

Autor: Ortmann, Jürgen.

Titel: Wie arbeiten Computer?

Quelle: Einführung in die PC-Grundlagen. München, 8. Auflage, 2003. S. 29-41.

Verlag: Addison-Wesley Verlag.

Die Veröffentlichung erfolgt mit freundlicher Genehmigung des Verlags.

Jürgen Ortmann

Wie arbeiten Computer?

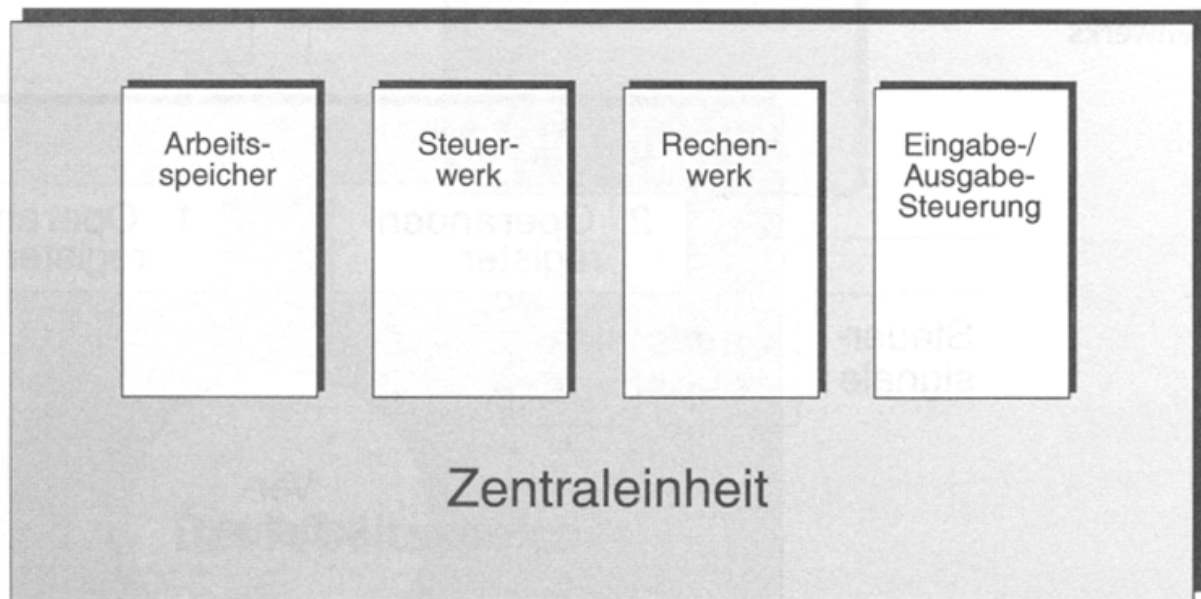
In diesem Kapitel wollen wir uns Computer einmal aus einem recht abstrakten Blickwinkel anschauen. Dieses Kapitel soll Ihnen ein wenig verdeutlichen, wie Informationen im Computer von den verschiedenen Teilsystemen bearbeitet werden.

1. Die Zentraleinheit

Computer bestehen im Wesentlichen aus vier Teilsystemen, die zusammen die Zentraleinheit ergeben

- Steuerwerk
- Rechenwerk
- Arbeitsspeicher
- Ein- und Ausgabesteuerung

Bild 2.1: Struktur der Zentraleinheit



Nur durch das geordnete Zusammenspiel dieser vier Grundeinheiten sind die hohen Leistungen moderner Computer möglich.

Rechenwerk, Arbeitsspeicher

Die Verarbeitung der eingehenden Daten geschieht im Rechenwerk. Heute sind Rechengeschwindigkeiten von vielen Millionen Additionen pro Sekunde möglich. Eine derart hohe Rechengeschwindigkeit setzt voraus, dass die Daten dem Rechenwerk entsprechend schnell bereitgestellt werden können. Deshalb benutzt man einen Arbeitsspeicher, in dem die zu verarbeitenden Daten abgelegt werden. Das Rechenwerk kann jederzeit auf diese Daten zugreifen, sie verarbeiten und die Ergebnisse wieder im Arbeitsspeicher ablegen.

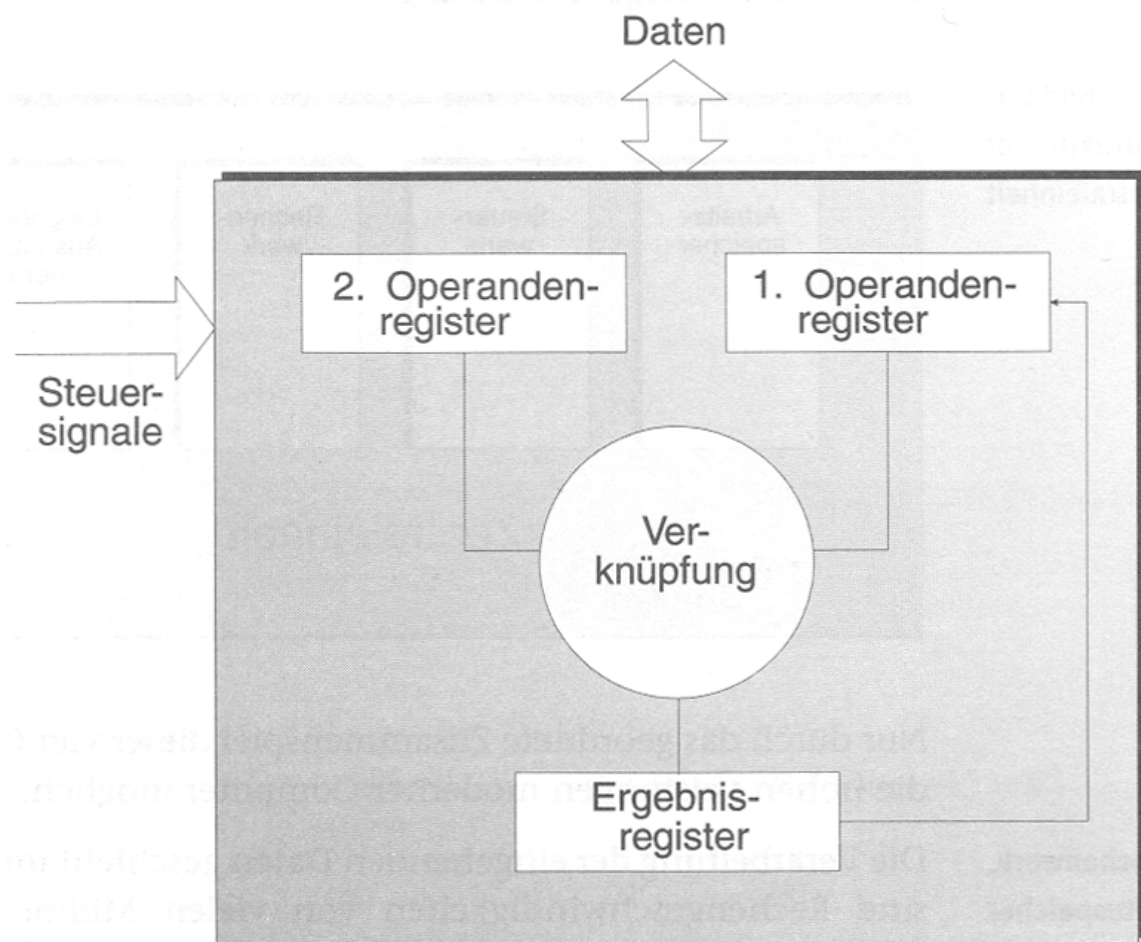
Moderne Computer arbeiten mit Programmen, die eine Vielzahl von Befehlen enthalten. Das Steuerwerk hat die Aufgabe, die Befehle zu verstehen und die Ausführung zu steuern. Es »managt« die Zusammenarbeit von Rechenwerk und Arbeitsspeicher. Weiterhin arbeitet es mit der Ein- und Ausgabesteuerung zusammen und koordiniert damit den Datenaustausch mit der Umwelt über Ein- und Ausgabegeräte.

