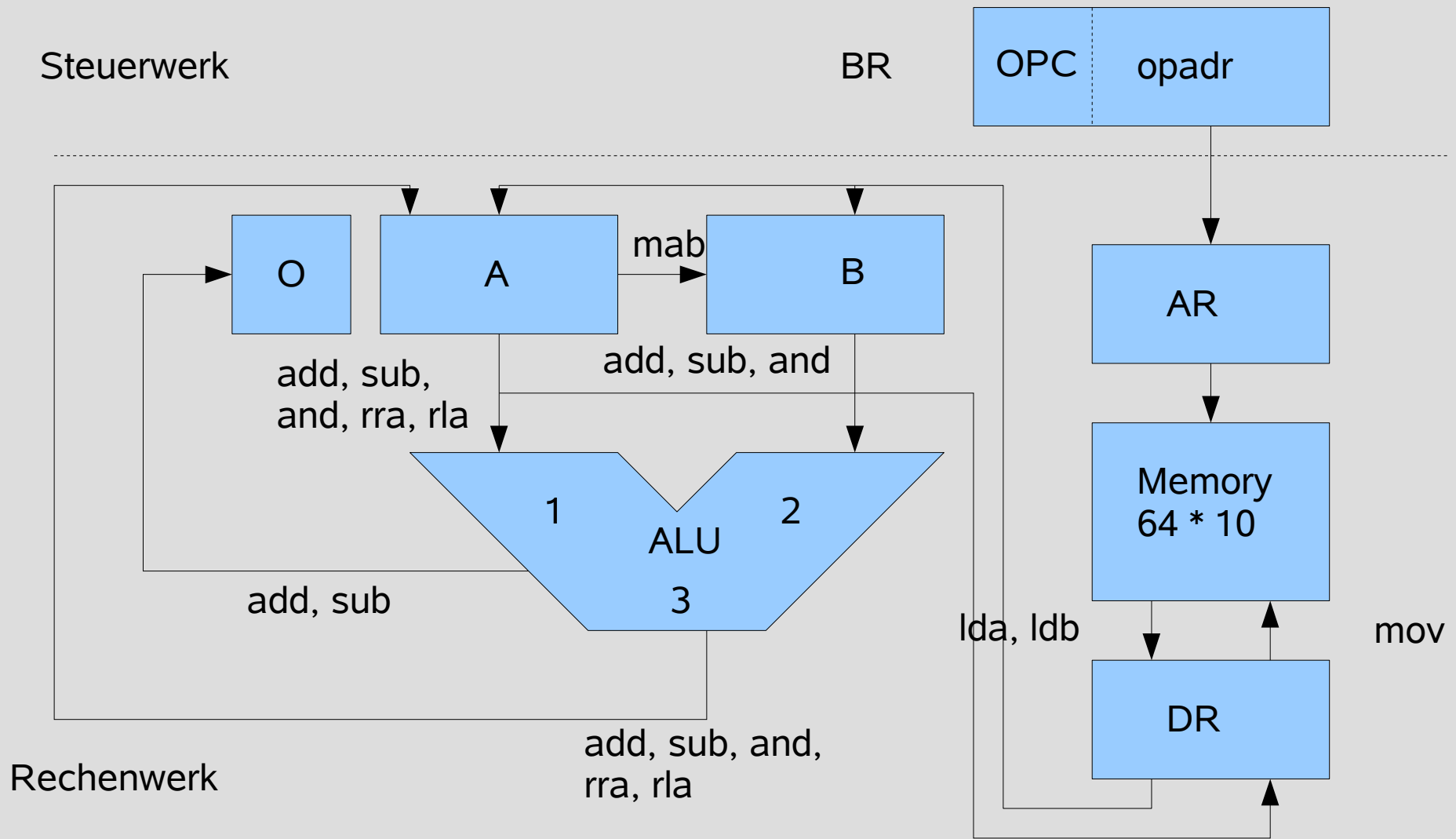


LC 1 (Forsetzung)



LC 1

- Arbeitsweise des Rechenwerkes
 - nach Fetchphase steht im der aktuelle Befehl im Befehlsregister **BR**
 - Steuerwerk (SW) liest den Operationskode **OPC**
 - von **OPC** abhängig:
 - Funktion der ALU
 - Datentransporte im Rechenwerk (RW)
 - z.B.: MOV opadr AR := opadr

