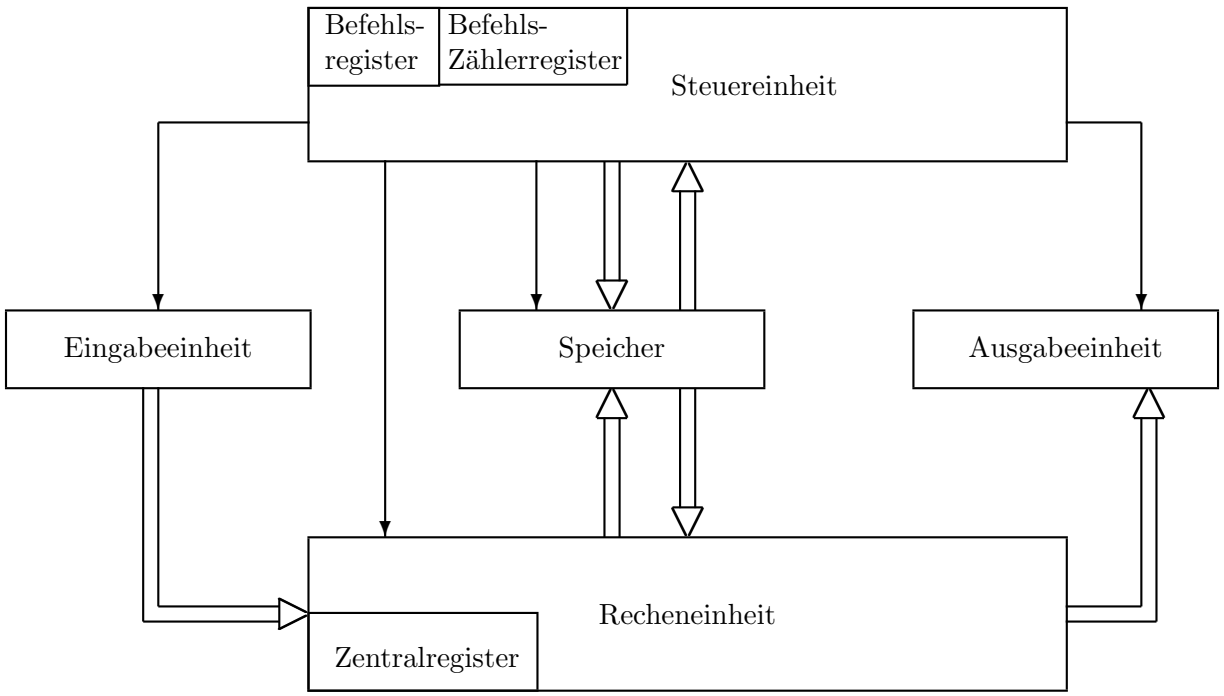
Rechnermodell nach J. von Neumann



Literatur

Programmiersprache C++ (C++11/14)

B. Stroustrup: *Programming: Principles and Practice Using C++. Second Edition.* Addison-Wesley 2014.

B. Stroustrup: *The C++ Programming Language. Fourth Edition.* Addison-Wesley 2013.

U. Breymann: *Der C++ Programmierer.* Hanser 2015.

N. Josuttis: *The C++ Standard Library - A Tutorial and Reference.* Addison-Wesley 2012.

Programmiersprache C (ANSI C)

B. W. Kernighan, D. M. Ritchie: *Programmieren in C. Zweite Ausgabe, ANSI C.* Hanser Verlag 1990.

P. J. Plauger: *The Standard C Library.* Prentice-Hall 1992.

Online-Referenzmaterial

- `http://cppreference.com` (C/C++)
- `http://cplusplus.com/reference` (C++)

Der hypothetische Rechner M

1. Aufbau:
Der Rechner besteht aus dem Zentralregister ZR und $N+1$ Speicherzellen $S\,0$,...,SN, die sowohl Zahlen als auch Befehle beinhalten können.
Schreibweise: $\langle Sn\rangle$:=Inhalt der Speicherzelle Sn.

2. Befehlsvorrat:

Ein-/Ausgabe:

READ	Einlesen der nächsten auf dem Eingabemedium zur Verarbeitung anstehenden Zahl ins ZR
PRINT	Ausgabe von $\langle ZR\rangle$ auf das Ausgabemedium
PRINT(n)	Ausgabe von $\langle Sn\rangle$ auf das Ausgabemedium
WRITE "... "	Ausgabe des Textes ... auf das Ausgabemedium ... darf den Textbegrenzer " nicht enthalten

Transporte:

STORE(n)	$\langle ZR\rangle$ nach Sn kopieren
RECALL(n)	$\langle Sn\rangle$ nach ZR kopieren
CHANGE(n)	$\langle ZR\rangle$ und $\langle Sn\rangle$ austauschen

Arithmetik:

ADD(n)	$\langle ZR\rangle + \langle Sn\rangle$ nach ZR bringen
SUB(n)	$\langle ZR\rangle - \langle Sn\rangle$ nach ZR bringen
MULT(n)	$\langle ZR\rangle * \langle Sn\rangle$ nach ZR bringen
DIV(n)	$\langle ZR\rangle / \langle Sn\rangle$ nach ZR bringen, falls $\langle Sn\rangle \neq 0$
ADD, MULT	analog, wobei hier auch als zweiter Operand statt Sn das ZR fungiert

Funktionen:

