Das Computer-Demo-Modell

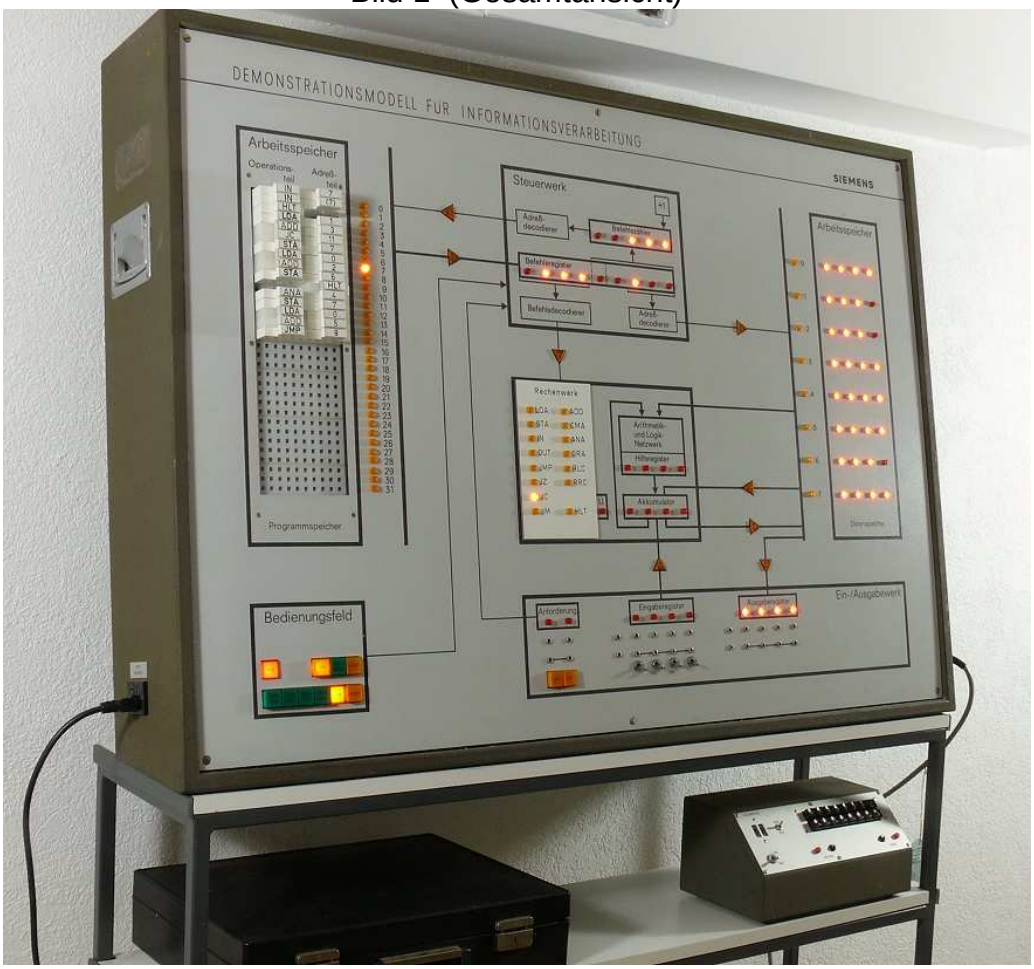
Überarbeitete Version des Originaltextes (Fortsetzung folgt)

Mit Hilfe des „Demonstrationsmodell für Informationsverarbeitung“ aus dem Jahre 1974 kann man auch heute noch die elementaren Vorgänge in Rechnern verstehen. Das Grundprinzip hat sich nicht geändert.
Schüler, die im **technikum29** über dieses Modell referieren, müssen natürlich nicht die gesamte Bedienungsanleitung durcharbeiten. Mit der Befehlsliste vor Ort nach dem Prinzip „learning by doing“ kommt man auch ans Ziel. Bei Unklarheiten: Hier oder im Originaltext nachschlagen!

Manche Begriffe sind etwas veraltet, wir haben diese aber dennoch meistens mit einbezogen, da sie am Modell so beschriftet sind.
Im Prinzip liegt hier ein programmgesteuerter Digitalrechner mit Parallelverarbeitung vor. Die einzelnen Blöcke wurden übersichtlich dargestellt und durch Signalflusslinien verbunden. Alle Vorgänge werden durch insgesamt 126 Lämpchen signalisiert.

Es können 6 verschiedene Betriebsarten gewählt werden. Damit lassen sich Taktzyklen, Befehle und Programmschleifen auch in Zeitlupe beobachten. Wird z.B. ein Befehl (= 4 Zyklen) von Hand ausgelöst, so kann jede einzelne Phase beliebig lange festgehalten werden.

Bild 1 (Gesamtansicht)



Programmierung

Die Programmierung erfolgt in Maschinensprache. Um nicht in binärcodierter Form (d.h. in Nullen und Einsen) programmieren zu müssen, wurde der Programmspeicher in Form eines Buchsenfeldes aufgebaut, in das die einzelnen Befehle durch vorgefertigte Steckerbausteine "eingeschrieben" werden. Diese beschrifteten Stecker lassen die Befehls- bzw. Adressverschlüsselung in ihrer codierten Form erkennen, ohne die Übersichtlichkeit des Programms zu schmälern. Auf diese Weise ist das im Buchsenfeld gesteckte Programm (maximal 32 Befehle) unmittelbar lesbar und kann jederzeit, z.B. beim Testen, durch Umstecken korrigiert werden. Darüber hinaus bleibt die Möglichkeit bestehen, mit Bitsteckern auch rein binär zu programmieren.

