

$$1. SP_{\text{chen pre}} = NB \cdot$$

$$(E \cdot S_0 \cdot \overline{S_1}) \cdot$$

$$(\overline{I_{31}} \cdot \overline{I_{30}} \cdot \overline{I_{27}} \cdot \overline{I_{26}} \cdot \overline{I_{25}} +$$

$$\overline{I_{31}} \cdot \overline{I_{30}} \cdot \overline{I_{27}} \cdot \overline{I_{26}} \cdot I_{25} +$$

$$I_{31} \cdot \overline{I_{30}} \cdot I_{29} \cdot I_{28} +$$

$$\overline{I_{31}} \cdot I_{30}) \cdot$$

$$I_{24} \cdot \overline{I_{23}} \cdot \overline{I_{22}} +$$

$$(h_0 \cdot h_1 \cdot h_2 + \overline{h_0} \cdot \overline{h_1} \cdot \overline{h_2})$$

Normalbetrieb

P_i von execute

Inkrementieren bei Compute

Dekrementieren bei Compute

Move

load

SP als Destination

hs₂ oder hs₇ bei Interrupt

$$IN_{\text{chen pre}} = (E \cdot S_0 \cdot \overline{S_1}) \cdot$$

$$(NB \cdot I_{31} \cdot I_{30} \cdot \overline{I_{26}} \cdot I_{25} +$$

$$h_0 \overline{h_1} \overline{h_2})$$

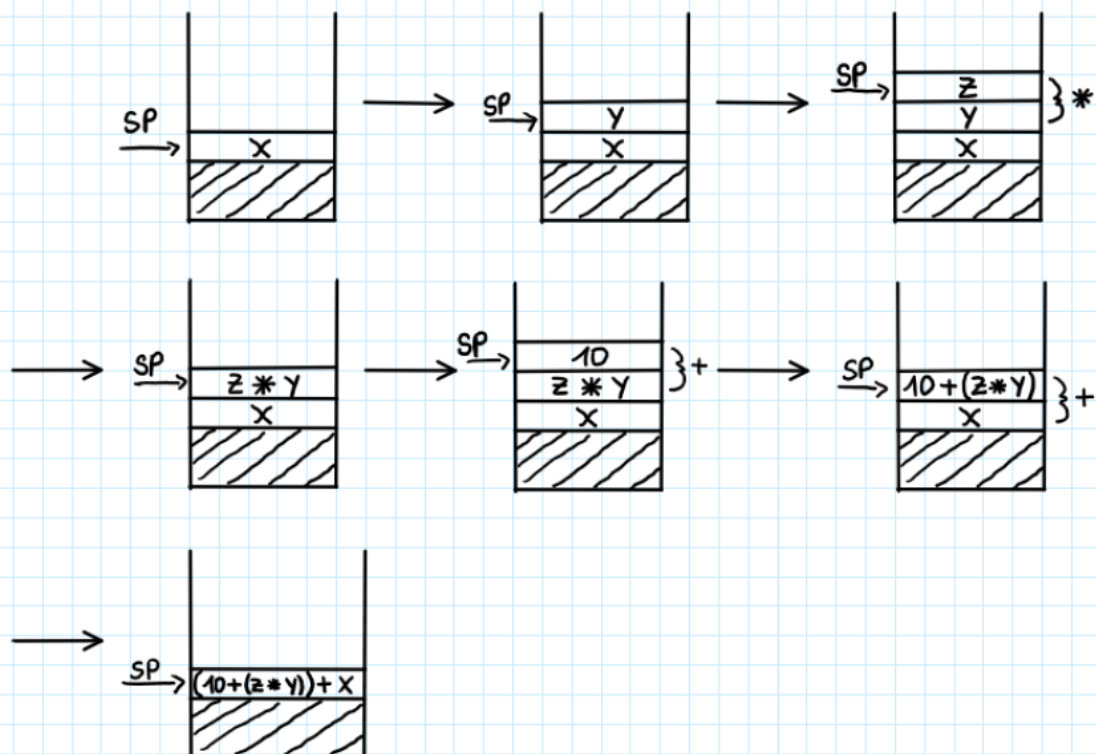
P_i von execute

Syscall im Normalbetrieb

hs₁ bei Interrupt

2

$$X = (X + ((Y \cdot Z) + 10))$$



1	LOAD ACC bds	$ACC := M(\langle bds \rangle) = x$
2	STORE ACC SP	$M(\langle SP \rangle) := ACC = x$
3	ADDI SP 1	$\langle SP \rangle := \langle SP \rangle + 1$
4	LOAD ACC bds+1	$ACC := M(\langle bds+1 \rangle) = y$
5	STORE ACC SP	$M(\langle SP \rangle) := ACC = y$
6	ADDI SP 1	$\langle SP \rangle := \langle SP \rangle + 1$
7	STORE z SP	$M(\langle SP \rangle) := z$
8	ADDI SP 1	$\langle SP \rangle := \langle SP \rangle + 1$
9	SUB SP 1	$\langle SP \rangle := \langle SP \rangle - 1$
10	LOAD SP ACC	$ACC := M(\langle SP \rangle) = z$
11	SUB SP 1	$\langle SP \rangle := \langle SP \rangle - 1$
12	LOADSP W ₁	$W_1 := M(\langle SP \rangle) = y$
13	MUL ACC W ₁	$ACC := ACC * W_1 = z * y = 15$
14	STORE ACC SP	$M(\langle SP \rangle) := ACC = 15$
15	ADDI SP 1	$\langle SP \rangle := \langle SP \rangle + 1$
