

Abi 2014 III

4a)
r=11
w=1
1<11: w=2
4<11: w=3
9<11: w=4
16<11 (f)
4*4-11>4 (w)
w=3

Also: w=3, r=11

4b) (w in Speicherzelle 100,
r in Speicherzelle 101)
loadi 1
beginnwiederhole: store 100
mul 100
sub 101
jmpnn endewiederhole
load 100
addi 1
jmp beginnwiederhole
endewiederhole: sub 100
jmpnp fertig
load 100
subi 1
store 100
fertig: hold

Abi 2014 IV

1a) Wenn a gerade ist, wird in Speicherzelle 99 die Zahl 0 abgelegt, sonst 1.

1b)

1: load 100	
2: sub 103	
3: jle 18	12: jump 1
4: load 100	13: load 100
5: div 102	14: add 103
6: mult 102	15: div 102
7: sub 100	16: store 100
8: jne 13 //a ungerade	17: jump 1
9: load 100	18: hold
10: div 102	
11: store 100	

4a) Befehlszyklus

- Fetch-Phase I
Befehl holen (entsprechend der Adresse im Befehlszähler, hier Zahl die sub entspricht)
 $BZ = BZ + 1$
- Decode-Phase
Opcode des Befehls bestimmen: sub
- Fetch-Phase II
Operand laden: hier 100
BZ um die Anzahl der gelesenen Speicherzellen weiter schalten (hier 1)
- Execute-Phase
vom Wert im Akkumulator den Speicherinhalt von 100 subtrahieren, Ergebnis in Akkumulator schreiben;

4b) Programm und Daten sind in einem Speicher.

4c)

Befehl	BZ	A	SR	100	101	102
	0			4	5	
load 100	2	4		4	5	
cmp 101	4	4	N	4	5	
jmpnn 12	6	4	N	4	5	
load 101	8	5	N	4	5	
store 102	10	5	N	4	5	5
jmp 14	14	5	N	4	5	5
hold	16	5	N	4	5	5

Es wird die größere der beiden Zahlen aus 100 und 101 in 102 kopiert.

4d) n in Speicherzelle 100

schleife:	load 100	
	cmpi 3	
	jmpn ende	addi 1
	modi 4	store 100
	(cmpi 0)	jmp schleife
	jmpnz falsch1	load 100
	load 100	addi 1
	divi 4	divi 2
	store 100	store 100
	jmp schleife	jmp schleife
falsch1:	load 100	load 100
	modi 2	hold
	(cmpi 0)	
	jmpnz falsch2	
	load 100	
	divi 2	
		falsch2:
		addi 1
		store 100
		jmp schleife
		load 100
		hold
		ende:

4e)

Dann könnte es zu einer Endlosschleife kommen, denn wenn $n=2$ ist, dann wird $2/2+1$ berechnet und in n gespeichert. Dies ist wieder 2, d.h. der Algorithmus terminiert nicht.