Bild 2 (Arbeitsspeicher links)

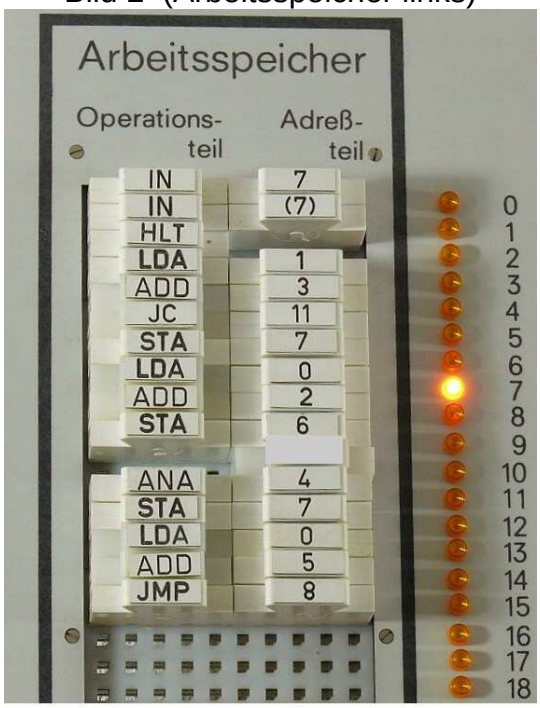
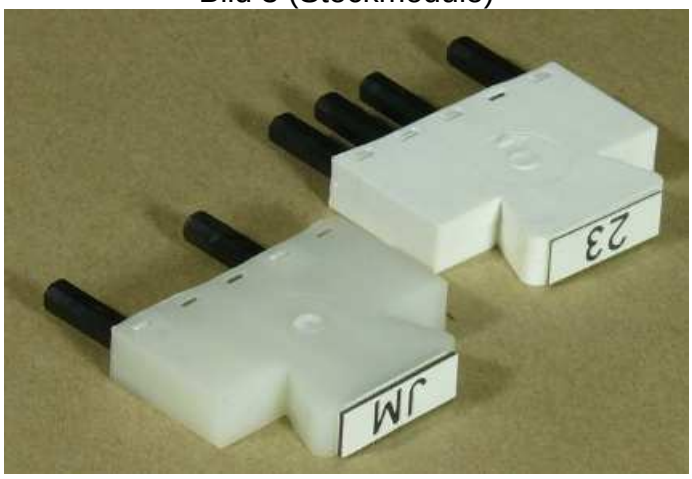
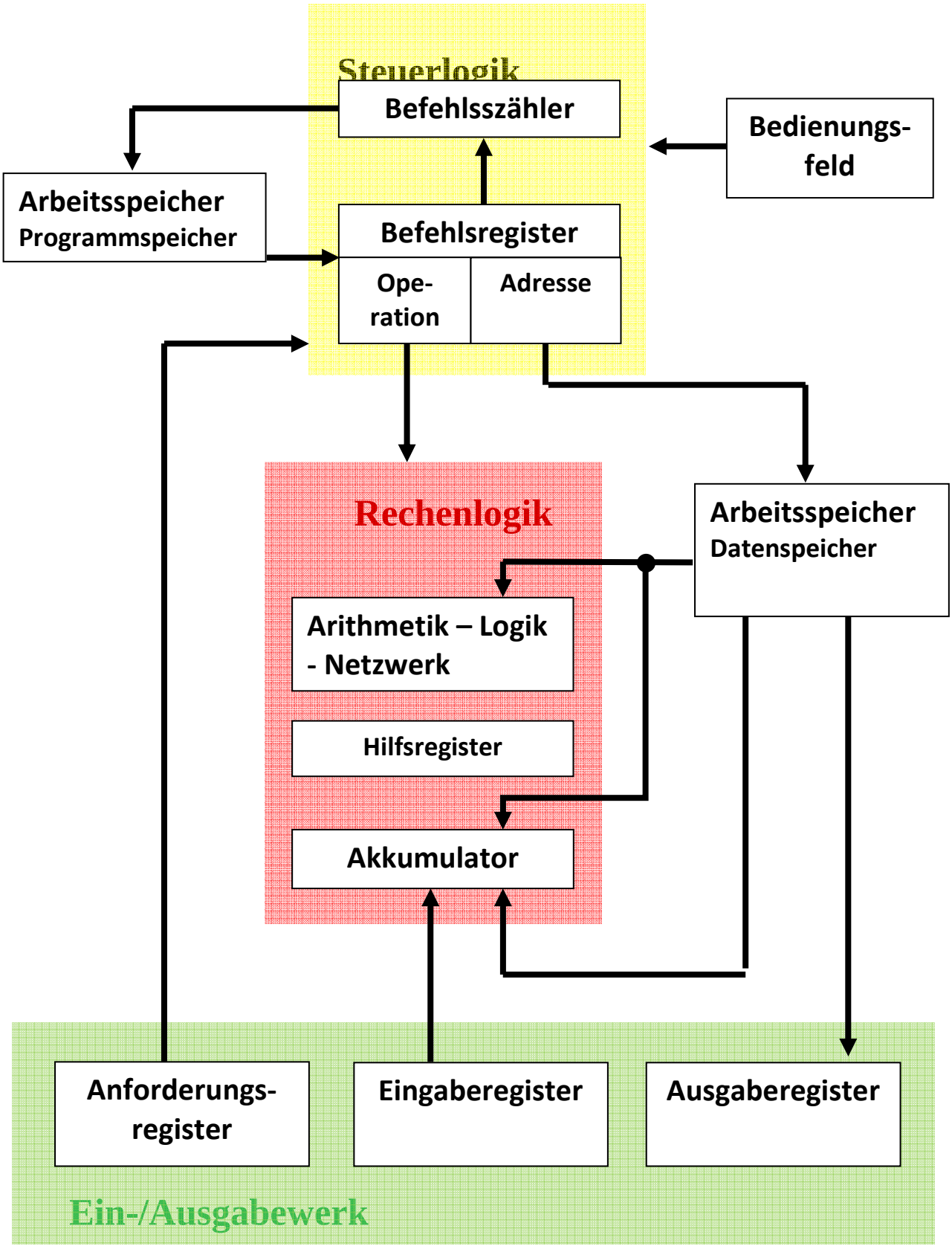