2. Der Mikroprozessor

Ein *Mikroprozessor* besteht im Wesentlichen aus einem kleinen Siliziumplättchen, dem *Chip*, auf dem aus Millionen von kleinsten Bauteilen das Rechenwerk und das Steuerwerk eines Computers aufgebaut sind.

2.1 Das Rechenwerk

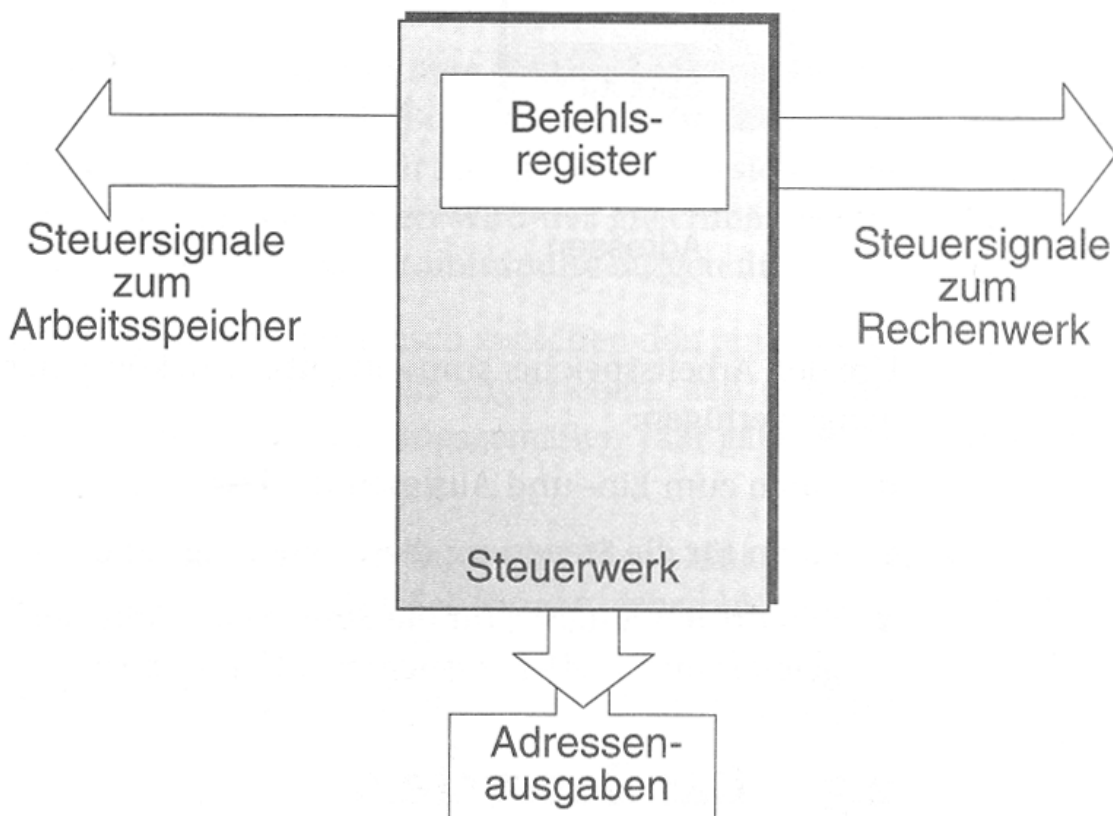
Im *Rechenwerk* werden die zu verrechnenden Daten verarbeitet. Diese Daten nennt man *Operanden*. Sie werden in kleinen, besonders schnellen Speichern innerhalb des Rechenwerks, den *Registern*, abgelegt.



Alle im Rechenwerk durchgeführten Grundrechenarten basieren auf der Addition von Zahlen. Die Subtraktion besteht aus der Addition von negativen Werten, Multiplikation und Division werden durch entsprechende Unterprogramme gelöst.

2.2 Das Steuerwerk

Das *Steuerwerk* hat die Aufgabe, das Zusammenspiel von Rechenwerk und Arbeitsspeicher zu koordinieren. Einzelne Programmbefehle werden aus dem Speicher zum Steuerwerk übertragen. Auch das Steuerwerk besitzt einen kleinen, besonders schnellen Speicher, das *Befehlsregister*. In dieses Register werden die einzelnen Programmbefehle nacheinander eingelesen und festgehalten. Das Steuerwerk gibt entsprechend dem eingelesenen Befehl Steuersignale an Rechenwerk und Arbeitsspeicher, um Daten aus dem Speicher in das Rechenwerk zu übertragen. Die Daten werden im Rechenwerk verarbeitet und das Ergebnis wieder im Arbeitsspeicher abgelegt.



2.3 Der Arbeitsspeicher

Funktion des Arbeitsspeichers

Der *Arbeitsspeicher* dient zum Ablegen der Daten, die im Rechenwerk verarbeitet werden sollen. Weiterhin werden auch die einzelnen Programmbefehle in diesem Speicher festgehalten.

