

MOPS – Modellrechner mit Pseudoassembler

Computer sind hochkomplexe Geräte, deren Funktion sich dem Einzelnen nur ansatzweise und dann nur auf einem vergleichsweise hohen Abstraktionsgrad erschließt. Dennoch ist es gerade für Elektrofachkräfte nützlich, grundlegende Funktionsprinzipien eines Rechners zu kennen und zu verstehen. Der Einsatz moderner Darstellungsmittel erweist sich dabei als hilfreich.

wurden bereits ein Jahrzehnt zuvor von Konrad Zuse entwickelt. Die Prinzipien nach denen ein VNR aufgebaut ist, sind in [1, S. 539] wie folgt formuliert:

1. Der Rechner besteht aus fünf Funktionseinheiten, dem Steuerwerk, dem Rechenwerk, dem Eingabewerk, und dem Ausgabewerk (Bild 2).
2. Die Struktur des VNR ist unabhängig von den zu bearbeitenden Problemen. Zur Lösung eines Problems muss von außen eine Bearbeitungsvorschrift, das Programm eingegeben und im Speicher abgelegt werden. Ohne dieses Programm ist die Maschine nicht arbeitsfähig.
3. Programme, Daten, Zwischen- und Endergebnisse werden in demselben Speicher abgelegt.
4. Der Speicher ist in gleichgroße Zellen unterteilt, die fortlaufend durchnummeriert sind. Über die Nummer (Adresse) einer Speicherzelle kann deren Inhalt abgerufen oder verändert werden.
5. Aufeinanderfolgende Befehle eines Programms werden in aufeinanderfolgenden Speicherzellen abgelegt. Das Ansprechen des nächsten Befehls geschieht vom Steuerwerk aus durch Erhöhen der Befehlsadresse um Eins.

6. Durch Sprungbefehle kann von der Bearbeitung der Befehle in der gespeicherten Reihenfolge abgewichen werden.

7. Es gibt zumindest

- arithmetische Befehle wie Addieren, Multiplizieren, Konstanten laden usw.
- logische Befehle wie Vergleiche, logisches nein, und, oder usw.
- Transportbefehle, z. B. vom Speicher zum Rechenwerk und für die Ein-/Ausgabe
- bedingte Sprünge
- sonstige Befehle wie Schieben, Unterbrechen, Warten usw.

Alle diese Befehle können in verschiedenen Adressierungsarten ausgeführt werden, ...

8. Alle Daten (Befehle, Adressen usw.) werden binär codiert. Geeignete Schaltwerke im Steuerwerk und an anderen Stellen sorgen für die richtige Entschlüsselung.

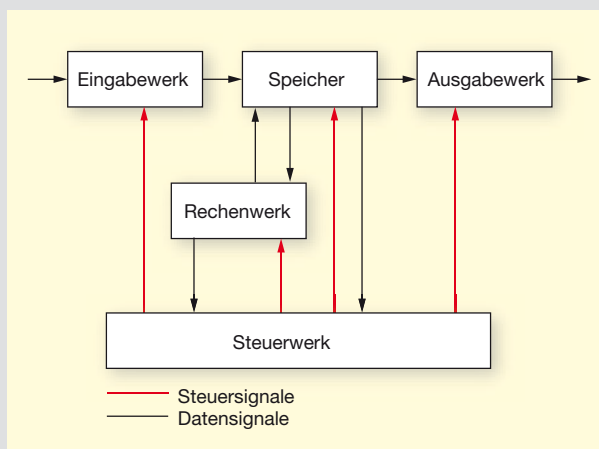
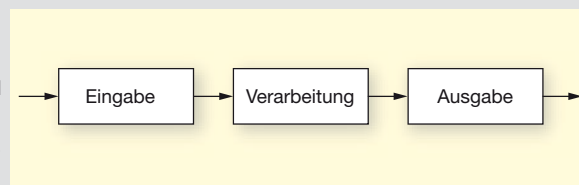
Die grafische Darstellung der Struktur und die formulierten Prinzipien, vermitteln eine erste, aber doch recht abstrakte Vorstellung von der Funktion eines VNR.