Bild 3 (Steckmodule)



Blockschaltbild



1. Steuerlogik (Steuerwerk)

Befehlszähler:

Der Befehlszähler (5 Bit) enthält die duale Adresse des nächsten zu verarbeitenden Befehls. Er ist damit Ausgangspunkt eines jeden Funktionsablaufs im Rechner. Der Befehlszählerstand wird bei jedem normalen Befehlszyklus mit dem Takt 4 um 1 erhöht, so dass ein Befehl nach dem anderen ablaufen kann, Bei Sprungbefehlen wird der Befehlszähler (mit dem Takt 3) neu eingestellt (gesetzt), um Programmverzweigungen oder Schleifen zu ermöglichen.

Befehlsregister:

Das Befehlsregister (9 Bit) speichert das aktuelle Befehlswort bei seiner Bearbeitung und zeigt die binäre Codierung von Operations- und Adressteil auf (Befehlsformat). Befehlszähler und Befehlsregister ermöglichen den automatischen Ablauf von Programmen. Sie sind der wesentlichste Teil des Steuerwerks.
Rein technisch gesehen schließt das Steuerwerk die Takterzeugung und Taktverteilung mit ein, die jedoch nicht direkt angezeigt werden.

2. Arbeitsspeicher

Programmspeicher:

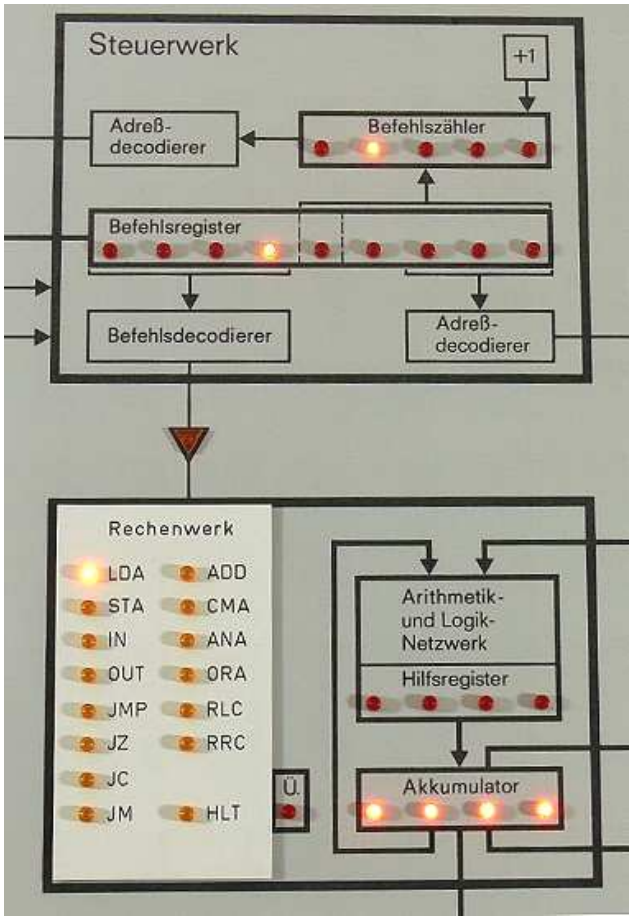
Der Programmspeicher nimmt das Programm (Befehlsfolge) auf. Er ist frei programmierbar durch Stecken von Operations- und Adressbausteinen entsprechend der Gliederung eines Befehlswortes in die Operation und die Operandenadresse (Einadressbefehlssystem!). Die aktuelle Befehlsadresse wird während eines Befehlszyklus in einer Signallampenleiste , welche die Adressleitung symbolisiert, anzeigt.

Datenspeicher:

Der Datenspeicher ist als frei beschreibbarer Halbleiterspeicher ausgebildet. Er beinhaltet 8 Worte zu je 4 Bit zur Aufnahme der Daten. Über Signallämpchen werden Speicherinhalt und -adresse sichtbar gemacht.
Datenspeicher und Programmspeicher sind im Allgemeinen in einem Arbeitsspeicher (Zentralspeicher) zusammengefasst, so dass die Befehlsworte wie die Daten eingelesen (nicht gesteckt) und gespeichert werden. Aus didaktischen Gründen - direkte Lesbarkeit der Programme und klare Unterscheidung - wurde jedoch eine Trennung vorgenommen.

3. Rechenlogik (Rechenwerk):

Im Rechenwerk werden die 4 Bit breiten Daten parallel verarbeitet nach Maßgabe der Befehle. Das Befehlsspektrum wurde möglichst breit gefächert.
Es besteht aus 4 Transferbefehlen für internen und externen Datenverkehr, 4 Sprungbefehlen, einem unbedingten und drei bedingten Sprungbefehlen, 2



arithmetischen Befehlen, 2 logischen Befehlen sowie 2 Verschiebebefehlen. Ein Teil der Befehle ist substituierbar, um eine Adressrechnung bzw. auch Indizierung zu ermöglichen.
Die Befehlsanzeige - jeder aktuelle Befehl leuchtet nach seiner Decodierung in den beiden Lampenleisten auf - wurde aus Gründen der Übersichtlichkeit im Rechenwerk zusammengefasst, obwohl natürlich nicht alle Befehle Rechenwerkbefehle sind. So wird z.B. der unbedingte Sprungbefehl JMP nur im Steuerwerk bearbeitet. Die bedingten Sprungbefehle JM, JZ, JC werden dagegen schon wieder mit Bedingungen aus dem Rechenwerk verknüpft.