	DR := A	
	MEMORY (AR) := DR	
ADD	ALU1 := A	Flags: SF, OF
	ALU2 := B	
	ALU3 := ALU1 + ALU2	
	A := ALU3	

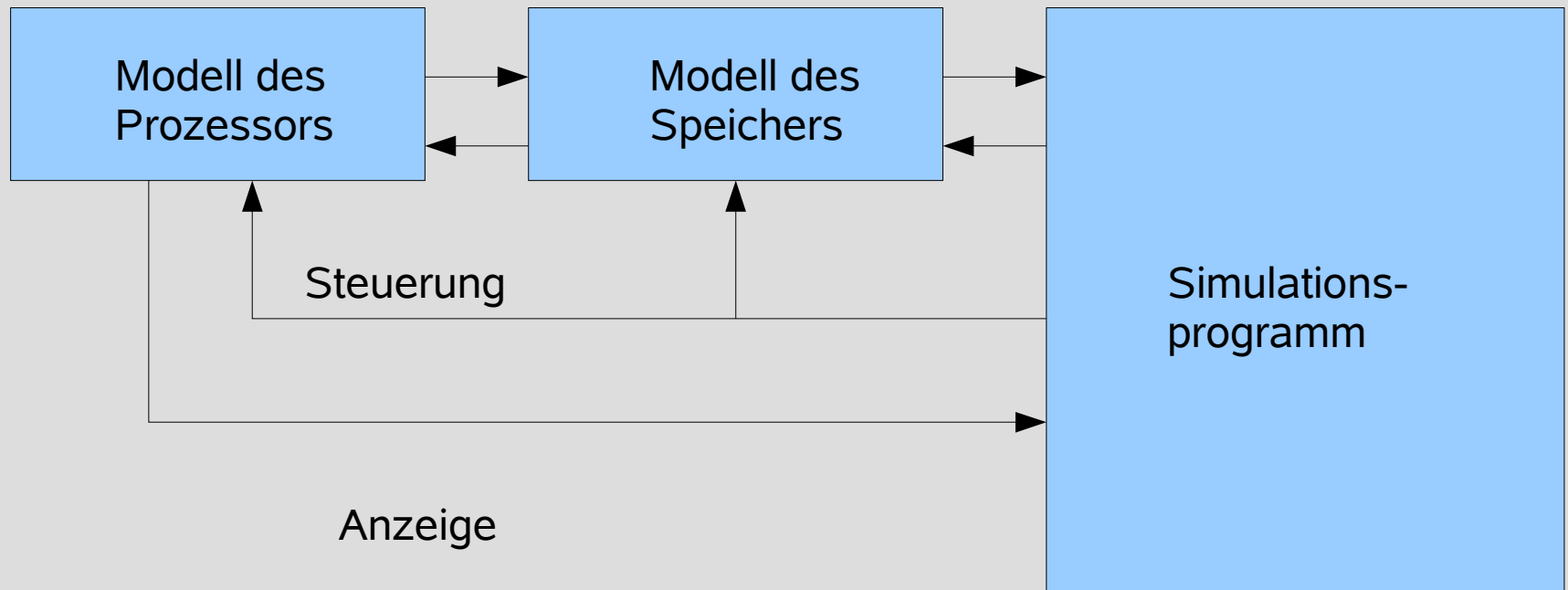
LC 1

- ein einfaches LC 1 Programm
 - $s := x + y;$ {Pascal-Notation}
 - LC-Programm

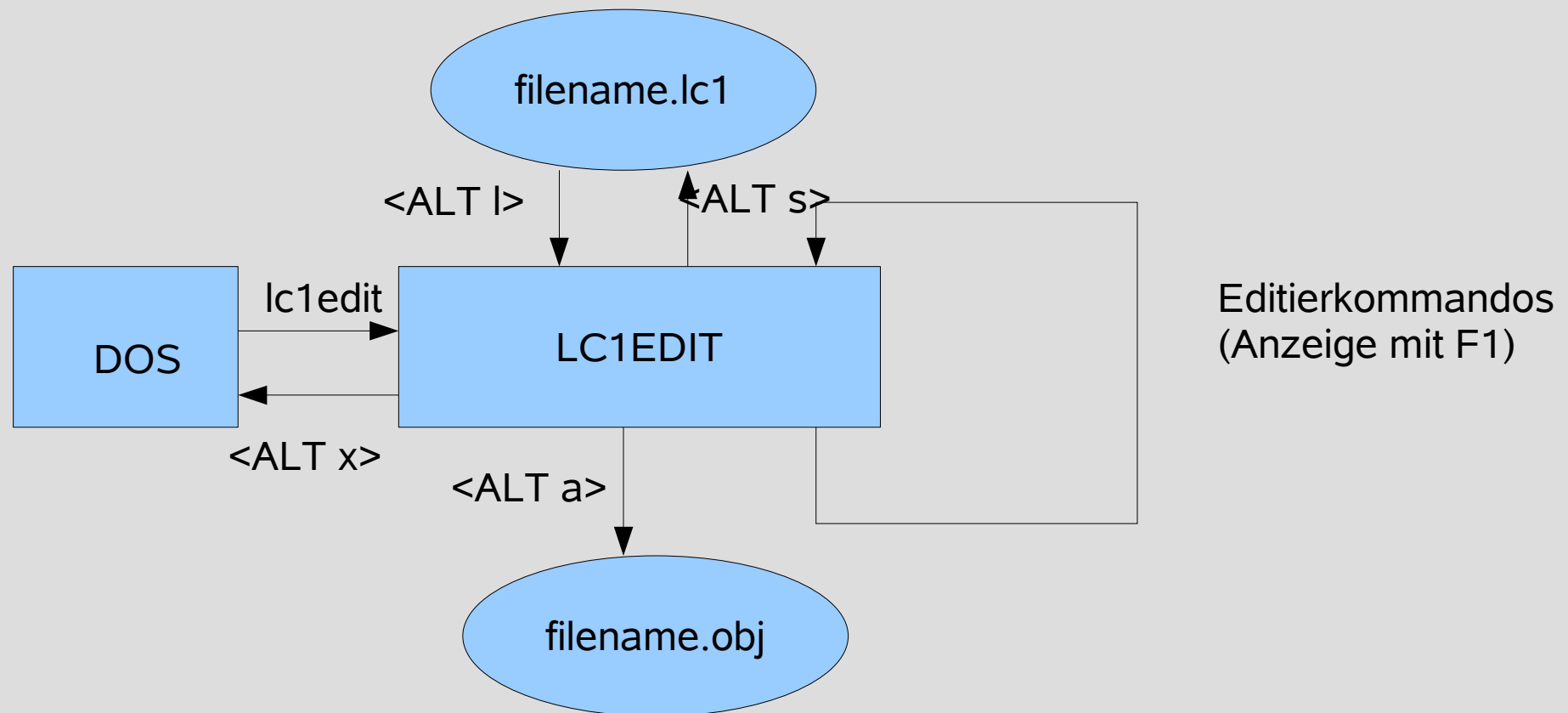
```
LDA X
LDB Y
ADD
MOV S