Der Arbeitsspeicher lässt sich vergleichen mit einem großen Regal mit vielen Fächern. Jedes dieser Fächer besitzt eine Nummer, wodurch es eindeutig von den anderen Fächern zu unterscheiden ist. In jedem dieser Fächer lassen sich Informationen ablegen.

Speicherzellen

Im Arbeitsspeicher eines Computers werden die Fächer als Speicherzellen bezeichnet. Die Nummerierung der Fächer entspricht den so genannten *Adressen* der einzelnen Speicherzellen.

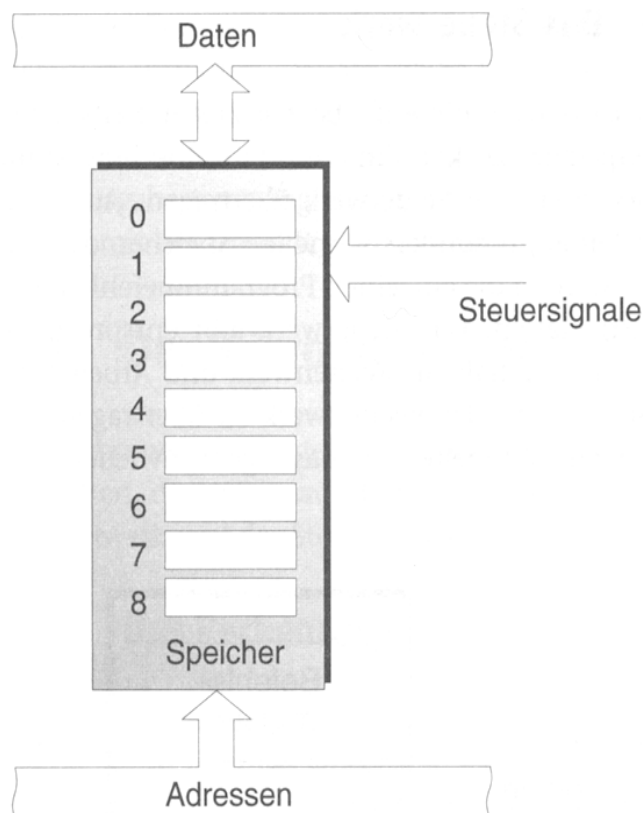


Bild 2.4: Struktur des Arbeitsspeichers

Um den Arbeitsspeicher sinnvoll nutzen zu können, muss er über drei Eingänge verfügen:

- einen zum Ein- und Auslesen der Daten,
- einen für die Steuerung dieses Vorgangs und
- den dritten Eingang für die Eingabe der Adressen, aus denen die Daten gelesen und wieder abgelegt werden sollen.

3 Das Bussystem

Bei der Beschreibung von Rechenwerk, Steuerwerk und Arbeitsspeicher konnten Sie sehen, dass verschiedene Informationen zwischen diesen Teilsystemen transportiert werden. Der Transport erfolgt über Leitungen, die alle Systeme der Zentraleinheit miteinander verbinden. Diese Leitungen werden Bus oder *Bussystem* genannt.

Bei einem Personalcomputer wird zwischen folgenden drei Bussystemen unterschieden:

<i>Bussystem</i>	<i>Funktion</i>
Adressbus	Übertragung der Adressen an den Arbeitsspeicher
Steuerbus	Übertragung der Steuersignale zwischen Steuerwerk und den anderen Teilsystemen
Datenbus	Transport der Daten zwischen Rechenwerk und der Ein- und Ausgabesteuerung

4. Die Ein- und Ausgabesteuerung

Die oben beschriebenen Teilsysteme sind sehr wichtig für die Verarbeitung von Daten. Die zu verarbeitenden Daten müssen jedoch zuvor in den Rechner eingegeben werden und die Ergebnisse sollen nach der Verarbeitung dem Benutzer zugänglich sein. Die Ein- und Ausgabe von Daten wird durch die *Ein- und Ausgabesteuerung* geregelt. Sie ermöglicht die Verbindung zwischen dem Rechner und dem Benutzer. Als mögliche Kontaktstellen wären hier die Tastatur als Eingabe- und der Monitor als Ausgabeeinheit zu nennen.

5. Taktgeber

Taktgeber

Innerhalb eines Computers werden Informationen in Form von elektrischem Strom über die Busleitungen von Grundeinheit zu Grundeinheit transportiert. Bei diesem Strom handelt es sich nicht um einen gleichförmigen Strom, stattdessen wird der gleichmäßige Stromfluss durch einen *Taktgeber* in einzelne Stromimpulse aufgeteilt.

Der Informationsaustausch zwischen den einzelnen Einheiten wird durch diesen periodischen Takt angetrieben, den Grundtakt des Mikroprozessors. Ohne einen ordnungsgemäßen Takt gäbe es keine ordnungsgemäße Informationsverarbeitung.

Grundtakt

Der *Grundtakt* des Prozessors wird durch einen Quarz bestimmt, der durch elektrischen Strom ins Schwingen gebracht wird (ähnlich wie bei Armbanduhren). Moderne Mikroprozessoren arbeiten mit Taktfrequenzen von mehreren hundert Millionen Hertz (Schwingungen pro Sekunde).