Bild 5 (Steuer- und Rechenlogik)

Akkumulator:

Der Akkumulator ist dem Rechenwerk direkt zugeordnet, und dient als Verarbeitungs- und Ergebnisregister. Er nimmt vor der Operation das zu verarbeitende Datenwort auf und enthält nach der Operation das Ergebnis. Sollen demnach zwei Operanden verknüpft werden, so muss zunächst der eine Operand in den Akkumulator gebracht werden - durch den Befehl "LDA w". Der zweite Operand wird nun mit dem Akkumulatorinhalt verknüpft z.B. durch den Befehl "ADD w" oder "AND w". Das Ergebnis wird in den Akkumulator eingetragen und überschreibt damit den ersten Operanden. Auch andere Operationen wie Verschiebebefehle oder der Komplementierungsbefehl, die sich nur auf einen Operanden beziehen, werden mit dem Akkumulatorinhalt durchgeführt, so dass auch hier der Operand zunächst in den Akkumulator geladen werden muss. in diese Operationen muss sich im allgemeinen ein Rücktransfer des Ergebnisses in den Speicher anschließen, durch den Befehl "STA w". Moderne Computer werden mit mehr als einem (4, 8, 16, …) Verarbeitungsregister ausgestattet. Dabei können auch Register untereinander verknüpft werden.

Hilfsregister:

Das Hilfsregister zeigt Ergebnisse am Ausgang des Arithmetik- und Logiknetzwerks auf, bevor die Ergebnisse in den Akkumulator eingeschrieben werden und damit einen der Operanden überschreiben.

Überlauf: Siehe Originaltext

4. Ein- und Ausgabe(-werk)

Über die digitalen Ein/ Ausgaberegister können statische Informationen programmgesteuert eingegeben ("IN w") bzw. ausgegeben ("OUT w") werden (Statische Informationen müssen so lange anstehen, bis sie abgeholt worden sind; dynamische Informationen sind Impulse, die auch dann noch gespeichert werden, wenn sie nicht mehr anstehen). Weitere Informationen: Siehe Originaltext.

Befehlsliste des Siemens Demo-Computers

Transfer-befehle	**LDA** (LAD)	**Laden (Load) des Akkumulators** Der Inhalt der angegebenen Speicherzelle (Adresse) wird in den Akkumulator übertragen
	STA (SPE)	**Speichern (Store)** Der Inhalt des Akkumulators wird in die Arbeitsspeicherzelle (Adresse) übertragen
	IN (EIN)	**Eingabe (Input)** Der Inhalt der Digitaleingabe wird in den Akkumulator und in die Arbeitsspeicherzelle (Adresse) übertragen
	OUT (AUS)	**Ausgabe (Output)** Der Inhalt der Arbeitsspeicherzelle (Adresse) wird in das Digitalausgaberegister übertragen
Arithmetische Befehle (Festpunkt)	**ADD**	**Addition** Der Inhalt der Arbeitsspeicherzelle (Adresse) wird zum Akkumulatorinhalt addiert
	CMA (KPL)	**Komplementieren** Der Akkumulatorinhalt wird komplementiert (2-er Komplement)
Logische Befehle	**ANA** (UND)	**UND (AND)** Der Inhalt des Akkumulators wird mit dem Inhalt der Arbeitsspeicherzelle (Adresse) im Sinne des „logischen UND" verknüpft
	ORA (ODR)	**ODER (OR)** Der Inhalt des Akkumulators wird mit dem Inhalt der Arbeitsspeicherzelle (Adresse) im Sinne des „logischen ODER" verknüpft

