Lerncomputer für Einsteiger:

4-Bit-Simulant

Mit dem Lerncomputer „Microtronic" kam ein Gerät auf den Markt, das sich zusammen mit der Begleitliteratur speziell an Einsteiger ohne Computer-Vorkenntnisse wendet. Obwohl es „nur" einen 4-Bit-Mikroprozessor enthält, wird ein recht interessantes und einfach durchschaubares Prozessor-Modell simuliert, dessen Konzept an modernste 8- und 16-Bit-Rechner erinnert.

Kernstück des Microtronic-Baukastens, der betriebsfertig im Gehäuse (*Bild 1*) mit durchsichtigem Klappdeckel geliefert wird, ist eine kundenspezifische Version des 4-Bit-Einchip-Computers TMS-1600 von Texas Instruments. Um dessen Register- und Befehlsstruktur, die für heutige Verhältnisse etwas primitiv wirkt, braucht sich der Anwender aber nicht zu scheren: Im ROM auf dem TMS-1600-Chip befindet sich nämlich ein Interpreter-Programm, das (ähnlich wie ein Basic-Interpreter in einem Tischcomputer) ein wesentlich intelligenteres Mikroprozessor-Modell simuliert (*Bild 2*). Diese Simulation kostet natürlich Zeit, weil jeder eingegebene Befehl vom TMS-1600 als eine ganze Folge von internen Maschinenbefehlen interpretiert werden muß. Der Microtronic führt deshalb die Programme eher mit der Geschwindigkeit eines programmierbaren Taschenrechners als mit der von typischen Mikroprozessoren aus. Hinzu kommt, daß der in PMOS-Technologie hergestellte TMS-1600 ohnehin nicht gerade der schnellste ist. Für den vorgesehenen Zweck, nämlich das Lernen am „lebenden Objekt", um das man beim Einsteigen in die Computertechnik nicht herumkommt, spielt dies jedoch keine Rolle. Viel wichtiger ist da schon die Überschaubarkeit des simulierten Prozessormodells.

12-Bit-Befehle

Das Microtronic-Prozessormodell arbeitet mit 16 Allzweck-Registern. In jedem von ihnen können arithmetische und logische Aufgaben durchgeführt werden. Dieses Konzept findet man auch in modernen 16-Bit-CPUs und stellt eine wesentliche Verbesserung gegenüber dem Akkumulator-Konzept herkömmlicher 8-Bit-Prozessoren dar. Jedes dieser Register ist vier Bit breit und kann also eine hexadezimale oder dezimale Ziffer festhalten.

Ein Unterprogramm-Rücksprungregister hält die Adresse fest, an der ein Unterprogramm-Sprungbefehl steht. Da für diesen Zweck kein Stack, sondern nur ein einziges Register vorhanden ist, kann man Unterprogramme nicht verschachtelt ausführen, man ist auf eine „Ebene" begrenzt. Für komplexe Programme könnte dies hinderlich werden, reicht im Normalfall aber aus.

Der Programmspeicher besteht aus 256 Worten von je 12 Bit. Das bedeutet, daß jeder Befehl mit drei hexadezimalen Ziffern dargestellt wird (es gibt nur 1-Wort-Befehle). Ein Beispiel: Der Befehl C00 bedeutet „Sprung an die Adresse hex 00". Das C ist der Code für „GOTO", und die restlichen 8 Bits stellen die Zieladressen dar. Ein weiteres Beispiel: Der Befehlscode F31 bedeutet „Anzeige von drei Registern ab Register 1 auf dem Siebensegment-Display".

Die Befehle sind also so aufgebaut, daß man sie sich auch in hexadezimaler Form sehr leicht merken kann. Und sie sind zum Teil recht leistungsfähig, was ein unnötiges „Dickbrettbohren" erspart. Anzeige, Tasteneingabe und Multiplikation sind drei Beispiele dafür, die weit über den Befehlssatz des „nackten" TMS-1600 hinausgehen.