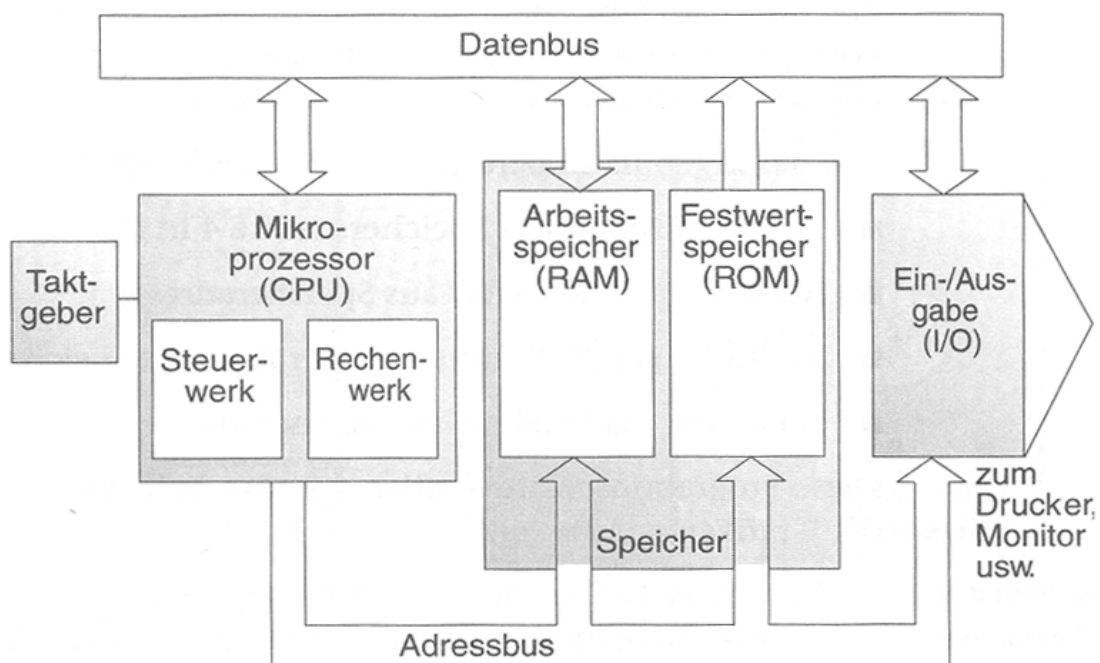


Bild 2.5: Komponenten der Zentraleinheit

6. Das Zusammenspiel von Prozessor, Speicher und Bussystem

Wie arbeiten nun die einzelnen Grundbausteine wie Steuerwerk, Rechenwerk und Arbeitsspeicher zusammen?

Die ersten programmgesteuerten Rechner wie z. B. die ZUSE Z3 wurden durch ein Programm gesteuert, das von außen über Lochstreifen eingegeben wurde. Über diesen Lochstreifen wurde dem Rechner jeder einzelne Arbeitsschritt in Form von Programmbefehlen mitgeteilt. Die Interpretation der einzelnen Befehle erfolgte im Steuerwerk. Für die Ausführung wurden die Daten über den Datenbus ins Rechenwerk transportiert und verarbeitet. Die Ergebnisse wurden im Arbeitsspeicher abgelegt.

Programmbefehle

Jeder einzelne Programmbefehl muss angeben, wo die zu verarbeitenden Daten im Arbeitsspeicher zu finden sind, und was mit ihnen geschehen soll. Zu diesem Zweck sind die Befehle aufgeteilt in einen *Operationsteil* und einen *Adressteil*:

Operation (Adresse)

Der Operationsteil gibt an, was zu tun ist; der Adressteil enthält die Adresse der Speicherzelle, mit deren Inhalt die Operation durchgeführt werden soll.

Beispiel

Um die bisher nur grob beschriebene Vorgehensweise besser kennen zu lernen, wollen wir uns die Zusammenarbeit der einzelnen Bausteine am Beispiel der Rechenaufgabe

$$(5 + 3) \times 4$$

näher ansehen. Zunächst sollen die Zahlen 5 und 3 addiert werden. Das Ergebnis wird anschließend mit 4 multipliziert. Die Zahlen 5, 3 und 4 sind vorgegeben und können im Arbeitsspeicher auf den Adressen 4, 5 und 6 abgelegt werden.

Um das Ergebnis zu berechnen, sind vier Programmschritte abzuarbeiten:

- Hole den Inhalt aus Speicheradresse 4 in das Rechenwerk.
- Addiere dazu den Wert aus Speicheradresse 5.
- Multipliziere die Summe mit der Zahl aus Speicheradresse 6.
- Speichere das Ergebnis in Speicheradresse 7 ab.

Diese Programmschritte wollen wir uns nun etwas ausführlicher anhand der Grafiken anschauen:

Einlesen des 1. Operanden

1. Um die erste Zahl aus dem Arbeitsspeicher auszulesen, muss das Steuerwerk ein Steuersignal für »Auslesen« an den Arbeitsspeicher senden. Gleichzeitig muss es über den Adressbus angeben, dass die Zahl aus der Speicherzelle 4 ausgegeben werden soll. Die Zahl wird dann über den Datenbus in das Operandenregister 1 im Rechenwerk transportiert und dort gespeichert (Abb. 2.6).

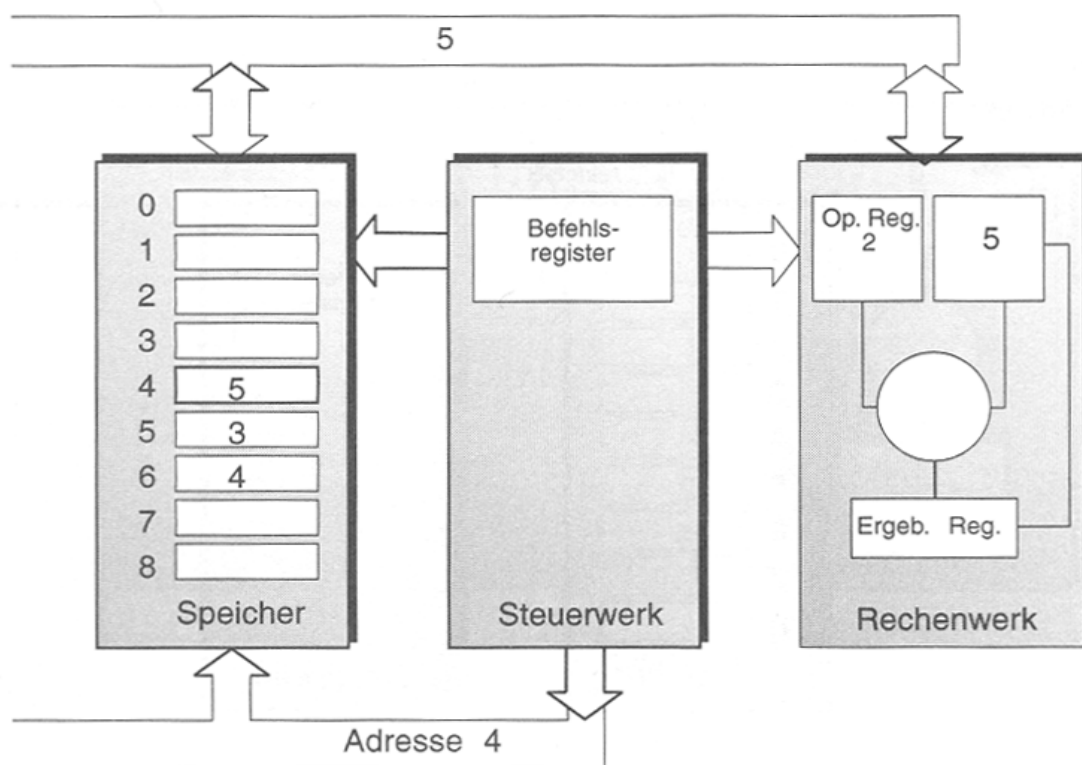


Bild 2.6: Einlesen des Operanden

Einlesen des 2. Operanden

2. Die Zahl aus Speicherzelle 5 soll in das Operandenregister 2 transportiert werden (wie Schritt 1).

Addition

3. Das Steuerwerk gibt nun den Befehl an das Rechenwerk, die beiden Operanden zu addieren. Das Ergebnis wird zuerst im Ergebnisregister 1 abgelegt (Abb. 2.7).