Die oben stehenden Befehle sind alle jeweils substituierbar

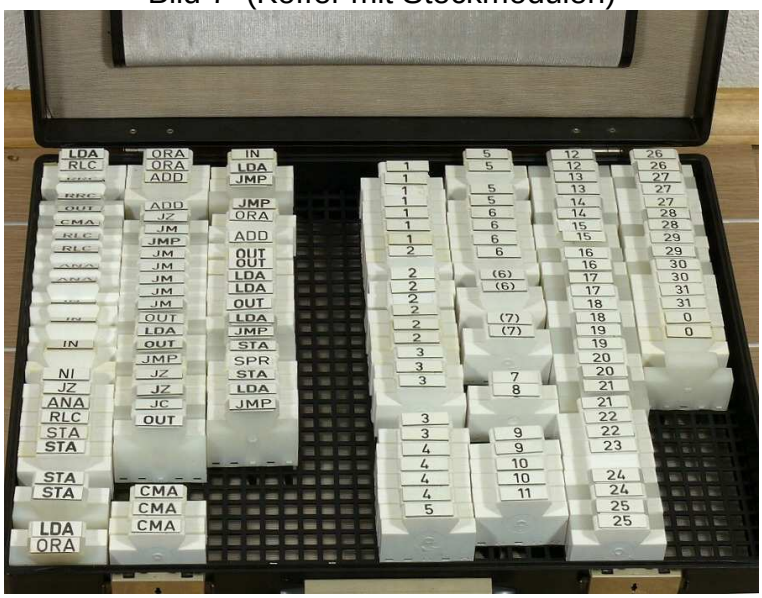
Verschiebe-befehle	**RCL** (VLL)	**Verschiebe logisch links** Der Akkumulatorinhalt wird um 1 Bit nach links verschoben. Von rechts werden Nullen nachgezogen. Die Vorzeichenstelle wird wie jede andere Bitstelle behandelt
	RRC (VLR)	**Verschiebe logisch rechts** Der Akkumulatorinhalt wird um 1 Bit nach rechts verschoben…… siehe oben
Sprung-befehle	**JMP** a (SPR)	**Springe (Jump)** Das Programm wird mit dem Befehlswort a (Programmspeicheradresse) fortgesetzt
	JZ a (SGN)	**Springe, falls Akkumulator gleich Null** (Jump if Zero). Ist der Inhalt des Akkumulators Null, so wird das Programm mit dem Befehlswort a (Programmspeicheradresse) fortgesetzt. Ist der Akku-Inhalt ungleich Null, so wird das Programm mit dem auf den JZ-Befehl folgenden Befehlswort fortgesetzt
	JM a (SAM)	**Springe, falls Akkumulator minus** Ist die Stelle 1 (links) des Akkumulators „1" (negatives Vorzeichen), so wird das Programm mit dem Befehlswort a (Programmspeicheradresse) fortgesetzt. Andernfalls wird es mit dem auf den JM-Befehl folgenden Befehlswort fortgesetzt
	JC a (SUL)	**Springe, falls Überlauf** Ist das Überlaufregister „1", so wird das Programm mit dem Befehlswort a (Programmspeicheradresse) fortgesetzt und der Überlauf gelöscht. Andernfalls wird es mit dem auf den JC-Befehl folgenden Befehlswort fortgesetzt
Organisa-torischer Befehl	**HLT** (STP)	**Stop** Das Programm wird gestoppt

2. Spalte: In Klammern stehen die alten deutschen Begriffe

Codierung des Siemens Demo-Computers

Operationscode		
Mnemotechnischer Code (Assembler)	**Binärcode**	**Bedeutung**
LDA (LAD) **STA** (SPE) **IN** (EIN) **OUT** (AUS)	0001 0010 0011 0100	Transferbefehle
JMP (SPR) **JZ** (SGN) **JM** (SAM) **JC** (SUL)	0101 0110 0111 1000	Sprungbefehle
ADD **CMA** (KPL)	1001 1010	Arithmetische Befehle
ANA (UND) **ORA** (ODR)	1011 1100	Logische Befehle
RLC (VLL) **RRC** (VLR)	1101 1110	Verschiebe-Befehle
HLT (STP)	0000	Stoppbefehl
Adresscode		
Dezimal	**Binär**	**Bedeutung**
0,1,2, ……,7	000,001,010,……..,111	Adressen der Operanten
(0) (1) (2) (3) (4) (5) (6) (7)	10000 10001 10010 10011 10100 10101 10110 10111	Substituierte (indirekt angesprochene) Operantenadressen

Bild 7 (Koffer mit Steckmodulen)



Anwendungen:

Folgende Anwendungen, welche die Begriffsbildung zur Informatik betreffen, sind möglich:

- Binärzeichen, Codes, Umcodierung
- **Stellenwertsysteme:** Dual, Oktal, Umwandlungen
- Informationssicherung (Paritybit….)
- **Zahlendarstellung:** Festpunktzahlen, Gleitkommazahlen, Vorzeichen
- **Befehle:** Operationstypen, Befehlsformate

- **Arithmetik**: Binäraddition, Komplementbildung u. Subtraktion, Multikplikations- und Divisionsalgorithmus

- **Grundfunktionen der Logik** und Booleschen Algebra

- **Grundelemente der Rechnertechnik:** Register, Zähler, Decodierer, Logiknetzwerke, Prinzipien der Speicherung, Rechner-Funktionseinheiten, Aufbau und Organisation einer Zentraleinheit

- **Peripheriegeräte:** Ein- und Ausgabegeräte, Schnittstellen, Datenübertragung

- **Programmierung:** Software, Problemanalyse, Programmstruktur, Quelltext, Maschinenprogramm, Struktur und Ablaufplan, Assembler, Compiler usw.

Dabei gibt es drei Anwendungsgebiete:

1. **Prozessrechner** (automatische Informationsverarbeitung bei technischen Prozessen: Automatisierung)
2. **Kommerzielle Rechner** (Beispiele aus der EDV, viele Daten, wenig Rechnen)
3. **Wissenschaftliche Rechner** (Lösung von umfangreichen mathematischen Aufgaben)

Befehlsabläufe

Die Arbeitsweise einer Datenverarbeitungsanlage (= Zentraleinheit + Ein-/Ausgabegeräte) ist durch das Zusammenwirken von Geräten (Hardware) und Programmen (Software) gekennzeichnet.
Die Zentraleinheit als wesentlichster Bestandteil der "Hardware" stellt ein System von Funktionseinheiten zur Verfügung, das durch ein vom Menschen erstelltes Programm und nur durch ein Programm aktiviert wird. Auf Grund der unveränderlichen technischen Struktur der Zentraleinheit muss sich ein solches Programm - es stellt den veränderlichen Anteil dar - an strenge Gesetzmäßigkeiten halten:
Programme bestehen aus einer Folge von Befehlen. Die Befehle selbst sind Arbeitsanweisungen an den Rechner.
Welche Aussagen müssen Befehle zumindest beinhalten, damit sie verstanden und ausgeführt werden können?