Erweiterungsmöglichkeiten

Es ist etwas hinderlich, daß beim Ziehen des Netzsteckers der Speicherinhalt und damit das eingegebene Programm verlorengehen. Für dieses Problem gibt es zwei Lösungsmöglichkeiten. Die erste besteht darin, das PMOS-RAM-IC im Microtronic aus der Fassung zu ziehen und durch ein pinkompatibles CMOS-IC zu ersetzen, das vom gleichen Hersteller wie der Microtronic-Kasten erhältlich ist. Eine 9-V-Batterie sorgt dann über Monate hinweg dafür, daß der Programmspeicher nicht flüchtig ist.

Die zweite Möglichkeit ist die Nachrüstung des ebenfalls erhältlichen Kassetten-Interface. Zusammen mit einem handelsüblichen Kassettenrecorder kann man den Programmspeicher (256 Byte) abspeichern und wieder laden, was allerdings jeweils vier Minuten dauert. Dies entspricht einer Geschwindigkeit von rund 13 Bit/s – und hier ist wohl eine gewisse Kritik nicht unberechtigt,

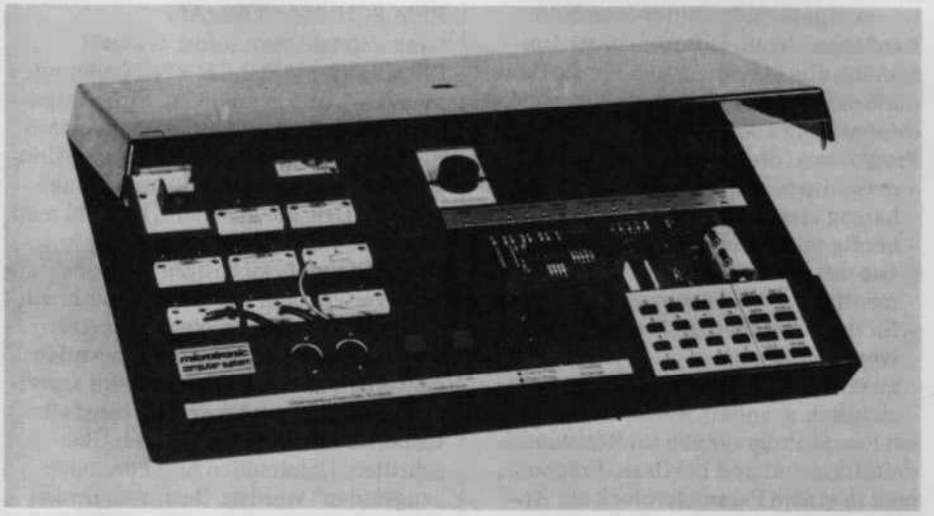

Bild 1. Unter dem Kunststoff-Klappdeckel des „Microtronic"-Lerncomputers lassen sich Bauelemente für Hardware-Erweiterungen unterbringen

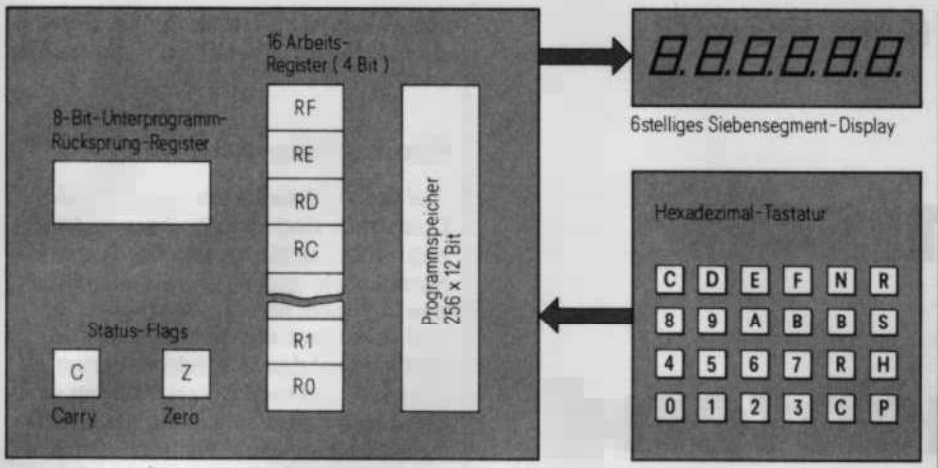

Bild 2. Funktionsmodell des „Microtronic". Die Datenbreite beträgt 4 Bit, die Befehlsbreite 12 Bit