Es hat daher in der Vergangenheit nicht an Versuchen gefehlt, anschaulichere Darstellungen für diese – ohne jeden Zweifel – komplizierten und komplexen Sachverhalte zu finden. Diese Versuche sind aber ausnahmslos im akademischen Bereich angesiedelt und daher in der beruflichen Ausbildung nur mit Einschränkungen nutzbar.

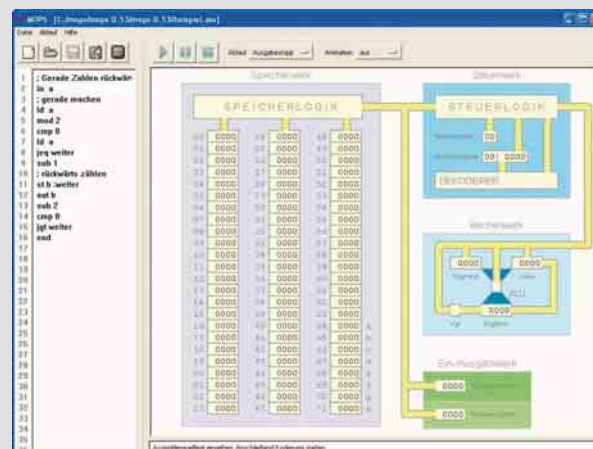
Von-Neumann-Rechner

Bei der Erklärung eines Rechners (Bild 1) wird vielfach vom EVA-Prinzip (EVA – Eingabe-Verarbeitung-Ausgabe) ausgegangen. Für gewöhnlich folgt dann noch ein Hinweis auf den Von-Neumann-Rechner (VNR) oder eben die Von-Neumann-Architektur. Das im Jahre 1946 von John von Neumann vorgeschlagene Konzept zur Strukturierung eines universell einsetzbaren Rechners findet nach wie vor in der Praxis Anwendung. Es wird heute als Modell zur Erklärung der Struktur und Funktion eines Rechners vorzugsweise genutzt. Einige der im VNR zusammengefassten Lösungsansätze

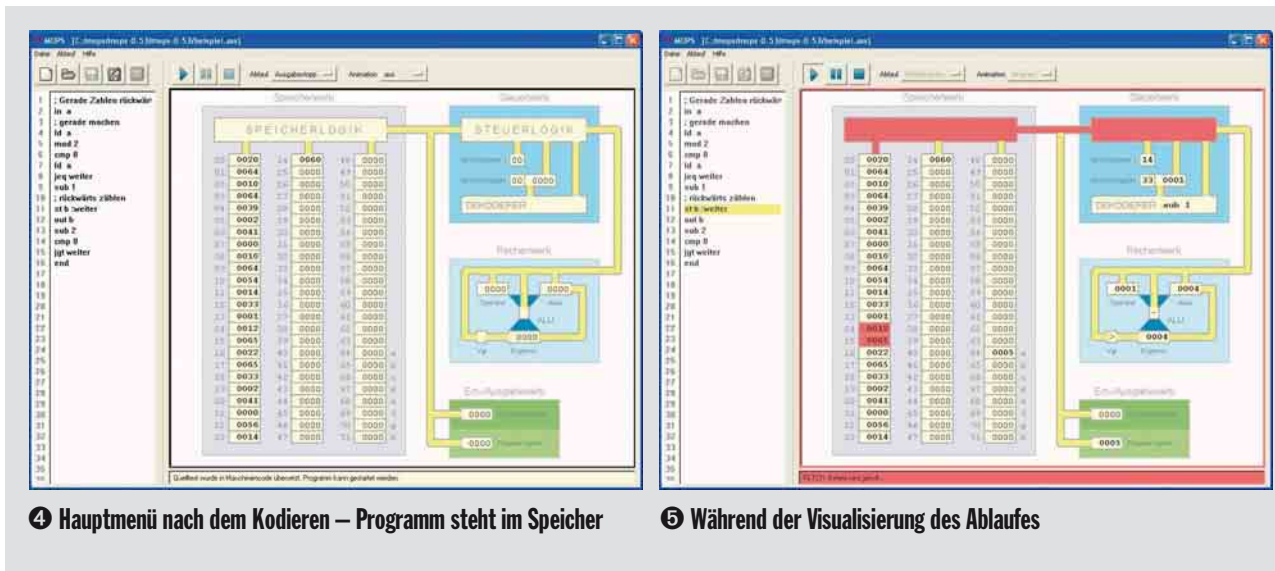
1 EVA-Prinzip – grundlegende Modellvorstellung zum Aufbau und Funktion eines informationsverarbeitenden Systems