SIGN	$\text{sgn}(\langle ZR\rangle)$	(Vorzeichen) nach ZR bringen
ABS	$\lvert\langle ZR\rangle\rvert$	(Betrag) nach ZR bringen
ENTIER	$[\langle ZR\rangle]$	(größte ganze Zahl kleiner gleich) nach ZR bringen
SQRT	$\sqrt{\langle ZR\rangle}$	(Quadratwurzel) nach ZR bringen, falls $\langle ZR\rangle \geq 0$
CONST(x)	die Zahl x im ZR erzeugen	

Verzweigungen:

GOTO(n)	Fortführung des Programmablaufs mit dem Befehl in Zelle Sn	
LESS(n)	falls $\langle ZR\rangle < 0$	Fortführung des Programmablaufs mit dem Befehl in Zelle Sn
GREATER(n)	falls $\langle ZR\rangle > 0$	
EQUAL(n)	falls $\langle ZR\rangle = 0$	

Programmsteuerung:

START(n)	Dieser Befehl muß in Zelle $S0$ stehen und definiert den Programmanfang; erster Programmbefehl steht in Zelle Sn
STOP	beendet den Programmablauf
NOP	Leeranweisung; Register und Speicherzellen bleiben unverändert

Beispiel 1 (Heronsche Formel)

a, b, c: Seiten eines Dreiecks
$F = \sqrt{s(s-a)(s-b)(s-c)}$ mit $s = (a+b+c)/2$: Dreiecksfläche
Eingabe: a,b,c Ausgabe: F

Maschinenprogramm:

Adr.	*Befehl*	*ZR*	*S1*	*S2*	*S3*	*S4*	*Kommentar*
S0	START(5)						
S1							Speicherplatz für a
S2							" b
S3							" c
S4							" s
S5	READ	a					read(a)
S6	STORE(1)		a				
S7	READ	b					read(b)
S8	STORE(2)			b			
S9	READ	c					read(c)
S10	STORE(3)				c		
S11	ADD(1)	$c+a$					
S12	ADD(2)	$c+a+b$					
S13	STORE(4)					$c+a+b$	
S14	CONST(2)	2					
S15	CHANGE(4)	$a+b+c$				2	
S16	DIV(4)	$(a+b+c)/2$					
S17	STORE(4)					s	
S18	SUB(1)	$s-a$					
S19	CHANGE(2)	b		$s-a$			
S20	SUB(4)	$b-s$					
S21	MULT(2)	$(b-s)(s-a)$					
S22	CHANGE(3)	c			$(b-s)(s-a)$		
S23	SUB(4)	$c-s$					
S24	MULT(3)	$(c-s)(b-s)\cdot(s-a)$					
S25	MULT(4)	$s(s-a)(s-b)\cdot(s-c)$					
S26	SQRT	F					
S27	WRITE "F = "						
S28	PRINT						Ausgabe von F
S29	STOP						

C++-Programm in Datei heron.cpp:

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
  double a,b,c,s,F;
  cout << "a b c: ";
  cin >> a >> b >> c;
  s = (a+b+c)/2.0;
  F = sqrt(s*(s-a)*(s-b)*(s-c));
  cout << "F=" << F << endl;
  return 0;
}
```

Übersetzen: `c++ heron.cpp` (oder: `clang++ heron.cpp`)
Starten: `./a.out`

Beispiel 2 (Berechnung des arithmetischen Mittels aus Messdaten)

Messdaten: $x_k \geq 0$, $k = 1, \ldots, n$ mit $n \in \mathbb{N}$, z.B. aus einem physikalischen Experiment
Eingabe: x_1, x_2, …, x_n, -1 ⟵ Endekennung
Die Messdaten können nicht gespeichert werden.

Ausgabe: $\bar{x} = \frac{1}{n} \sum_{k=1}^{n} x_k$

Maschinenprogramm:

Adr.	*Befehl*	*ZR*	*S1*	*S2*	*Kommentar*
S0	START(6)				
S1					$k-1$
S2					$s_{k-1} := \sum_{i=1}^{k-1} x_i$
S3					nicht belegt
S4					“
S5					“
S6	CONST(0)	0			Initialisieren
S7	STORE(1)		0		“
S8	STORE(2)			0	“
S9	READ	x_k			Schleifenanfang
S10	LESS(17)				
S11	ADD(2)	$s_{k-1} + x_k$			
S12	STORE(2)			s_k	
S13	CONST(1)	1			
S14	ADD(1)	$(k-1)+1$			
S15	STORE(1)		k		
S16	GOTO(9)				Schleifenende
S17	RECALL(2)	s_n	n	s_n	
S18	DIV(1)	s_n/n			
S19	WRITE “MW=“				MW ausgeben
S20	PRINT				“
S21	STOP				

Die im Kommentar angegebene Speicherbelegung (S1-S2) bezieht sich auf die Situation nach Ausführung der Anweisung S9.

C++-Programm in Datei mittelw.cpp:

```
#include <iostream>
using namespace std;

int main()
{
  int k=0;
  double x,s=0;
  cout << "x1 ... xn -1: ";
  cin >> x;
  while(x>=0) {
    s=s+x;
    k=k+1;
    cin >> x;
  }
  cout << "MW=" << s/k << endl;
  return 0;
}
```

Übersetzen: `c++ mittelw.cpp` (oder: `clang++ mittelw.cpp`)
Starten: `./a.out`