

Es wird keine Gewähr übernommen, daß die
in dieser Schrift angegebenen Schaltungen, Geräte, Maschinen,
Anlagen, Bauelemente, Baugruppen oder Verfahren
frei von Schutzrechten sind.

Ratschläge in den Valvo-Briefen
sind unverbindliche und keine Haftung begründende Empfehlungen.

Nachdruck, auch auszugsweise, ist nicht gestattet.

VALVO, Burchardstraße 19, 2000 Hamburg 1



13. April 1977

Valvo Brief

Wie arbeitet der Mikroprozessor 2650

Die Ausführung eines Befehls, gezeigt an einem einfachen Beispiel

Einleitung

Beim praktischen Einsatz von Mikroprozessoren hat man es im allgemeinen mit Systemen zu tun, die im Prinzip nach Bild 1 aufgebaut sind. Ein solches System, häufig als Mikrocomputer bezeichnet, besteht im wesentlichen aus dem Mikroprozessor, dem Programm- und Datenspeicher sowie den Ein-/Ausgabeeinheiten. Ein Takt bestimmt den zeitlichen Ablauf der Arbeitsschritte. Im Gegensatz zu festverdrahteten Logikschaltungen arbeitet ein Mikrocomputer nach Programmen. Ein Programm besteht aus einer Folge von Befehlen.

Die Befehlsfolgen, die der Mikroprozessor ausführen soll, z. B. um einen Fahrstuhl zu steuern, sind im Programmspeicher aufbewahrt, der meistens als programmierbarer Festwertspeicher

(PROM) oder maskenprogrammierter Festwertspeicher (ROM) ausgeführt ist. Aus diesem Speicher ruft der Prozessor die Befehle nacheinander durch Angeben der jeweiligen Adresse ab und führt sie aus. Die Daten, die der Prozessor nach Vorschrift des Programms zu verarbeiten hat, holt er sich — durch Angeben von Adressen — aus dem Datenspeicher, einem Schreib-/Lesespeicher (RAM), oder aus den ebenfalls über Adressen ansprechbaren Ein-/Ausgabeeinheiten. Die Ergebnisse der Verarbeitung gibt der Prozessor wieder an den Datenspeicher oder die Ein-/Ausgabeeinheiten ab. Befehle und Daten unterscheiden sich lediglich dadurch, daß sie in verschiedenen Speicherbereichen — im Programm- und im Datenspeicher — aufbewahrt sind; im übrigen stellen sie sich dem Mikroprozessor in gleicher Form dar. Es sind binäre Wörter, die aus einer bestimmten Anzahl von Nul-

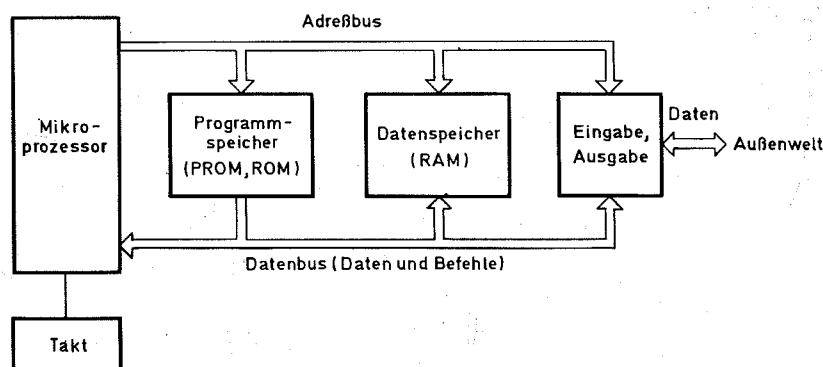


Bild 1. Struktur eines Mikroprozessorsystems (Steuerleitungen sind nicht mit dargestellt)

len und Einsen bestehen. Beim Mikroprozessor 2650 sind diese Befehls- oder Datenwörter 8 bit = 1 Byte lang. Die Wortlängen von Mikroprozessoren liegen im allgemeinen zwischen 4 und 16 bit. Zur Übertragung der Befehls- und Datenwörter zwischen den Funktionseinheiten dient der Datenbus, während die Adresse (maximal 15 bit beim 2650) auf einem Adreßbus an die Speicher und die Ein-/Ausgabeeinheiten gelangt.

Die Ein-/Ausgabeeinheiten stellen die Verbindung zwischen Mikrocomputer und Außenwelt her. Sie machen es eigentlich erst möglich, daß der Mikroprozessor eine Aufgabe wie die Fahrstuhlsteuerung ausführen kann. Über die Ein-/Ausgabeeinheiten erhält der Mikrocomputer Meldungen wie z. B. „rufe den Fahrstuhl“ oder „gewünschtes Stockwerk erreicht“, und er gibt Signale ab wie z. B. „Tür schließen“ oder „Motor halt“.