1. Was soll getan werden?
2. Womit soll etwas getan werden?

Entsprechend diesen beiden Befehlselementen wurde ein Befehl in

1. Operationsteil und
2. Adressteil gegliedert.

Mit dem Adressteil wird ein Platz im Speicher gekennzeichnet, dessen Inhalt bei der Operation gemeint ist. Dieser Inhalt, der im Befehl adressierten Zelle, heißt Operand. Man bezeichnet die im Adressteil eines Befehls stehende Adresse auch als Operanden-Adresse.
Solche Befehle heißen z.B. ADD 5 (Addiere Inhalt von 5)
STA 3 (Speichere nach 3)

Welcher Anteil ist in diesen Beispielen Operationsteil?	ADD, STA
Was ist mit den Zahlen angesprochen?	- Adressen
Wo stehen diese Adressen?	im Arbeitsspeicher
Was beinhalten diese Adressen?	Operanden (Daten).

Transferbefehle

Ein Digitalrechner (Zentraleinheit) besteht aus Speicher, Rechenwerk, Steuerlogik und Ein-/Ausgabe. Über Eingabegeräte wird die Zentraleinheit mit Programmen und Daten versorgt, die gespeichert werden müssen. Zur Verarbeitung werden die Daten vom Speicher ins Rechenwerk gebracht. Die hier ermittelten Ergebnisse müssen ebenfalls gespeichert und später ausgegeben werden. Damit sind vier Transferbefehle angesprochen, ohne die keine Verarbeitung im Rechner möglich ist. Eingabe in den Speicher (IN), 2. Ausgabe (OUT), 3. Laden (LDA), 4. Speichern (STA).

Führen Sie nun mit dem Modell folgende Übung durch: Stecken Sie die folgenden Befehle wie angegeben im Arbeitspeicher (Programmspeicher) und führen Sie die darauf folgenden Bedienungen mit dem eingeschalteten Modell durch:

Programmspeicher

IN	5	0
IN	7	1
OUT	5	2
OUT	7	3
LDA	5	4
LDA	7	5
STA	6	6
STA	4	7

Erste Spalte: Operationsteil, zweite Spalte: Adressteil, dritte Spalte Schrittzähler

A) 1) Betriebsart BEFEHL einschalten
2) Taste RÜCKSETZEN drücken
3) Kippschalter auf 1111 einstellen, (alle Lampen im Eingaberegister leuchten),
4) Beobachten Sie den Datenspeicher während Sie auf Taste START drücken!

Vergleichen Sie nun den ersten gesteckten Befehl, "IN 5", mit den Signalleuchten, die aufgeleuchtet sind!
Die Eingabe des Inhalts des Eingaberegisters erfolgt in Adresse 5 des Datenspeichers. (Die gleichzeitige Eingabe in den Akkumulator im Rechenwerk hat praktische Vorteile bei der späteren Programmierung, ist hier aber nicht von grundsätzlicher Bedeutung).
Die Signalleuchte 1 weist darauf hin, dass nun der Befehl "IN 7" folgt.

B) 1) Kippschalter auf 0001 einstellen,
2) Taste START drücken und Datenspeicher beobachten!
Die Eingabe der 0001, die im Eingaberegister steht, ist nun im Datenspeicher an dem Platz, der durch die Adressangabe im Befehl gekennzeichnet war.

C) 1) Taste START drücken und Ausgaberegister beobachten,
2) Taste START erneut drücken!
Die beiden Befehle bewirkten die Ausgabe des Inhalts der im Befehl angegebenen Adressen ins Ausgaberegister

D) 1) Taste START drücken und Akkumulator (Verarbeitungsregister) beobachten,
2) Taste START erneut drücken!
Die Befehle LDA 5 und LDA 7 haben die Inhalte der Adresse 5 und dann der Adresse 7 in den Akkumulator gebracht (geladen).

E) 1) Taste START drücken und Datenspeicher beobachten,
2) Taste START erneut drücken!
Der Inhalt des Akkumulators wurde in den Datenspeicher in die Adressen 6 und 4 transferiert (gespeichert).

Nun stellen wir die Betriebsart SCHRITT ein, drücken Taste RÜCKSETZEN und führen alle Bedienungen nochmals durch. Wo jedoch bisher die Taste START einmal zu betätigen war, muss nun 4mal gedrückt werden, da nun ein Befehl in 4

Taktschritte unterteilt wird. Wir zählen die Taktschritte jeweils bis 4 und beobachten im Schritt 1 und 2 besonders die Steuerlogik. Die Schritte 3 und 4 vergleichen wir mit den vorherigen Ergebnissen.
Wie würden Sie die beiden Begriffe Befehlsbereitstellung und Befehlsausführung zu den Schritten 1 bis 4 zuordnen?
Befehlsbereitstellung = Schritt 1 und 2
Befehlsausführung = Schritt 3 und 4
Die Befehle stehen im Speicher. Es wird aber immer nur ein Befehl nach dem anderen ausgeführt. Die Steuerlogik veranlasst die Befehlsausführung.

Allen Befehlen ist die nun folgende Befehlsbereitstellung gemeinsam:
Das Steuerwerk muss die Befehle lesen und interpretieren. Welche Aufgabe hat dabei der Befehlszähler mit dem angeschlossenen Adressdecodierer?
Der Befehlszähler stößt den Lesevorgang an, indem er den nächsten zu bearbeitenden Befehl adressiert, auswählt, ansteuert. Die gelesene Information wird in das Befehlsregister gebracht.
Vom Befehlsregister aus kann der Befehl auf das Rechenwerk und die anderen Funktionseinheiten wirken. Dazu muss das binäre Befehlswort decodiert werden.
Was bedeutet das?
Der Rechner erkennt (entschlüsselt, decodiert) aus dem Operationscode die betroffene Operation und aus dem dualen Adresscode, welche Adresse gemeint ist und schaltet die zugehörigen Steuerleitungen für die folgende Befehlsausführung.
Die Erhöhung des Befehlszählers im Schritt 4 um 1, damit der nächste Befehl angesteuert werden kann, wird häufig auch noch zur Befehlsbereitstellung gezählt.
Während der Befehlsausführung werden die schon vorher gezeigten Transfer-operationen durchgeführt.