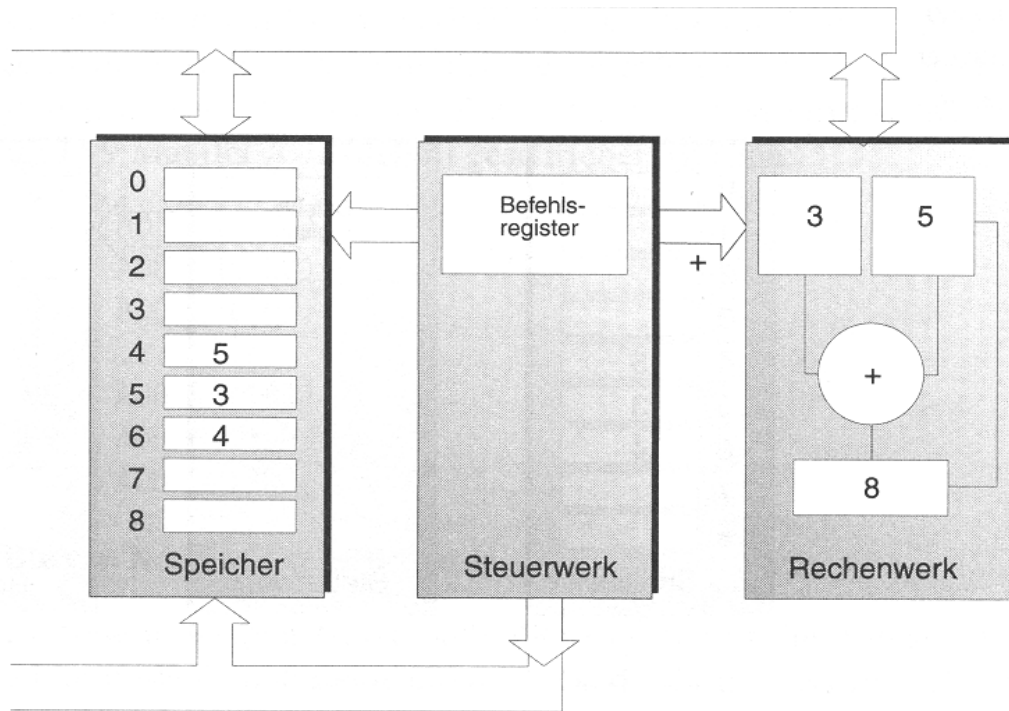


Bild 2.7: Einlesen des 2. Operanden und Addition

Einlesen des 3. Operanden

4. Die Werte in den Operandenregistern werden nun nicht mehr benötigt und können gelöscht werden. Dafür wird der Wert 8 aus dem Ergebnisregister in das Operandenregister 1 übertragen, während der Inhalt der Speicherzelle 6 in das zweite Operandenregister gelesen wird (Abb. 2.8).

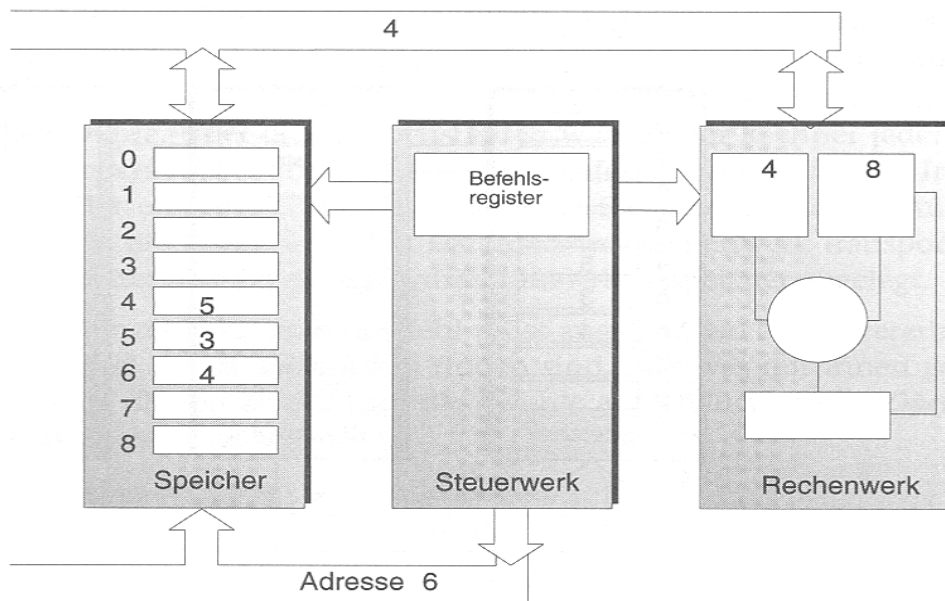


Bild 2.8: Einlesen des 3. Operanden

Multiplikation

5. Das Steuerwerk gibt das Steuersignal, dass mit den beiden Werten eine Multiplikation durchgeführt werden soll. Der errechnete Wert wird zunächst in das Ergebnisregister gestellt und von dort in das Operandenregister 1 übertragen (Abb. 2.9).

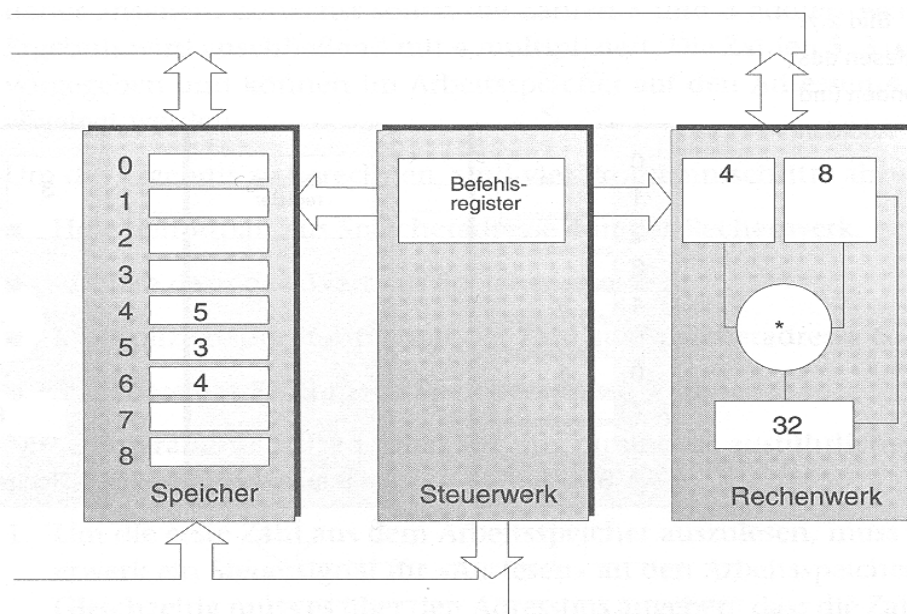


Bild 2.9: Durchführung der Multiplikation

Ergebnis speichern

6. Zum Abschluss der Rechenoperation wird nun das Ergebnis in den Arbeitsspeicher transportiert und in Zelle 7 abgespeichert (Abb. 2.10).