Als Beispiel für die Arbeitsweise eines Mikroprozessors wird im folgenden die Ausführung eines Befehls durch den Mikroprozessor 2650 beschrieben. Wir nehmen an, daß das Programm bereits geschrieben und im Programmspeicher in Form von Befehlen abgelegt ist.

Start

Zu Beginn muß der Mikroprozessor wissen, wo er mit der Ausführung des Programms anfangen soll. Häufig wird eine Start- oder Rückstell-Taste vorgesehen, nach deren Betätigung der Mikroprozessor auf eine bestimmte Speicher-Adresse zurückgreift und die hier vorgefundene Information als ersten Befehl des Programms verwendet. Daher muß man darauf achten, daß der erste ausführbare Befehl des Programms auch wirklich unter dieser Start-Adresse abgespeichert wird. Beim 2650 lautet die Start-Adresse in der Regel Null.

Beim Start setzt das Rücksetzsignal am Eingang RESET alle Register des Mikroprozessors in ihre Anfangsposition. Danach besteht die weitere Ausführung des Programms in einem Hin- und Herfließen von Information — d. h. sowohl von Adressen als auch von Befehlen und Daten — zwischen dem Mikroprozessor und den Speichern (Bild 1) sowie in einem gelegentlichen Informationsaustausch zwischen Mikroprozessor und Ein-/Ausgabeeinheiten.

Ein Befehl wird geholt

Der Mikroprozessor 2650, dessen Blockschaltung in Bild 2 angegeben ist, arbeitet das Programm

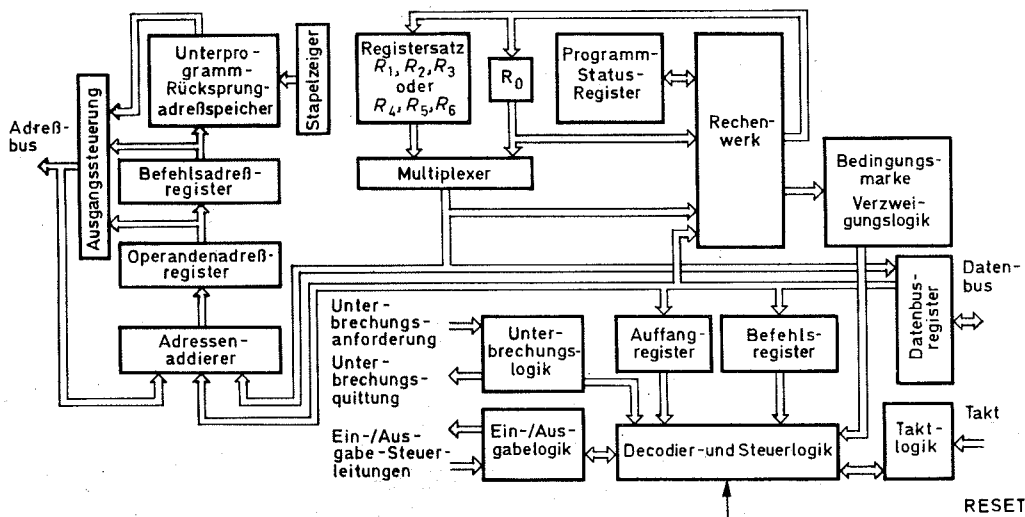


Bild 2. Blockschaltung des Mikroprozessors 2650 (Steuerleitungen sind nicht mit dargestellt)