HLT
X: DEF 12
Y: DEF -37
S: DEF 0
```

LC1

- Bedienung



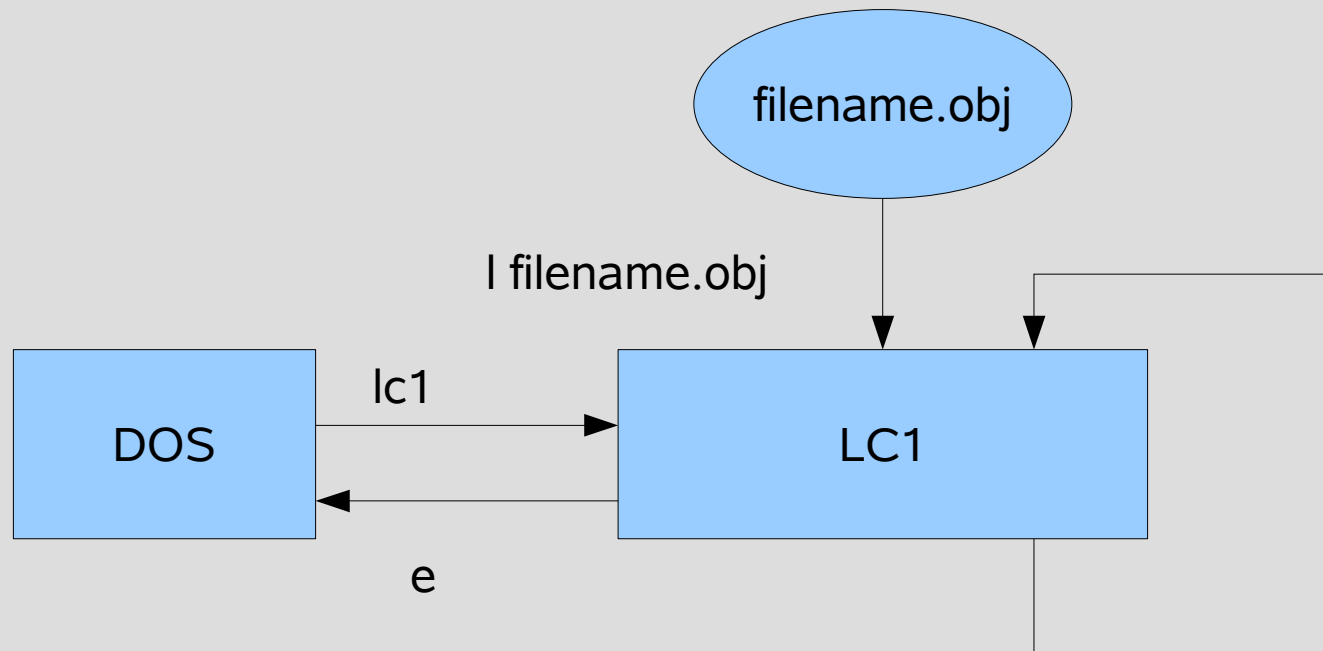
LC1EDIT



LC1EDIT

- Hotkeys:
 - ALT + I Laden eines lc1-Assemblerprogr.
 - ALT + s Speichern eines lc1-Assemblerprogr.
 - ALT + a, ESC Übersetzen eines lc1-Assmb.pr.
 - ALT + x Verlassen von LC1EDIT

