Einführung in die Mikroprozessorarchitektur

Computer sind Geräte, welche eine Liste von Befehlen abarbeiten. Diese Befehle befinden sich im *Speicher*. Ausgeführt werden diese Befehle von einem *Prozessor*. Prozessor und Speicher sind durch *Busse* miteinander verbunden.

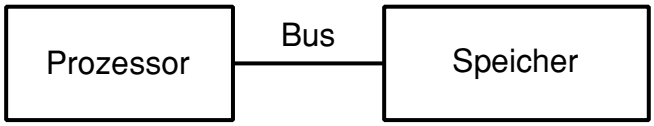

Illustration 1: Einfachste Darstellung eines Computers

Man kann sich den Speicher wie eine lange Liste mit Zeilennummern vorstellen. In jeder Zeile steht ein Befehl.

Veranschaulicht sieht das in etwa so aus:

Zeile	**Befehl**
1	Gehe einkaufen!
2	Putze das Fahrrad!
3	Staubsauge das Wohnzimmer!
4	Lese die GAG-News!

Tabelle 1

Grundsätzlich funktioniert das so: Der Prozessor nennt dem Speicher die Zeilennummer, deren Befehl er wissen möchte. Der Speicher nennt dem Prozessor diesen Befehl, und der Prozessor führt ihn aus.

Statt Zeile sagt man beim Computer jedoch *Adresse*. Technisch umgesetzt wurde dieser Ablauf mit Hilfe eines *Programmzählers* (englisch *programm counter*, kurz *PC*), einem *Adress-* und einem *Datenbus*.

Im Programmzähler steht die Adresse, welche der Prozessor wissen möchte. Der Programmzähler fängt nach dem Anschalten bei Null an und wird nach jeder Ausführung eines Befehls automatisch erhöht. Der Programmzähler ist über den *Adressbus* mit dem Speicher verbunden. Der Speicher gibt den Inhalt der Adresse, welcher im PC steht und über den Adressbus an den Speicher gemeldet wird, auf den Datenbus aus. Der Datenbus ist ebenfalls mit dem Prozessor verbunden. Der Prozessor "sieht" so, was an der Adresse steht, dessen Adresse er über den Adressbus an den Speicher meldet. Dies ist eine ganz starke Vereinfachung des tatsächlichen Vorgangs. Aber im Prinzip funktioniert es so.

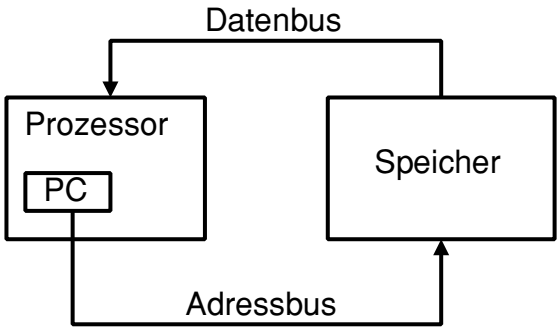

Illustration 2: Computer mit Programmzähler (PC), Adress- und Datenbus

Während der Mensch viele verschiedene Zeichen kennt, z. B. die Ziffern 0 bis 9 oder die 26 Buchstaben des Alphabets, arbeitet der Computer für gewöhnlich nur mit zwei verschiedenen "Zeichen". Diese können z. B. mit verschiedenen Spannungswerten verwirklicht werden (es liegt Spannung an oder nicht). Man kann sich das wie einen Schalter vorstellen, der zwei verschiedene Positionen einnehmen kann: an oder aus. Man schreibt dafür auch 0 oder 1. Jeder Schalter, z. B. ein Lichtschalter, ist damit auch automatisch ein Speicher. Er kann sich nämlich die Stellung "merken".

Würde nun jede Adresse des Speichers nur über einen einzigen Schalter verfügen, also nur die Zustände an oder aus annehmen können, könnte man nicht allzuviel machen. Denn dann könnte der Computer im besten Fall nur über zwei verschiedene Befehle verfügen. Er könnte kaum etwas unterscheiden.