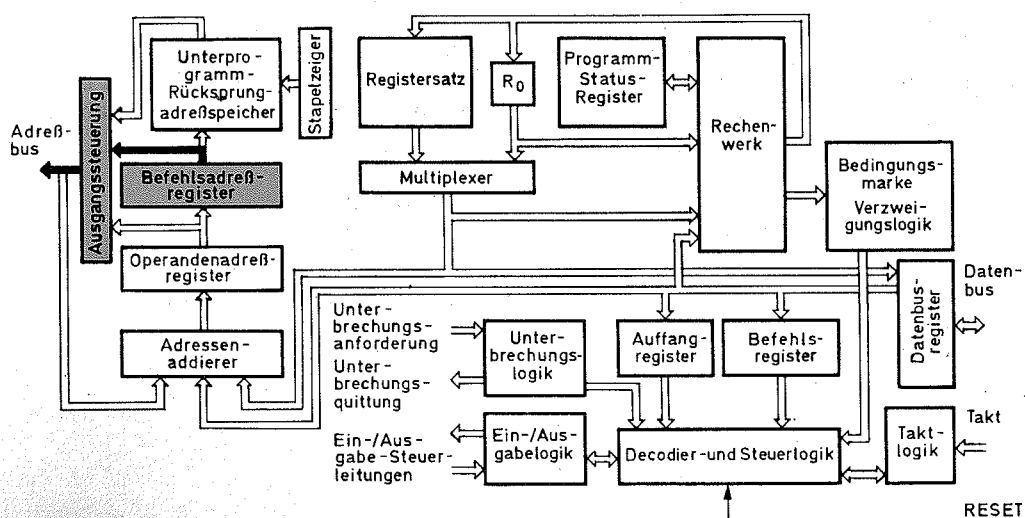


Bild 3. Ausgeben der Befehlsadresse 0

folgendermaßen ab. Das Rücksetzsignal am Eingang RESET setzt alle Bits des Befehlsadßregisters auf 0. Seine Ausgänge werden über die Ausgangssteuerung mit dem Adßbus verbunden (Bild 3), und der Prozessor fordert den Programmspeicher auf, das unter der Adresse Null gespeicherte Befehlsbyte auf den Datenbus zu geben, wo es vom Prozessor empfangen und im Befehlsregister abgelegt wird (Bild 4). Dieses Befehlsbyte sagt dem Prozessor, was er als nächstes zu tun hat.

Da der Befehlsvorrat des 2650 eine Auswahl zwischen 75 verschiedenen Befehlen erlaubt, müssen wir nun zu einem speziellen Beispiel übergehen, um den weiteren Ablauf einer Befehlsausführung zu verfolgen. Wir wählen dafür den Befehl „laden“. Ein derartiger Befehl wird verwendet, wenn man ein Datenwort in den Prozessor übertragen möchte, um es dann z. B. einer speziellen Rechenoperation zu unterziehen oder an eine Ausgabeschaltung weiterzugeben, wofür weitere Befehle erforderlich sind. Im vorliegenden Beispiel beschreiben wir jedoch nur die Ausführung des Befehls „laden“.

Das Befehlsbyte, das der Prozessor nach Bild 5 aus der Adßposition Null des Programmspeichers geholt und in seinem Befehlsregister gespeichert hat, enthält folgende Angaben:

- Art der Operation, die der Prozessor ausführen soll, im vorliegenden Fall „laden“.
- Wie das Datenwort zu finden ist (s. unten).
- Wohin das vom Speicher empfangene Datenwort übertragen werden soll (s. unten).

Aus der Befehlsliste, von der hier nur ein Auszug wiedergegeben ist, geht hervor, daß es beim 2650 vier verschiedene Lade-Befehle gibt, die mnemonisch mit LODZ, LODI, LODR und LODA bezeichnet werden. Die Buchstaben Z, I, R und A kennzeichnen das sogenannte Befehlsformat.

Wir wollen hier nicht auf die unterschiedlichen Bedeutungen dieser Befehle eingehen, sondern uns mit der Feststellung begnügen, daß sich die Formate gemäß Bild 6 durch die Anzahl der Befehlsbytes und die Bedeutung der darin enthaltenen Informationen zur Befehlsausführung und Adressierung unterscheiden. Mit Z, I, R und