Simulator LC1



Simulationskommandos:

- r – run
- x – single step
- v adr - Anzeige ab adr

Befehlsliste

- lda adr $A \leftarrow \text{Memory}(\text{adr}), [S]$
- ldb adr $B \leftarrow \text{Memory}(\text{adr})$
- mov adr $\text{Memory}(\text{adr}) \leftarrow A$
- mab $B \leftarrow A$
- add $A := A + B$
- sub $A := A - B$
- and $A := A \text{ **and** } B \text{ (bitweise)}, [S]$
- not $A := \text{**not** } A \text{ (bitweise)}, [S]$
- jmp adr $\text{PC} := \text{adr}$

Befehlsliste (2)

- jps adr if $S = 1$ then $PC := \text{adr}$
- jpo adr if $O = 1$ then $PC := \text{adr}$
- rra n $A_{9....0} := A_{09....1}$, n – mal
- rrl n $A_{9....0} := A_{8....09}$, n – mal
- hlt stop

Befehlssatz (3)

- Unterprogrammbeefehle
 - cal adr Memory(SP) := PC
 SP := SP -1
 PC := adr
 - ret SP := SP +1
 PC := Memory(SP)
- Assemblerdirektive
 - adr def wert Memory(adr) := wert

Vergleichsoperation

- Ergebnisse von Operationen werden nicht unmittelbar, sondern später (mindestens ein Befehl) ausgewertet (insbesondere bei 0 – 3 – Adressmaschinen)
- Dazu werden Aussagen über Berechnungsergebnisse in **Flags** gespeichert.
- Welche Vergleiche sind möglich?
 - Wie beeinflussen (verknüpfende) Befehle die Flags
 - Welche Auswertemöglichkeiten gibt es?

Vergleichsoperation (2)

- Auswertung des Ergebnisses der Subtraktion:
 - Operanden A,B ---> $\{-512 \dots 511\}$
 - Ergebnis $C=A-B$ ---> $\{-1024 \dots 1023\}$ (aber nur $\{-512 \dots 511\}$ darstellbar)

-1023...-513	-512...-1	0...511	512...1023
--------------	-----------	---------	------------

Vergleichsoperation (2)

- Auswertung des Ergebnisses der Subtraktion:
 - Operanden A,B ---> $\{-512 \dots 511\}$
 - Ergebnis $C=A-B$ ---> $\{-1024 \dots 1023\}$ (aber nur $\{-512 \dots 511\}$ darstellbar)

-1023...-513	-512...-1	0...511	512...1023
--------------	-----------	---------	------------

$S = 1$
 $O = 0$

$S = 0$
 $O = 0$

Vergleichsoperation (2)

- Auswertung des Ergebnisses der Subtraktion:
 - Operanden A,B ---> $\{-512 \dots 511\}$
 - Ergebnis $C=A-B$ ---> $\{-1024 \dots 1023\}$ (aber nur $\{-512 \dots 511\}$ darstellbar)

-1023...-513	-512...-1	0...511	512...1023
--------------	-----------	---------	------------

S = 1
O = 0

S = 0
O = 0

+1...+511

S = 0
O = 1

-512 ... -1

S = 1
O = 1

Vergleichsoperation (3)

-1023...-513	-512...-1	0...511	512...1023
--------------	-----------	---------	------------

S = 1
O = 0

S = 0
O = 0

+1...+511

S = 0
O = 1

-512 ... -1

S = 1
O = 1

A < B	A < B	A ≥ B	A > B
A < B		A ≥ B	

Beispiel

- $\text{produkt} = \text{faktor1} * \text{faktor2}$
- Lösungsansatz: faktor1-fache Addition von faktor2
- Wertebereich von LV1: -512 +511
- Produkt muss in diesem Intervall liegen, Faktoren sind entsprechend zu wählen.

Beispiel (2)

loop:	lda	faktor1	; A:= faktor1
	ldb	eins	; B := 1
	sub		; A := A – B
	jps	fertig	; if A < 0 then goto fertig
	mov	faktor1	; faktor1 := A
	lda	produkt	; A := produkt
	ldb	faktor 2	; B := faktor2
	add		; A := A + B
	mov	produkt	; produkt := A
	jmp	loop	; goto loop
fertig:	lda	produkt	; A := produkt
	hlt		
faktor1:	def	5	
faktor2:	def	3	
produkt:	def	0	
eins:	def	1	