Das gleiche betrifft auch den Programmzähler (PC): würde dieser nur über einen einzigen Schalter verfügen, könnte er nur die zwei Adressen 0 und 1 vom Speicher unterscheiden. Also nicht weiter zählen als bis eins. Er würde abwechselnd hin- und herschalten (an und aus, bzw. 0 und 1). Das wäre alles.

Man arbeitet deshalb mit einer Vielzahl von Schaltern, mit einer Gruppe von Schaltern an jeder Adresse.

Die aktuelle Adresse steht im Programmzähler. Diese Adresse wird über den Adressbus dem Speicher mitgeteilt. Die Stellungen all der Schalter an der betreffenden Adresse im Speicher werden dann vom Speicher über den Datenbus dem Prozessor rückgemeldet. Ein Bus besteht hier also aus einer Vielzahl von Leitungen und dient einem gemeinsamen Zweck, einem gleichzeitigen Vorgang.

Wieviele verschiedene Kombinationen können nun eine bestimmte Anzahl von Schalter haben? Bei einem Schalter wissen wir: er kann die Stellungen an oder aus (wir schreiben hierfür 0 oder 1) haben. Kombiniert man mehrere Schalter, schreibt man die möglichen Kombinationen am besten unter Verwendung eines Übertrages einfach hin. Ein Übertrag bedeutet, daß man einen weiteren Schalter hinzufügt, wenn man alle möglichen Kombinationen der bisherigen Schalter durch hat. Man merkt sich so diesen Übertrag. Dann fängt man bei den hinteren Schaltern wieder von vorne zu zählen an:

0		0	
1		1	
10	(1. Übertrag)	2	
11		3	
100	(2. Übertrag)	4	
101		5	
110		6	
111		7	
1000	(3. Übertrag)	8	
1001		9	
1010		10	(1. Übertrag)
1011		11	
1100		12	
1101		13	
1110		14	
1111		15	

Tabelle 2

Man hat eine gewisse Anzahl von verschiedenen Zeichen, die man durchzählt. Sind alle Zeichen durch, fügt man eine Stelle hinzu (Übertrag) und fängt wieder von vorne an. Das funktioniert ungeachtet der Anzahl von Zeichen immer gleich. Hat man weniger Zeichen zur Verfügung, findet ein Übertrag früher statt. Der Übertrag steht normalerweise immer davor. Die linke Stelle ist damit die Höchstwertige, die rechte die niedrigwertigste.

Wir sehen also: Mit einem Schalter sind zwei verschiedene Kombinationen möglich. Mit zwei Schaltern vier. Mit drei Schaltern acht. Mit vier Schaltern sind es bereits sechszehn. Die Anzahl der möglichen Kombinationen ist damit 2^n (2^n bedeutet, die Zahl 2 wird n mal mit sich selbst malgenommen, also z. B. für $n = 3$: $2^3 = 2 * 2 * 2 = 8$). Mit acht Schaltern kann man bereits $2^8 = 256$ verschiedene Kombinationen einstellen. Auf diese Anzahl hat man sich anfangs für sehr viele Computer geeinigt wie den Commodore 64 oder Schneider CPC.

Weil viele Schalter schnell sehr unübersichtlich werden, kann man diese anders darstellen. In Tabelle 2 sehen wir rechts die Folge im Zehnersystem. Dieses System verwendet zehn verschiedene Zeichen (0, 1, 2, 3, 4, 5, 6, 7, 8 und 9) und lernen wir in der Schule. Wir können statt 1111, wie in der ersten Spalte angegeben, also auch 15 schreiben (das hat dann die Bedeutung: vier Schalter sind gesetzt). Es gibt hier eine eindeutige Zuordnung.