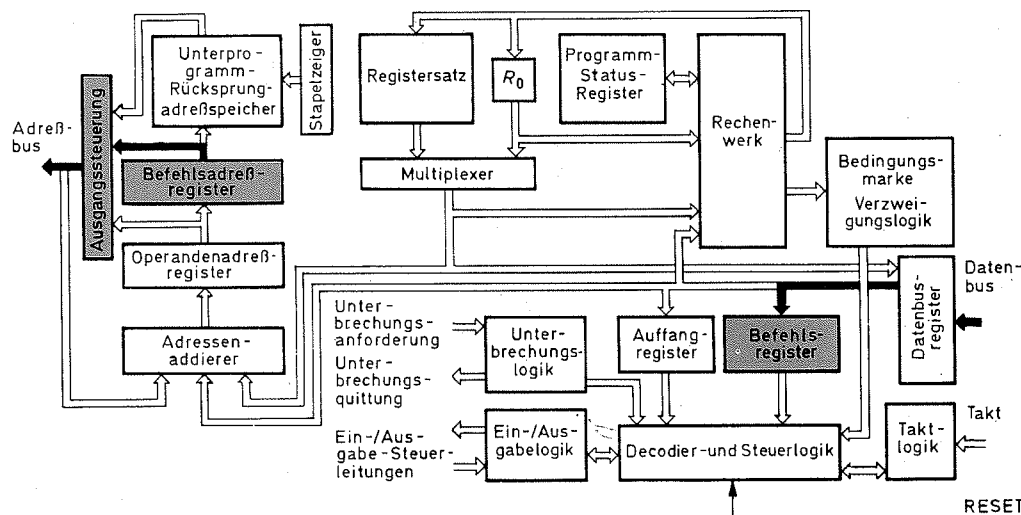


Bild 4. Ausgeben der Befehlsadresse 0 und Holen des ersten Befehlsbytes

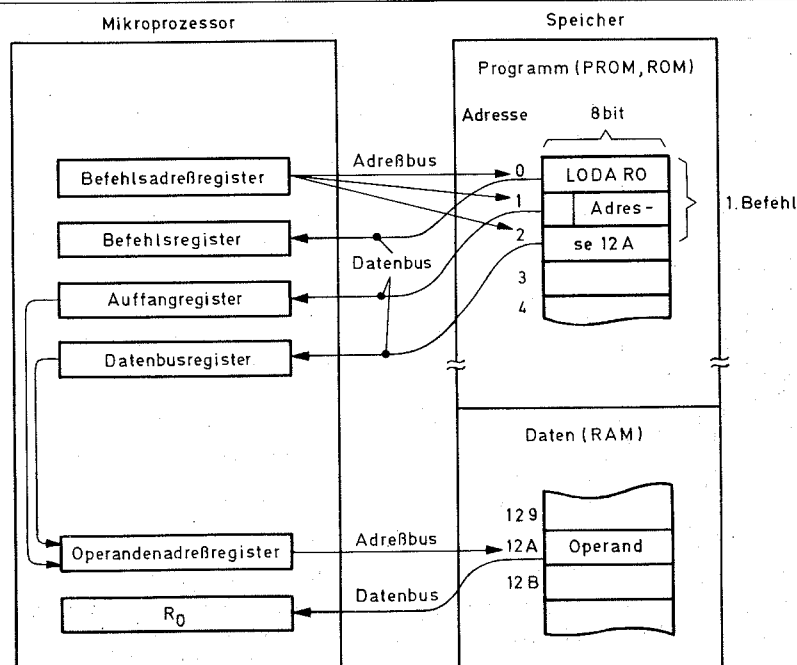


Bild 5. Holen eines 3 Byte-Befehls und Laden eines Datenwortes in das Register R₀

mnemonischer Code		Opera- tionscode	Format ¹⁾	Erläuterung des Befehls	Zyklen	
laden/ speichern	LOD	Z	000 000	1Z	Load Register Zero	2
		I	000 001	2I	Load Immediate	2
		R	000 010	2R	Load Relative	3
		A	000 011	3A	Load Absolute	4
	STR	Z	110 000	1Z	Store Register Zero (r ≠ 0)	2
		R	110 010	2R	Store Relative	3
A		110 011	3A	Store Absolute	4	
arithmetische Operation	ADD	Z	100 000	1Z	Add to Register Zero w/wo Carry	2
		I	100 001	2I	Add Immediate w/wo Carry	2
		R	100 010	2R	Add Relative w/wo Carry	3
		A	100 011	3A	Add Absolute w/wo Carry	4
	SUB	Z	101 000	1Z	Subtract from Register Zero w/wo Borrow	2
		I	101 001	2I	Subtract Immediate w/wo Borrow	2
		R	101 010	2R	Subtract Relative w/wo Borrow	3
		A	101 011	3A	Subtract Absolute w/wo Borrow	4
	DAR		100 101	1Z	Decimal Adjust Register	3
logische Operation	AND	Z	010 000	1Z	AND to Register Zero (r ≠ 0)	2
		I	010 001	2I	AND Immediate	2
		R	010 010	2R	AND Relative	3
		A	010 011	3A	AND Absolute	4
	IOR	Z	011 000	1Z	Inclusive OR to Register Zero	2
		I	011 001	2I	Inclusive OR Immediate	2
		R	011 010	2R	Inclusive OR Relative	3
		A	011 011	3A	Inclusive OR Absolute	4
	EOR	Z	001 000	1Z	Exclusive OR to Register Zero	2
		I	001 001	2I	Exclusive OR Immediate	2
		R	001 010	2R	Exclusive OR Relative	3
		A	001 011	3A	Exclusive OR Absolute	4