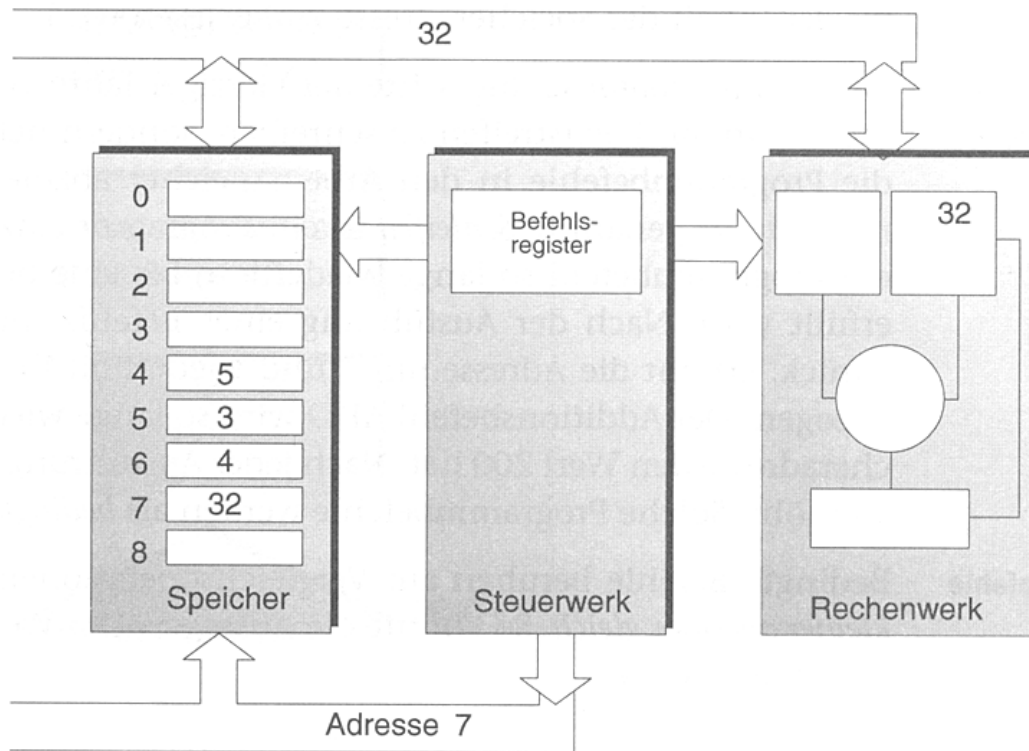


Bild 2.10: Speichern des Ergebnisses

Die bis hier beschriebenen Schritte könnten in die folgenden vier Programmbefehle mit der Syntax Operation (Adresse) übersetzt werden:

1. *HOLE* (4)
2. *ADD* (5)
3. *MUL* (6)
4. *SPEI* (7)

Diese Art der Programmierung funktioniert zwar, wird aber spätestens bei einer großen Zahl sich wiederholender Operationen zu umständlich. Sollen zum Beispiel 100 Zahlen addiert werden, die in den Speicherzellen mit den Adressen 101 bis 200 abgelegt sind, muss einmal der HOLE-Befehl und 99 Mal der ADD-Befehl geschrieben werden:

1. HOLE (101)
2. ADD (102)
3. ADD (103)
- ...
98. ADD (198)
99. ADD (199)
100. ADD (200)

Die von Neumann'sche Struktur

In diesem Beispiel ändert sich aber immer nur die Adresse, an der die zu addierende Zahl abgespeichert ist. Da die Zahlen in hintereinander liegenden Speicherzellen abgelegt sind, erhöht sich lediglich die Adresse um 1, bis sie den Wert 200 erreicht, wo der letzte zu addierende Wert gespeichert ist. Einfacher wäre es, wenn der Additionsbefehl nur einmal geschrieben werden brauchte. Nur die Adresse müsste so lange um 1 erhöht werden, bis der Inhalt der Speicheradresse 200 gelesen wird.

John von Neumann Schleifen

John von Neumann schlug Mitte der vierziger Jahre vor, Programme nicht mehr starr auf Lochstreifen zu schreiben, sondern neben den Daten auch die Programmbefehle in den Arbeitsspeicher abzulegen. Dadurch ist es möglich, so genannte *Schleifen* in ein Programm einzubauen. Dabei wird ein Programmbefehl so lange wiederholt, bis eine bestimmte Bedingung erfüllt wird. Nach der Ausführung eines Befehls springt das Programm zurück, erhöht die Adresse um 1 und wiederholt ihn. Auf unser Beispiel bezogen: Der Additionsbefehl ADD wird so lange wiederholt, bis die Speicheradresse den Wert 200 hat. Nach jeder Ausführung wird die Adresse um 1 erhöht. Solche Programmbefehle werden als *bedingte Befehle* bezeichnet.

Bedingte Befehle

beruhen auf Vergleichsoperationen wie z. B. *größer als*, *kleiner als* oder *gleich*. So könnte die Aussage: »Das Wetter ist schön« in die logische Aussage

Wetter = schön

übersetzt werden. Schaut man nach draußen, lässt sich bestimmen, ob die Aussage wahr oder falsch ist. Ein an diese Aussage anschließender Befehl kann dadurch an die aktuelle Wetterlage angepasst werden. Bei schönem Wetter gehen wir ins Freibad, bei schlechtem Wetter muss der Abwasch erledigt werden.

Das folgende Ablaufdiagramm (Bild 2.11) stellt die Addition der 200 Zahlen grafisch dar. Die Rechtecke enthalten einfache Anweisungen, die Raute stellt einen bedingten Befehl dar. Ist die in der Raute beschriebene Bedingung nicht erfüllt, springt das Programm an die Stelle im Diagramm zurück, die durch den Pfeil gekennzeichnet ist.