Weil es sich besser aufgeht, hat man dem Zehnersystem die sechs weiteren Zeichen A, B, C, D, E und F hinzugefügt. Ergänzen wir die Tabelle 2 damit und schreiben wir das ganze noch einmal hin:

0	0	0
1	1	1

10	(1. Übertrag)	2		2	
11		3		3	
100	(2. Übertrag)	4		4	
101		5		5	
110		6		6	
111		7		7	
1000	(3. Übertrag)	8		8	
1001		9		9	
1010		10	(1. Übertrag)	A	
1011		11		B	
1100		12		C	
1101		13		D	
1110		14		E	
1111		15		F	
1000	(4. Übertrag)	16		10	(1. Übertrag)

Tabelle 3

Wie man in Tabelle 3 sieht, findet hier der erste Übertrag erst an 17-ter Stelle oder Zeile statt. Damit entsprechen vier Schalter mit den möglichen Kombinationen an (0) und aus (1) immer einer Stelle dem System in der dritten Spalte. Wenn man sechszehn verschiedene Zeichen verwendet, läßt sich mit nur einem dieser Zeichen immer eindeutig eine mögliche Kombination von vier Schaltern darstellen. Das ist von Vorteil.

Natürlich lassen sich diese Systeme leicht verwechseln, da man ja mitunter die gleichen Zeichen verwendet. Ohne weitere Angabe ist unklar, was 10 sein soll. Man muß deshalb immer angeben,was gemeint ist!

In der ersten Spalte von Tabelle 3 verwenden wir *zwei* verschiedene Zeichen. Deshalb heißt dieses System Dualsystem (von lat. duo = zwei).

In der zweiten Spalte von Tabelle 3 verwenden wir *zehn* verschiedene Zeichen. Deshalb heißt dieses System Dezimalsystem.

In der dritten Spalte von Tabelle 3 verwenden wir *sechszehn* verschiedene Zeichen. Deshalb heißt dieses System Sechszehner oder Hexdezimalsystem. Zur Markierung schreibt man in der Regel ein & vor die Zahl. Also z. B. &1F.

Statt von Schaltern spricht man in der Fachsprache jedoch von Bits. Spricht man von Bitbreite, ist damit gemeint, mit wievielen Bits das System gleichzeitig arbeitet, also in einer Zeile oder an einer Adresse stehen. Typisch sind 8 Bit, 16 Bit, 32 Bit oder mittlweile 64 Bit.

Ein Byte sind acht Bit. Viele frühe Computer wie der Commodore 64 oder Amstrad CPC arbeiten mit einem Byte bzw. acht Bit. Dazu zählt aber auch der relativ neue Mega65 von

Trentz Elektronik, welchen man durchaus als Nachfolger des nie auf den Markt gekommenen Commodore 65 sehen kann. Der Acorn Archimedes arbeitet mit 32 Bit oder 4 Byte. Diesen 32 Bit gibt man auch die Einheit *Word* (englisch für Wort). Ein Word sind also 4 Byte oder 32 Bit.

32 Bit = 4 Byte = 1 Word

Nun enthält der Speicher nicht nur *Befehle* für den Prozessor, sondern auch Daten. Und der Prozessor kann den Inhalt einer Adresse nicht nur *lesen*, sondern auch *schreiben*. Mit Schreiben ist gemeint, daß er die Schalter umstellen, die Bits ändern kann. Ob der Inhalt einer Adresse als Befehl oder als Datum verstanden werden muß, hängt von der Logik der Befehle, vom Programm ab. Hierbei können Fehler passieren. Diese führen dazu, daß ein Programm nicht richtig, nicht wie gedacht funktioniert.

Prozessoren verfügen über mehrere Einheiten von gemeinsamen Schaltern. Diese heißen *Register*. Ein Register entspricht von der Struktur her dem Inhalt einer Zeile oder Adresse im Speicher. Damit arbeitet und rechnet der Prozessor. Der Programmzähler ist ein spezielles Register. In diesem wird mit jedem Takt der Inhalt um eins erhöht, also um eins weitergezählt oder die Schalter bzw. Bits entsprechend verändert wie in Tabelle 3 aufgeführt.

Man kann den Inhalt in diesem Register beeinflussen. Dafür gibt es einen Befehl. Der Prozessor sieht den nächsten Befehl dann dort im Speicher, dessen Addresse im PC steht.