Valvo-Brief
13. April 1977
Seite 4

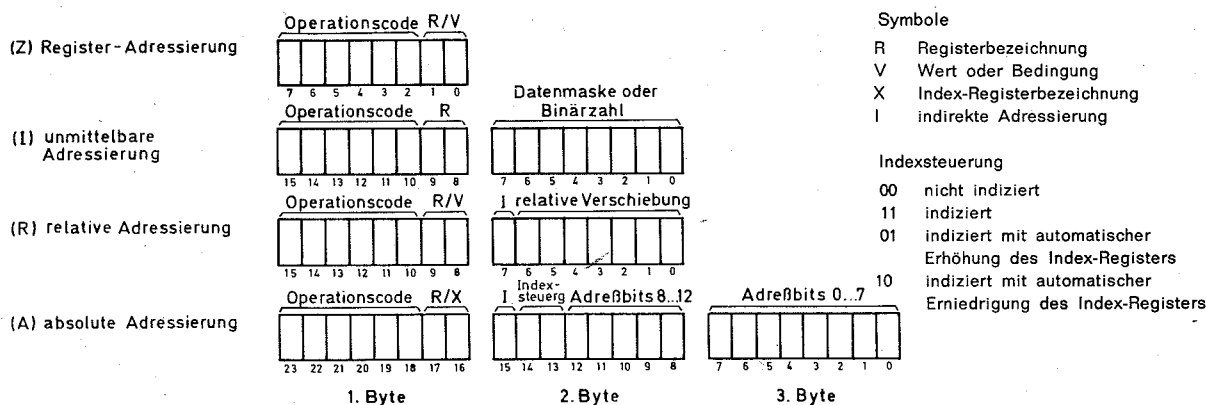


Bild 6. Einige Befehlsformate des Mikroprozessors 2650

A wird also festgelegt, welche der in Bild 6 aufgeführten Adressierungsarten angewandt werden soll. Als Beispiel erläutern wir im folgenden die Ausführung des Befehls LODA („lade absolut“).

Decodieren des Befehls

In der zweiten Spalte der Befehlsliste steht der Operationscode; das ist der in der Maschinsprache des 2650 geschriebene Befehl. Der Mikroprozessor kann Befehle nur in dieser binären Form verstehen. In dieser Form sind sie auch im Programmspeicher abgelegt.

Der Operationscode beansprucht die ersten sechs Bits ²⁾ des ersten Befehlsbytes. Dementsprechend werden auch die betreffenden Felder der in Bild 6 angegebenen Befehlsformate als Operationscode-Felder bezeichnet.

¹⁾ Die Ziffer bezieht sich auf die Anzahl der Bytes, der Buchstabe auf das Format (Bild 6)

²⁾ Mit 6 bit lassen sich nur 64 unterschiedliche Operationscode-Wörter bilden. Bei einigen, im Auszug der Befehlsliste nicht aufgeführten Befehlen werden jedoch die restlichen zwei Bits des ersten Befehlsbytes mit zur Befehlsdefinition verwendet, so daß man beim Mikroprozessor 2650 auf insgesamt 75 Befehle kommt.

Die 6 bit des Operationscodes drücken den mnemonischen Befehl LODA durch den Maschinenbefehl 000 011 aus. Sie enthalten die oben erwähnten Angaben a) und b); denn bei genauer Betrachtung der Befehlsliste sieht man, daß die Befehlsart (LOD, STR, ADD, SUB usw.) durch die ersten vier Bits des Operationscodes festgelegt ist, während die letzten zwei Bits zur Kennzeichnung der Adressierungsarten Z, I, R, A dienen. Die Angabe A (absolute Adressierung) — gekennzeichnet durch 11 in den letzten zwei Bits des Operationscodes — enthält für den Prozessor die Mitteilung, daß noch weitere Angaben über die spezielle Art der Adressierung (direkt oder indirekt; Indizierung) folgen.

Die Angabe c) findet der Mikroprozessor beim Befehl LODA in den letzten beiden Bits des ersten Befehlsbytes. Es sind die mit R/X bezeichneten Felder in Bild 6, Zeile (A). Das R steht für Register, d. h. diese beiden Bits kennzeichnen — bei diesem Befehlstyp — eines der vier 8 bit-Register R_0 , R_1 , R_2 , R_3 oder R_0 , R_1' , R_2' , R_3' (Bild 2), in welches das Datenwort geladen werden soll. In unserem Beispiel sei es das Register R_0 .

Mit dem ersten Befehlsbyte erhält also der Mikroprozessor folgende Angaben:

Art der Operation und Adressierungsart: Es handelt sich um eine LODA-Operation. Das erkennt der Mikroprozessor aus dem Operationscode 000 011.

Ziel-Register: Hierüber geben die letzten beiden Bits des ersten Befehlsbytes Auskunft.

Das erste Byte des ersten Befehls lautet somit im vorliegenden Beispiel

0000 11 00 in 2650-Maschinensprache,
LOD A R0 in 2650-ASSEMBLER-Sprache.