Werden dem Steuerwerk die Programmbefehle von außen durch Lochstreifen vorgegeben, muss sich das Steuerwerk nicht um die Reihenfolge der Befehle kümmern. Dadurch, dass die Befehle nun im Arbeitsspeicher abgelegt sind, hat das Steuerwerk die zusätzliche Aufgabe, den einzelnen Befehlen Adressen zuzuweisen, sie einzulesen und zu verarbeiten. Beim Verarbeiten muss das Steuerwerk bedingte Befehle berücksichtigen, um Programmsprünge durchführen zu können.

Programmzähler

Für diesen Zweck wird das Steuerwerk durch einen *Programmzähler* erweitert, der nach dem Einlesen eines Befehls in das Befehlsregister um 1 erhöht wird, damit der Befehl mit der nächsten Adresse eingelesen werden kann. Die Adressen für die zu verarbeitenden Befehle gibt also der Programmzähler an. Die Adressen für die zu verarbeitenden Daten sind im Adressteil der einzelnen Befehle enthalten und werden im Befehlsregister gelesen.

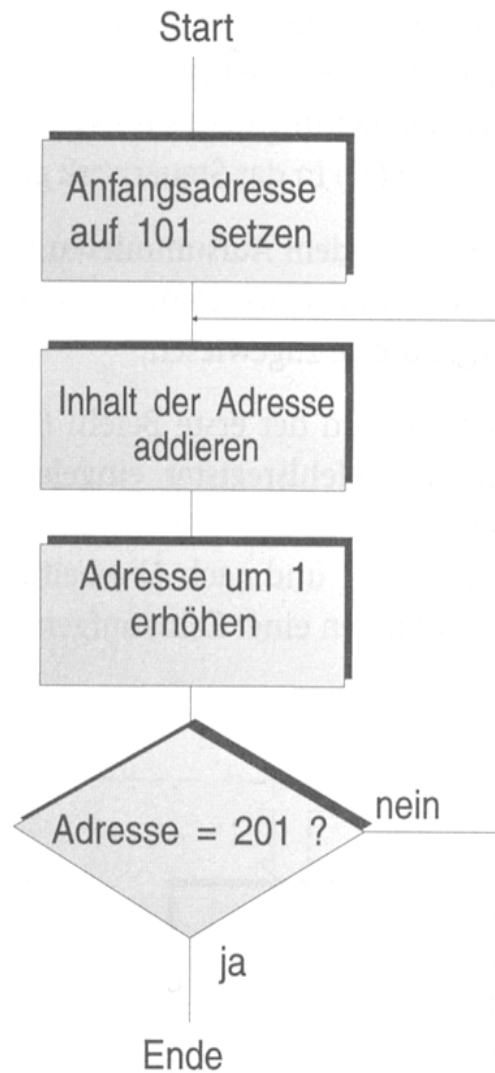


Bild 2.11: Aufbaudiagramm

Beim Laden eines Programms in den Arbeitsspeicher wird die Anfangsadresse im Arbeitsspeicher von einem übergeordneten Programm, dem Betriebssystem (siehe Kapitel 8), festgelegt. Für unser erstes Rechenbeispiel wird die Anfangsadresse auf 0 gesetzt.

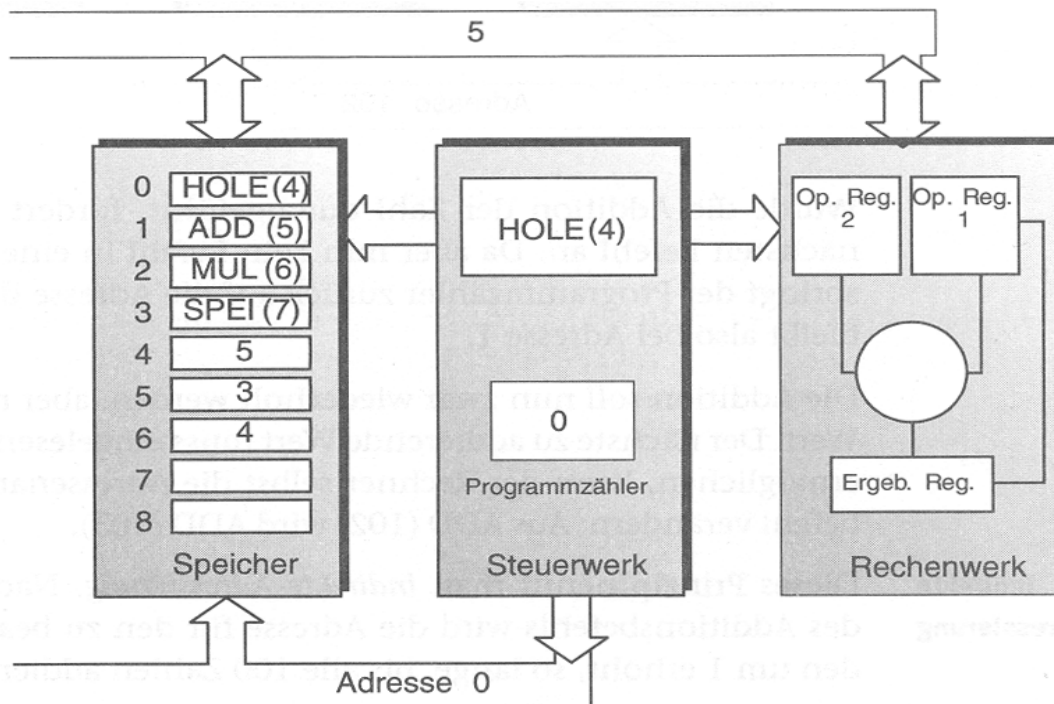


Bild 2.12: Bearbeiten der Aufgabe $(5 + 3) * 4$

Bei Aufruf des Programms wird der Inhalt dieser Speicherzelle, also der Befehl HOLE (4), in das Steuerwerk eingelesen und der Programmzähler um 1 erhöht. Nach der Abarbeitung dieses ersten Befehls wird der Inhalt der Speicheradresse 1 (ADD (5)) in das Steuerwerk geholt usw.

Bei unserem zweiten Beispiel, dem Aufsummieren von 100 Zahlen, sieht der Ablauf dagegen etwas anders aus. Dem Programm wird wieder vom Betriebssystem eine Startadresse zugewiesen.

Beim Start des Programms wird der erste Befehl (HOLE den Inhalt von Speicherzelle 101) in das Befehlsregister eingelesen. Aufgrund dieser Anweisung wird die erste Zahl in das Rechenwerk transportiert. Der Programmzähler erhöht sich um 1 und nach Abarbeitung des ersten Befehls wird der zweite Befehl, Addieren einer Zahl, aufgerufen.