2 Funktionseinheiten eines Von-Neumann-Rechners [1, S. 540]



3 Modellrechner mit Pseudoassembler – Hauptmenü nach der Programmerrfassung



④ Hauptmenü nach dem Kodieren – Programm steht im Speicher

⑤ Während der Visualisierung des Ablaufes

Interessanter Lösungsansatz

Einen neuen interessanten Lösungsansatz hat M. Haase mit dem **Modellrechner** mit **Pseudoassembler** (Mops) zur Unterstützung des Informatikunterrichts an einem Gymnasium entwickelt. Dieser Modellrechner ist an die Von-Neumann-Architektur angelehnt und enthält darüber hinaus einen Quelltext-Editor und einen Assembler. Die Lehr- und Lernhilfe ist frei verfügbar und kann kostenlos von der Homepage www.viktorianer.de des Autors heruntergeladen werden. Die etwa 3 MByte große Datei braucht danach lediglich entpackt zu werden. Eine Installation ist nicht erforderlich. Das Programm kann auf Windows-Rechnern ab Windows 98 genutzt werden. Da Python als Programmiersprache verwendet wurde, ist es unter bestimmten Voraussetzungen auch unter Linux und MacOS lauffähig. Hinweise zum Programm und dessen Handhabung liefert ein rund 10 Seiten umfassendes Handbuch im PDF-Format.

Besonderheiten

Gegenüber den durch von Neumann formulierten Prinzipien weist der Modellrechner mit Pseudoassembler einige didaktisch und programmtechnisch bedingte Besonderheiten auf. Die Wichtigsten davon sind:

1. Bei dem 72 Speicherzellen umfassenden Hauptspeicher sind von vornherein 8 als Datenspeicher vorgesehen und können unter Verwendung von Variablen (a ... h) statt durch Speicheradressen ohne Variablendeklaration adressiert werden. Diese Vorgehensweise vereinfacht die Assemblerprogrammierung und hilft beim Verständnis von Abläufen.

Tafel ① Eingebauter Befehlssatz

Befehl	Code	Funktion
ld adr	10	load: Lade den Wert an der Adresse <i>adr</i> in den Akku
ld val	11	load: Lade den Wert <i>val</i> in den Akku
st adr	12	store: Speichere den Wert des Akkus an der Adresse <i>adr</i>
in adr	20	input: Schreibe den Wert des Eingaberegisters an die Adresse <i>adr</i>
out adr	22	output: Schreibe den Wert an der Adresse <i>adr</i> ins Ausgaberegister
out val	23	output: Schreibe den Wert <i>val</i> ins Ausgaberegister
add adr	30	add: Addiere den Wert an der Adresse <i>adr</i> zum Akku
add val	31	add: Addiere den Wert <i>val</i> zum Akku
sub adr	32	subtract: Subtrahiere den Wert an der Adresse <i>adr</i> vom Akku
sub val	33	subtract: Subtrahiere den Wert <i>val</i> vom Akku
mul adr	34	multiply: Multipliziere den Wert an der Adresse <i>adr</i> mit dem Akku
mul val	35	multiply: Multipliziere den Wert <i>val</i> mit dem Akku
div adr	36	divide: Dividiere den Akku durch den Wert an der Adresse <i>adr</i>
div val	37	divide: Dividiere den Akku durch den Wert <i>val</i> – nur ganzzahliger Teil
mod adr	38	modulo: Rest bei Division des Akkus durch den Wert an der Adresse <i>adr</i>
mod val	39	modulo: Rest bei Division des Akkus durch den Wert <i>val</i>
cmp adr	40	compare: Vergleiche den Akkuinhalt mit dem Wert an der Adresse <i>adr</i>
cmp val	41	compare: Vergleiche den Akkuinhalt mit dem Wert <i>val</i>
jmp tar	50	jump: Springe zum Zielpunkt <i>tar</i> (Zeilennummer oder Marke)
jlt tar	52	jump if less than: Springe ..., wenn bei cmp der Akkuinhalt kleiner war
jeq tar	54	jump if equal: Springe ..., wenn bei cmp der Akkuinhalt gleich war
jgt tar	56	jump if greater than: Springe ..., wenn bei cmp der Akkuinhalt größer war
end	60	end: Beendet ein Programm