Die hier besprochenen Bedeutungen werden durch die Decodier- und Steuerlogik (Bild 4) entschlüsselt, und es werden in Abhängigkeit vom Befehl bestimmte Impulsfolgen ausgelöst, mit denen u. a. Gatter freigegeben, Zähler betätigt oder Flipflops gesetzt werden. Einige dieser Operationen, mit denen der Mikroprozessor den gegebenen Befehl praktisch ausführt, werden im folgenden, wenn wir die Bearbeitung des Befehls LODA betrachten, noch näher erläutert.

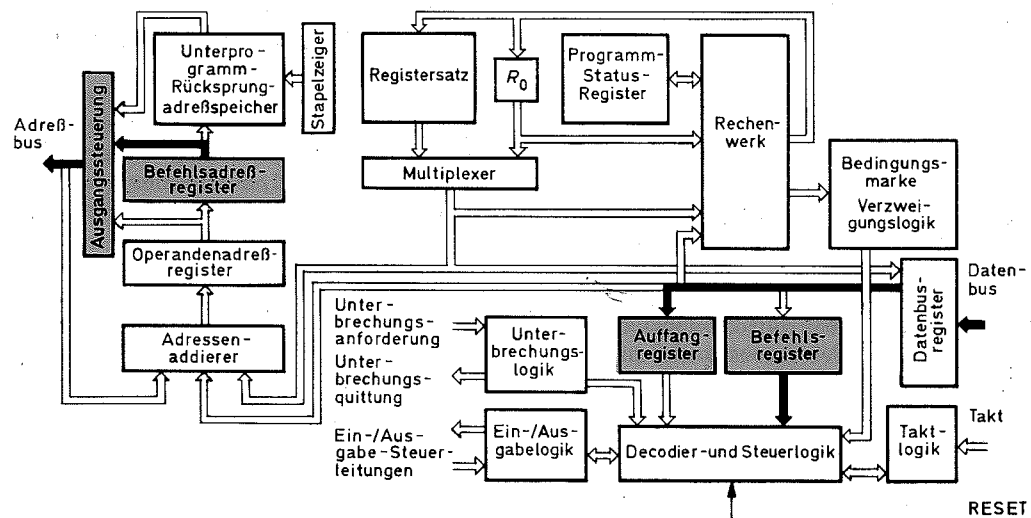


Bild 7. Ausgeben der Befehlsadresse 1 und Holen des zweiten Befehlsbytes

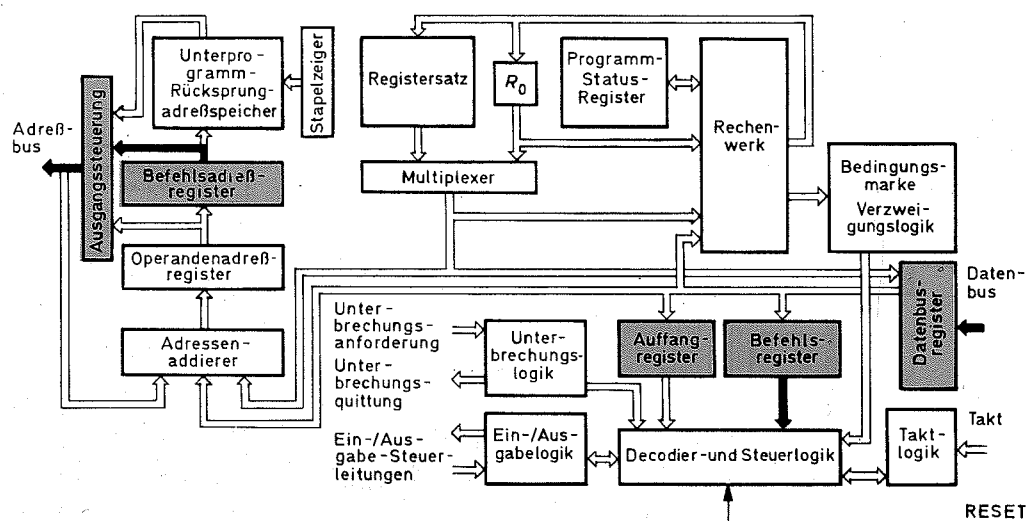


Bild 8. Ausgeben der Befehlsadresse 2 und Holen des dritten Befehlsbytes

Ausführen des Befehls

Nach der Befehlsliste gehört zum Befehl LODA das Format 3 A, d. h., es handelt sich um einen 3 Byte-Befehl vom Format A (Bild 6). Aufgrund dieser Information veranlaßt die Decodier- und Steuerlogik, daß auch das zweite und dritte Byte des Befehls nach dem Schema von Bild 5 aus dem Programmspeicher geholt wird:

- Der Adressenaddierer, der wie ein Zähler arbeitet, schaltet einen Schritt weiter; er gibt also jetzt die Speicheradresse 1 auf das Befehlsadreibregister und damit über die Ausgangssteuerung auf den Adreßbus.
- Das unter der Adresse 1 gespeicherte zweite Byte des Befehls gelangt auf dem Datenbus zum Mikroprozessor und wird in das Auffangregister aufgenommen (Bild 7).
- Das Befehlsadreibregister wird vom Adressenaddierer mit der Adresse 2 geladen.
- Der Mikroprozessor holt unter der Adresse 2 das dritte Byte des Befehls aus dem Speicher und legt es im Datenbusregister ab (Bild 8).