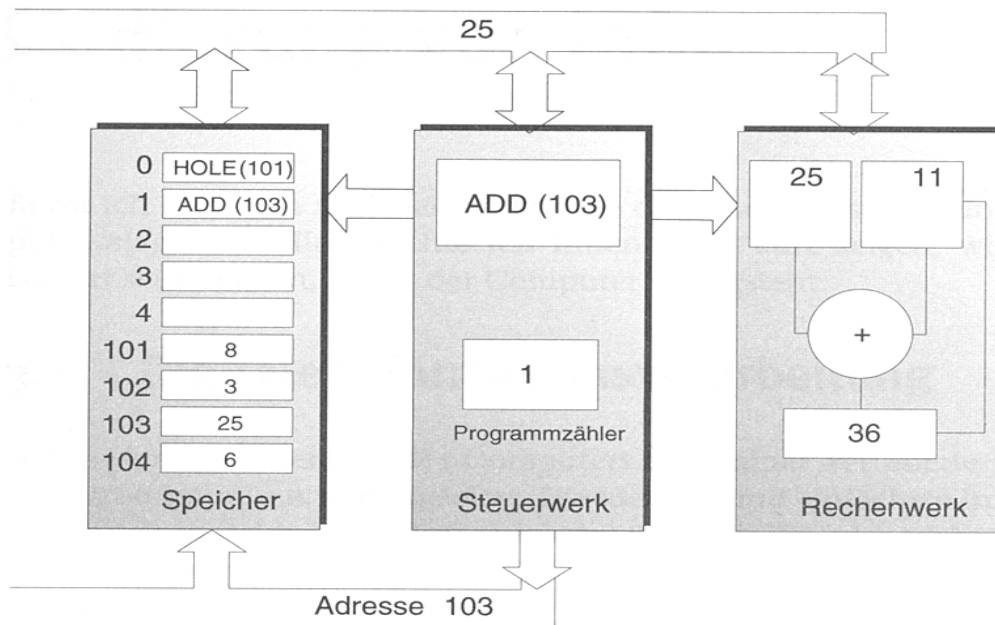


Bild 2.13: Aufsummieren von 100 Zahlen (Schritt 1)

Wurde die Addition der Zahl durchgeführt, fordert das Steuerwerk den nächsten Befehl an. Da aber nun kein Befehl in einer Folgeadresse steht, springt der Programmzähler zurück auf die Adresse des Additionsbefehls, bleibt also bei Adresse 1.

Die Addition soll nun zwar wiederholt werden, aber nicht mit demselben Wert. Der nächste zu addierende Wert muss eingelesen werden. Um dies zu ermöglichen, kann der Rechner selbst die Adressenangabe im Additionsbefehl verändern: Aus ADD (102) wird ADD (103).

Indirekte Adressierung

Dieses Prinzip nennt man *indirekte Adressierung*. Nach jeder Ausführung des Additionsbefehls wird die Adresse für den zu bearbeitenden Operanden um 1 erhöht, so lange, bis alle 100 Zahlen addiert wurden.

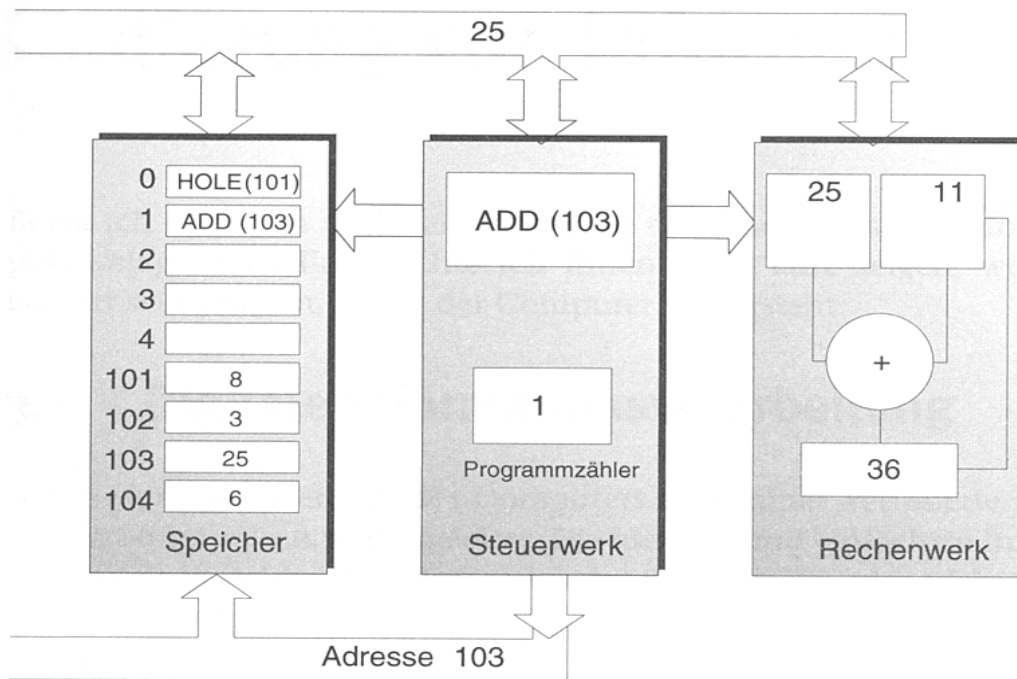


Bild 2.14: Aufsummieren von 100 Zahlen (Schritt 2)

Zusammenfassung

Zusammenfassend lässt sich die Verarbeitung eines Programmbefehls folgendermaßen beschreiben:

- Die Adresse des Befehls muss vom Steuerwerk entsprechend dem Stand des Programmzählers ausgegeben werden.
- Der Befehl wird aus dem Arbeitsspeicher in das Steuerwerk geladen.
- Der Befehl im Steuerwerk enthält die Adresse der zu verarbeitenden Zahl, die an den Arbeitsspeicher weitergegeben wird.
- Der Operand wird aus dem Speicher ausgelesen und in das Rechenwerk geladen.
- Das Ergebnis wird nach der Verarbeitung wieder ausgegeben und im Arbeitsspeicher abgelegt.

Ein Glück, dass ein Computer so schnell ist und diese verwirrenden Einzelschritte ausführen kann, ohne dass wir etwas davon merken, außer dass sich auf dem Bildschirm etwas verändert.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Rechteinhabers unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Speicherung und Verarbeitung in elektronischen Systemen.