Arithmetische Befehle (gegenüber dem Original stark gekürzt)

Die bekanntesten Verarbeitungsbefehle sind die arithmetischen Befehle. Im Modell vorhanden sind der Additionsbefehl "ADD w" und der Komplementierungsbefehl "CMA", der für die Subtraktion verwendet wird.
Stecken Sie die folgenden Befehle und führen Sie die angegebenen Bedienungen durch!

Programm

IN	2	0
IN	1	1
ADD	2	2

Ausführen:

1) BEFEHL, RÜCKSETZEN, Kippschalter: 0011 (duale 3), START
2) Kippschalter: 0010 (duale 2), START
3) SCHRITT, START, START,
   Beobachten Sie dabei die Befehlsbereitstellung für "ADD 2";
   START, beobachten Sie das Hilfsregister
   START, beobachten Sie den Akkumulator!

Was hat der Befehl "ADD 2" bewirkt?
- Der Inhalt der Adresse 2 wurde zum Akkumulatorinhalt addiert (im Hilfsregister wurde die Summe angezeigt und dann in den Akkumulator eingeschrieben).
Wiederholen Sie die Addition mit anderen Dualzahlen!
Das viermalige Drücken von „START“ bewirkt folgendes:
START 1: Befehl lesen
START 2: Befehl decodieren
START 3: Befehl ausführen
START 4: Das Ergebnis wird vom Hilfsregister in den Akkumulator eingeschrieben (einer der Summanden wird überschrieben). Der Befehlszählerstand wird um 1 erhöht (entspr. der Adresse des folgenden Befehls).

Stecken Sie nun folgende Befehle und führen Sie die angegebenen Bedienungen durch!

Programm

IN	1	0
CMA		1

1) BEFEHL, RÜCKSETZEN
   Kippschalter beliebig einstellen, START
2) SCHRITT, START, START, Befehlsbereitstellung, START, START,

Beobachten Sie die Befehlsausführung für "CMA" im Hilfsregister und Akkumulator.
Was hat der Befehl "KPL" bewirkt?
-- Der Inhalt des Akkumulators wurde komplementiert (im Hilfsregister angezeigt und in den Akkumulator eingeschrieben). Wiederholen Sie die Komplementierung mit anderen Dualzahlen!

Logische Befehle

Bei den arithmetischen Befehlen wurden Dualzahlen verarbeitet. Die logischen Befehle für die UND- bzw. ODER-Verknüpfung beziehen sich auf Bitstellen. Es werden Bit 1 und Bit 1 oder Bit 3 und Bit 3 der angesprochenen Operanden für sich verknüpft.
Stecken Sie die Befehle und führen Sie die Bedienungen wie angegeben durch:

Programm

IN	4	0
IN	3	1
ANA	4	2

1) BEFEHL, RÜCKSETZEN Kippschalter: z.B. 0011, START;
2) Kippschalter: z.B. 0101, START;
3) SCHRITT, START, START, Befehlsbereitstellung; START, START

Beobachten Sie die Befehlsausführung für "ANA 4" und beobachten Sie dabei Hilfsregister und Akkumulator sowie den Inhalt der Adresse 4.
Der Befehl „ANA 4“ bewirkt, dass der Inhalt der Adresse 4 mit dem Akkumulatorinhalt bitweise nach der logischen Funktion ANA verknüpft wird. Anzeige im Hilfsregister, Einschreibung in Akkumulator.
Wiederholen Sie diesen Befehl mit anderen Kippschaltereinstellungen!

Ersetzen Sie nun die Operation ANA im Programmspeicher durch ORA und führen Sie die gleichen Bedienungen durch!

Verschiebebefehle

Die Verschiebebefehle (vergleiche Beispielgruppe "Schieberegister") beziehen sich auf Akkumulatorinhalte und sind Grundvoraussetzung für Multiplikation und Division.
Stecken Sie folgende Befehle und bedienen Sie wie angegeben:

Programm

IN	7	0
RRC		1

1) BEFEHL, RÜCKSETZEN, Kippschalter beliebig einstellen, START,
2) SCHRITT, START, START, Befehlsbereitstellung; START, START, Befehlsausführung, beobachten Sie den Akkumulatorinhalt.

Der Befehl RRC verschiebt den Inhalt des Akkumulators um 1 Bit nach rechts; von links werden Nullen nachgezogen.
Wiederholen Sie die Bedienungen mit anderen Zahlen (Kippschalter) und beachten Sie, dass die Rechtsverschiebung einer Division durch 2 entspricht (Rundungsfehler!)
Ersetzen Sie den Befehl RRC durch RLC und wiederholen Sie alle Bedienungen für den Befehlsablauf "Linksverschiebung".

Sprungbefehle

Ein Programm wird normalerweise Befehl für Befehl in ansteigender Reihenfolge linear durchlaufen. Dieser Programmablauf kann durch die Sprungbefehle verändert werden. Neben dem unbedingten Sprungbefehl "JMP a" gibt es im Modell die bedingten Sprungbefehle: "JM a", "JZ a" und "JC a". ("a" ist die Fortsetzadresse des Programms, vergleiche Befehlsliste)
Stecken Sie die folgenden Befehle und bedienen Sie wie angegeben:

Programm

JMP	12	0
….	….	1
JMP	9	2
….	….	9
….	….	10
JMP	1	12

Beachte an den Lampen, wie die Sprünge erfolgen.
Beachte ebenso die Befehlszähler und Befehlsregister.