Der weitere Verlauf hängt nun davon ab, welche Art der absoluten Adressierung der Programmierer gewählt hat. Hierfür bietet der Mikroprozessor 2650 acht Möglichkeiten, die durch die Befehlsbits 13, 14, 15 (Bild 6) im zweiten Befehlsbyte gekennzeichnet werden. Darauf soll hier nicht näher eingegangen werden. In unserem Beispiel wollen wir uns für die einfachste dieser Möglichkeiten, die „direkte, nicht indizierte“ absolute Adressierung entscheiden. Die Bits 13, 14 und 15 sind dann alle 0.

Aufgrund dieser Information weiß die Decodier- und Steuerlogik des Mikroprozessors, daß die Bits 0 bis 12 als effektive Adresse der zu ladenden Daten aufzufassen sind und daß diese Adresse sich, wie beschrieben, im Datenbusregister (Bits 0 bis 7) und im Auffangregister (Bits 8 bis 12) befindet. Beträgt die Adresse z. B. $12A_{16}$ (hexadezimal), so lauten unsere drei Befehlsbytes, mit denen der Programmspeicher in den Adressen 0, 1 und 2 zu programmieren wäre:

Oper.-Code						R/X	
0	0	0	0	1	1	0	0
LOD						A	R_0

Adreßbits																									
12												8			7							0			
0 0 0 0 0 0 0 1												0 0 1 0 1 0 1 0													
dir. nicht indiziert												2							$10_{10} = A_{16}$						
$12A_{16}$																									

Der Mikroprozessor hat jetzt alle Informationen, die er zur Ausführung des eigentlichen Ladebefehls braucht:

- Art der Operation: LOD, d. h. übertrage Daten.
- Woher die Daten geholt werden sollen: Aus dem Speicher unter der Adresse, die durch die Adreßbits 0 ... 12 gegeben ist.
- Wohin die Daten sollen: In das Register R_0 des Mikroprozessors.

Übertragen der Daten

Zunächst befindet sich die Adresse für die zu holenden Daten noch im Datenbus- und Auffangregister des Mikroprozessors. Um sie auf den Adreßbus zu geben, muß sie in das Operandenadreibregister gebracht werden (Bild 9). Das wird wiederum von der Decodier- und Steuerlogik veranlaßt. Sie erzeugt aufgrund des Operationscodes 000011, der immer noch im Befehlsregister gehalten wird, die für die Übertragung erforderlichen Impulsfolgen.

Anschließend trennt der mit Ausgangssteuerung bezeichnete Block, der nichts anderes ist als ein Multiplexer, die Ausgänge des Befehlsadreibregisters vom Adreßbus ab und verbindet die Ausgänge des Operandenadreibregisters mit dem Adreßbus. Damit liegt die Adresse $12A_{16}$ am Speicher, und der Mikroprozessor holt sich das betreffende Datenwort über den Datenbus in das Datenbusregister (Bild 5). Von hier aus gelangt es über das Rechenwerk, das im Falle unseres