16	STORE 10 SP	$M(\langle SP \rangle) := 10$
17	ADDI SP 1	$\langle SP \rangle := \langle SP \rangle + 1$
18	SUB SP 1	$\langle SP \rangle := \langle SP \rangle - 1$
19	LOAD SP ACC	$ACC := M(\langle SP \rangle) = 10$
20	SUB SP 1	$\langle SP \rangle := \langle SP \rangle - 1$
21	LOADSP W ₁	$W_1 := M(\langle SP \rangle) = 15$
22	ADD ACC W ₁	$ACC := ACC + W_1 = 10 + 15 = 25$
23	STORE ACC SP	$M(\langle SP \rangle) := ACC = 25$
24	ADDI SP 1	$\langle SP \rangle := \langle SP \rangle + 1$
25	SUB SP 1	$\langle SP \rangle := \langle SP \rangle - 1$
26	LOAD SP ACC	$ACC := M(\langle SP \rangle) = 25$
27	SUB SP 1	$\langle SP \rangle := \langle SP \rangle - 1$
28	LOADSP W ₁	$W_1 := M(\langle SP \rangle) = x$
29	ADD ACC W ₁	$ACC := ACC + W_1 = 27$
30	STORE ACC bds	$M(\langle bds \rangle) := ACC$

c

- 1.) Maximale Anzahl an Teilergebnissen: n
 $(x_1 \circ (x_2 \circ (\dots (x_{n-1} \circ x_n) \dots)))$
- 2.) Minimale Anzahl an Teilergebnissen: 2
 $(\dots (x_n \circ x_{n-1}) \circ x_{n-2}) \dots) \circ x_1$

Überlauf nur bei unterschiedlichen Vorzeichen,
ansonsten kann $x \leq y$ über $x - y \leq 0$ geprüft werden

1	LOAD ACC 11	ACC = y	
2	LOAD W1 10	W1 = x	
3	ANDI ACC 10...0 ₂	ACC = y ∧ 10...0 ₂	ist y negativ
4	ANDI W1 10...0 ₂	W1 = x ∧ 10...0 ₂	ist x negativ
5	SUB ACC W1	ACC = ACC - W1	
6	Jump=3	PC = PC + 3	wird ausgeführt, wenn beide selben Vorzeichen haben
7	Jump<6	PC = PC + 6	wird ausgeführt, wenn $ACC - W1 < 0$, also $y < 0$ und $x > 0$ ist, wegen $ACC = 10...0_2$ und $W1 = 0$
8	Jump>7	PC = PC + 7	wird ausgeführt, wenn $ACC - W1 > 0$, also $y > 0$ und $x < 0$ ist, wegen $ACC = 0$ und $W1 = 10...0_2$
9	LOAD ACC 10	ACC = x	dieser Teil wird ausgeführt, wenn Vorzeichen gleich sind, also kann über $x - y \leq 0$ geprüft werden
10	LOAD W1 11	W1 = y	
11	SUB ACC W1	ACC = ACC - W1	
12	Jumpz3	PC = PC + 3	
13	LOADI ACC 1	ACC = 1	
14	Jumpz	PC = PC + 2	
15	LOADI ACC 0	ACC = 0	
16	END		