Stecke nun folgendes Programm

IN	3	0
JZ	12	1

1) Befehl, Rücksetzen, Kippschalter auf 0010, START, notiere den Akkumulatorinhalt, START

Der Sprung wurde nicht ausgeführt, weil die Sprungbedingung „Springe falls der Akkumulator gleich Null ist“ nicht erfüllt war.

Stecke nun folgendes Programm

IN	0	0
ADD	0	1
JC	12	2

1) BEFEHL, RÜCKSETZEN; Kippschalter auf 0011, START, START
Notiere den Inhalt des Überlaufregisters links vom Akkumulator

2) START
Der Sprung wurde nicht ausgeführt, weil die Sprungbedingung „Springe falls Überlauf vorhanden“ nicht erfüllt war.

Wiederhole die Bedingungen mit den Kippschaltereinstellungen 0111. Was passiert nun mit dem Überlauf nach Ausführung des Sprungs?

Eingabetastatur und Dezimalanzeige

Mit dieser Tastatur können die bei jedem Programmablauf nötigen Daten dezimal eingegeben werden, was die Vorführung bei größeren Programmen wesentlich vereinfacht.
Zusätzlich können die ausgegebenen Daten (im Ausgaberegister) dezimal angezeigt werden. Dies ist insbesondere bei negativen Binärzahlen vorteilhaft, die in der dualen (komplimentierten) Form schwer zu übersehen sind.

Bedienungselemente:

-Umschalter BCD / Oktal:
In der Stellung „BCD“ werden die Zahlen 0 bis 9 angezeigt.
In der Stellung „+/- Okt.“ Werden die Zahlen -8,….0,…7 angezeigt, entsprechend der Zahlendarstellung mit Vorzeichen (1. Bit links ist das Vorzeichenbit)

Adresse/Inhalt:
Die Lämpchen zeigen an, ob eine Adresse oder ein Inhalt eingegeben wird. Mit Hilfe der Taster kann man die Vorwahl bestimmen (wechselt ansonsten automatisch nach jeder Eingabe)

Bedienung:

Der Rechner wird auf die Betriebsart für den schnellen Programmablauf PROGR. 40kHz eingestellt. Die Kippschalter müssen auf Null stehen, sonst gibt es eine Verfälschung der Eingabe.
Folgendes Programm muss gesteckt werden:

IN	7	0
IN	(7)	1
OUT	(7)	2
HLT		3

Wenn die Lampe „Adresse“ leuchtet, wird beim Drücken einer Taste (0….7) zunächst nur die Adresse für die darauf folgende eigentliche Dateneingabe im Rechner hinterlegt. Nach der Adresseneingabe leuchtet die Lampe „Inhalt“. Beim Drücken der Taste (-8…..0…..7), die dem gewünschten Inhalt entspricht, wird das Äquivalent der Taste in die vorgewählte Adresse des Arbeitsspeichers eingegeben.

Beim normalen Ablauf werden demnach fortlaufend Adresse 1, erster Inhalt, Adresse 2, zweiter Inhalt usw. eingegeben. Falls man sich bei der Eingabe geirrt hat, kann man diese Eingabe (Adresse oder Inhalt) wiederholen, indem man durch Drücken einer der Tasten Adresse oder Inhalt den ursprünglichen Zustand wieder herstellt.

Funktionsweise:

Bei einem Tastendruck der Eingabetastatur wird eine Umcodierung der dezimalen Zahl in ihre duale Form vorgenommen und auf das Eingaberegister geschaltet. Die Eingabe in den Arbeitsspeicher kann nur vom Rechner selbst durch entsprechende Befehle vollzogen werden. Dazu wird bei jedem Tastendruck eine Anforderung ausgelöst, die einen Programmstart bewirkt.

Adressen werden über die Anforderung 0 (ANF 0) eingegeben, die einen Programmstart ab Befehlsadresse 0 auslöst. Die gewählte Adresse wird bei dem obigen Programm in eine Hilfszelle (Adresse 7) und in die gewünschte Adresse selbst eingetragen. Der abschliessende Ausgabebefehl soll lediglich eine Anzeige auf der Dezimalanzeige veranlassen.

Inhalte werden über die Anforderung 1 (ANF 1) eingegeben, die einen Programmstart ab Befehlsadresse 1 auslöst. Der Befehl IN (7) veranlasst eine

Eingabe in die vorgewählte und in Adresse 7 hinterlegte Adresse durch indirekten (Substitution) Speicherzugriff.

Da keine Speicherung der Zahlen, die eingegeben werden, vorgesehen wurde (um trotzdem über die Kippschalter und Anforderungstasten eingeben zu können), muss bei langsameren Betriebsarten als PROG. 40 kHz die gewählte Taste so lange gedrückt werden, wie das Programm läuft da sonst die Null eingeschrieben wird.

Die Dezimalanzeige setzt die während des Programmablaufs ausgegebenen Daten in die dezimale Form um, solange sie im Ausgaberegister stehen. Entsprechend der Zahlendarstellung im Demo-Rechner (3 Bit plus 1 Vorzeichenbit) werden die 16 dualen Kombinationen in die dezimalen Zahlen -8….0….7 umgesetzt und mit zwei 7-Segment-Anzeigen für Betrag und Vorzeichen angezeigt. Außerdem kann die Anzeige auf binär codierte Dezimalzahlen (BCD) umgeschaltet werden (0….9).

(Fortsetzung: Demo-Programme……)