Valvo-Brief
13. April 1977
Seite 6

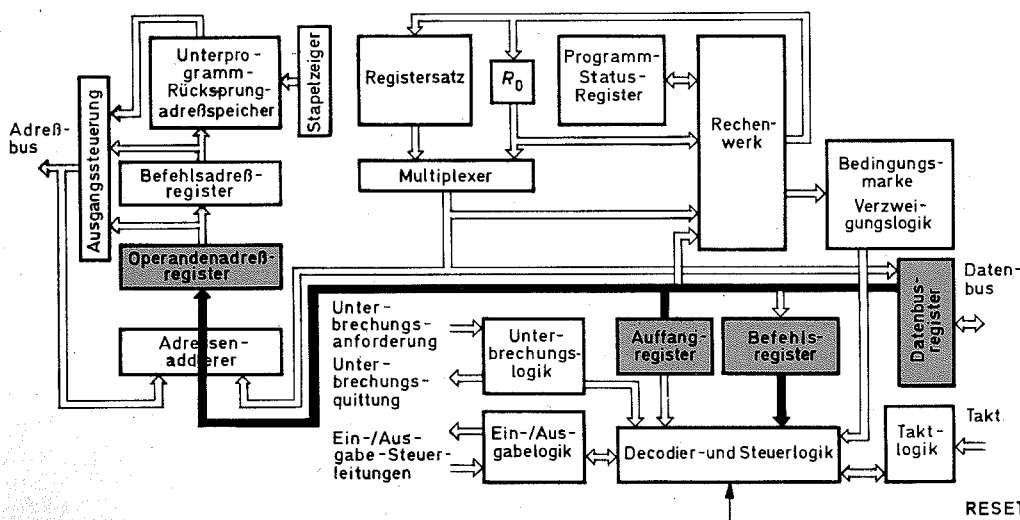


Bild 9. Übertragen der Adresse in das Operandenadreibregister

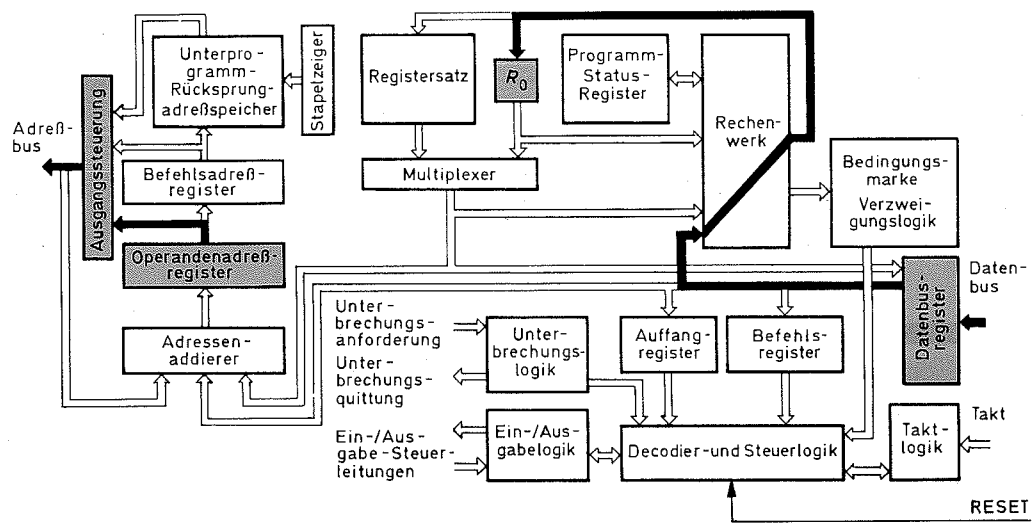


Bild 10. Ausgeben der Adresse und Laden des Operanden in das Register R_0

Beispiels keine Verarbeitung des Datenworts vornimmt, in das Register R_0 (Bild 10). Damit ist die Ausführung des ersten Befehls abgeschlossen.

Vorbereiten auf den nächsten Befehl

Während die Daten übertragen und gegebenenfalls im Rechenwerk verarbeitet werden, erledigt der Prozessor auch noch andere Aufgaben, die nicht unmittelbar mit der Ausführung des laufenden Befehls zusammenhängen, die jedoch für die Fortführung des Programmablaufs wichtig sind. So wird z. B. der Inhalt des Adressen-addierers um einen weiteren Schritt erhöht. Der Prozessor bereitet sich also — noch während der letzte Schritt des laufenden Befehls von ihm ausgeführt wird —, darauf vor, den nächsten Befehl aus dem Programmspeicher zu holen. Da der erste Befehl 3 Byte lang war, hat das Befehls-adreßregister bereits die Stellungen 0, 1 und 2

durchlaufen und steht jetzt auf Adresse 3. Unter dieser Adresse kann dann vom Programmspeicher das erste Byte des zweiten Befehls geholt werden.

Ausführungszeit

Die Ausführung des ersten Befehls ist damit abgeschlossen. Es wurde ein 8 bit-Datenwort aus dem Datenspeicher in das Register R_0 des Mikroprozessors übertragen. Die für diese Operation benötigte Zeit kann man aus der letzten Spalte der Befehlsliste ermitteln. Hiernach beansprucht die Ausführung eines Befehls LODA vier Zyklen. Ein Zyklus des Mikroprozessors 2650 dauert 2,4 μ s. Das entspricht drei Taktperioden bei der maximalen Taktfrequenz von 1,25 MHz. Der gesamte Ablauf, dessen hier vorliegende, vereinfachte Beschreibung vielleicht 10 Minuten Leszeit beansprucht hat, wird also vom Mikroprozessor in 9,6 μ s bearbeitet.