Programmieren und Kodieren

Um den Modellrechner in Aktion zu erleben, muss zunächst über den Quelltext-Editor ein Programm eingegeben werden (Bild ③). Der Befehlssatz des Modellrechners umfasst derzeit lediglich 15 Befehle (Tafel ①). Diese Beschränkung erlaubt einen schnellen Überblick und ist auf den Zweck des Programms, die Simula-

tion und Visualisierung von Abläufen in einem Von-Neumann-Rechner ausgerichtet. Als Ideenquelle zum Entwurf eines eigenen Programms, kann das in der Anleitung angegebene Beispiel genutzt werden. Nachdem das „Assembler-Programm“ mit dem Editor eingegeben wurde, erfolgt im nächsten Arbeitsschritt dessen Übersetzung in „Maschinencode“. Dazu wird die Funktion Kodieren im Ablauf-Menü aktiviert. Wahlweise kann dazu auch die F4-Taste genutzt werden. Bei diesem Arbeitsschritt sieht man, wie der Quelltext Zeile für Zeile übersetzt wird und die Speicherzellen Zeile um Zeile mit dem Maschinencode (Bild 4) gefüllt werden. Fehler in der Notation werden an den betreffenden Stellen angezeigt.

Ablauf und Animationen

Wenn das Programm im Speicher steht, kann dessen Abarbeitung gestartet werden. Bezüglich der Visualisierung des Ablaufes (Bild 5) bestehen verschiedene Möglichkeiten, um Haltepunkte – also Punkte nach deren Abarbeitung automatisch angehalten wird – zu setzen. Bei den Animationen kann zwischen drei verschiedenen Geschwindigkeiten gewählt werden. Die grafische Darstellung des Rechners und die Beobachtung der Abläufe ermöglichen und vertiefen das Verständnis der durch von Neumann formulierten Prinzipien.

Fazit

Der Modellrechner mit Pseudoassembler ist eine tolle Idee. Trotz des noch recht frühen Versionsstandes (V 0.53) ist diese Lehr- und Lernhilfe für die Praxis geeignet. Seitens des Autors ist das Angebot zwar in erster Linie an Lehrende gerichtet, aber dessen Nutzung kann auch jedem interessierten Lernenden empfohlen werden. Allerdings sollte man schon etwas Zeit einplanen, um sich mit Möglichkeiten dieses Programms zu beschäftigen. Interessant und hervorhebenswert ist die gelungene Verbindung der Visualisierung von Abläufen in einem VNR mit der Vermittlung von Anfangskenntnissen in der Assemblerprogrammierung.

H. Möbus

Literatur

[1] Claus, V.; Schwill, A.: Schüler-Duden – Die Informatik; Ein Sachlexikon für die Schule. Dudenverlag Mannheim/Wien/Zürich 1991.

Technische Entwicklungen in der Elektrotechnik

Teil 1: Gleichspannung und Wahl der Nennspannung

In dieser neuen Beitragsreihe wird der Frage nachgegangen, warum sich die Elektrotechnik gerade so entwickelt hat, wie wir sie heute kennen.

Anliegen der Beitragsfolge

Die Elektrofachkraft von heute muss zahlreiche Normen und Vorschriften kennen, sie hat die handwerklichen Fertigkeiten zu beherrschen und muss sich oft auch mit wirtschaftlichen Zwängen auseinandersetzen. In diesem Zusammenhang kommt das physikalische Verständnis, die Beantwortung der Frage „warum ist das so?“ oft sehr kurz.

Die Beitragsfolge soll diese meist nicht gestellt Frage in den Vordergrund stellen. Wenn auch der Geschichte der Elektrotechnik dabei kritisch nachgegangen wird, so soll sie doch nicht vordergründig abgehandelt werden. Es besteht vielmehr die Absicht, die Entwicklung bezüglich der physikalischen Zwänge zu analysieren. Es soll gefragt werden, warum die gewählten Lösungen in der Entstehungszeit gerade so vorteilhaft waren, welche Probleme damit verbunden waren, ob es Alternativen gegeben hätte und ob sie noch heute Bestand haben können bzw. ob sie auch jetzt noch vorteilhaft sind.

Am Anfang war die Gleichspannung

Eine Energieversorgung mit Gleichspannung ist gegenwärtig auf Sonderfälle im gewerblichen Bereich beschränkt. Eine Betrachtung der Gleichspannungsversorgung hilft hier, einige Probleme zu behandeln, die heute keine Aufmerksamkeit finden. Es lohnt sich jedoch darüber nachzudenken, warum die Verbraucher parallel geschaltet werden und nicht in Reihe und wie die Nennspannung für die Verbraucher gewählt wurde.

Die Entwicklung des Akkumulators kann den Beginn der technischen Verwertung des elektrischen Stromes markieren. Andere Stromquellen standen zu Beginn des

19. Jahrhunderts noch nicht zur Verfügung. Anwendungen der elektrochemisch erzeugten Elektroenergie ließen dann nicht lange auf sich warten. Man konnte mit einem Widerstandsdraht Wärme erzeugen, die elektrische Bogenlampe wurde erfunden und die ersten Modelle von Gleichstrommotoren wurden entwickelt. Auch nachrichtentechnische Einrichtungen, z. B. Telegrafen wurden entwickelt. Wir wollen uns dabei im Klaren sein, dass das, was so in zwei Sätzen aufgeschrieben wurde, ein schwieriger Prozess war, der sich über etwa drei Jahrzehnte hingezogen hat.

Der eigentliche technische Durchbruch wurde eingeleitet, als man entdeckte, dass sich die Wirkung des Gleichstrommotors auch umkehren ließ, in dem der Motor (z. B. mit einer Dampfmaschine) angetrieben wurde und so zum Generator wurde. Dies brachte eine Reihe von Vorteilen mit sich:

- Es konnten höhere Gleichspannungen erzeugt werden.
- Die Erzeugung größerer Leistungen wurde möglich.
- Die zeitliche Begrenzung der Energielieferung durch die Erschöpfung der chemischen Energie im Akkumulator entfiel.

Jetzt konnten erste kleinere Kraftwerke entstehen, die der örtlichen Stromversorgung dienten. Es wurde z. B. von einem solchen Kraftwerk berichtet [1], dass in dem Ort Bühlau bei Dresden etwa ein Dutzend Bogenlampen und den Motor der örtlichen Mühle speiste.

Im Übrigen soll nicht der Eindruck entstehen, dass Energieversorgung auf Gleichstrombasis nur eine Angelegenheit des 19. Jahrhunderts gewesen sei. In Teilen von Leipzig gab es bis in die 1960er Jahre Gleichstromversorgung. Auch aus der Umgebung dieser Stadt wird berichtet: „An der Elster wurde 1901 das heute noch bestehende Wasserkraftwerk in Betrieb genommen, dessen drei Francisturbinen die Brauerei und später Teile von Lützenscha bis in die 60er Jahre mit Gleichstrom versorgten“. Das Gas- und E-Werk Singen gibt die Einstellung der Gleichstromversorgung ab dem 1.12.1962 an. Und als letztes Beispiel sei genannt: „Im Laufe des Jahres (1927) stellte die Berliner Elektrizitätsversorgung AG. (BEWAG) ihre städtischen Stromverteilernetze von