denn der gleiche Speicherinhalt wird von manch anderem Computer mit 800 Bit/s oder mehr in weniger als vier Sekunden auf Kassette abgespeichert. Es ist zwar einzusehen, daß der TMS-1600 nicht so schnell wie ein Z80 oder 6502 ist und deshalb Probleme bei den Kassetten-Routinen im Betriebssystem bekommt, aber die hier erreichte Geschwindigkeit kann man ja fast noch „mithören".

Begleitliteratur

Eine Stärke des Microtronic sind dagegen die beiden mitgelieferten Handbücher. Sie sind wirklich auch für absolute Laien noch verständlich – wenn man nur ein wenig Interesse an der Materie mitbringt, kann man sicher sein, nach deren Durcharbeiten bei Diskussionen von Computer-Freaks mitreden zu können. Die Begleitliteratur besteht aus zwei aufeinander aufbauenden Teilen, wobei der zweite Teil auch auf Kombinationsmöglichkeiten mit dem Elektronik-Baukasten „Elotronic" des gleichen Herstellers eingeht. Damit ist es dann auch möglich, den Computer zur Steuerung externer Geräte heranzuziehen – bis hin zur Verbindung mit einer elektrischen Modelleisenbahn.

Zusammenfassend kann gesagt werden, daß der „Microtronic" ein optimal an Lernzwecke angepaßter Mikrocomputer ist, ein Gerät zum ersten Kennenlernen der Funktionsweise von Computern. Wer mit ihm gearbeitet hat, dem fällt ein späterer Aufstieg zu Basic-Computern oder Entwicklungssystemen leichter.

Herwig Feichtinger

Grafik mit dem MX-80

Woher nimmt man das achte Bit beim Apple?

Wenn Sie sich zu Ihrem Apple-II einen MX-80 gekauft und endlich alle Druckersteuerzeichen ausgeknobelt haben, kommt bald der Moment, an dem Sie Lust verspüren, die im Zeichenvorrat des MX-80 vorhandenen Grafiksymbole zu verwenden. Sie errechnen sich die nötigen CHR$-Befehle und versuchen es. Aber Sie werden eine herbe Enttäuschung erleben.

Das für Grafiksymbole benötigte achte Bit ist auf der Interfaceplatine fest auf Masse gelegt, es ist also nicht ansteuerbar. Wenn Sie dann weitersuchen, werden Sie feststellen, daß die CHR$-Befehle das achte Bit gar nicht ansteuern, obwohl CHR$(128) bis CHR$(255) durchaus erlaubt sind. Wie kann man dieses Bit trotzdem ansteuern?

Das Prinzip ist folgendes: Man verwendet einen TTL-Ausgang (AN2) des Game-Paddle-Anschlusses. Dazu ist lediglich eine kleine Hardwareänderung notwendig. Auf der Interfaceplatine befindet sich eine Drahtbrücke unterhalb des IC 3A (EPROM), bezeichnet mit P4. Sie löten oder schneiden diese Drahtbrücke auf und verbinden das rechte Lötauge von M4 oder das linke Lötauge von P4 mit dem Game-Paddle-Anschluß AN2 (Pin 13). Der Anschluß auf der Interfaceplatine muß mit aller Sorgfalt hergestellt werden. Wenn Sie das falsche Lötauge erwischen, könnte im Computer das Ansteuer-IC des Game-Paddle-Anschlusses beschädigt werden. Prüfen Sie daher auf der Rückseite der Interfaceplatine, ob Sie wirklich ein Lötauge anschließen, das mit dem Stecker des Druckers verbunden ist. Die andern, „falschen" Lötaugen sind mit Masse bzw. IC 2A (74LS175) verbunden.

Wenn Sie diese Operation glücklich hinter sich gebracht haben, können Sie den Drucker mit POKE-16291,0 auf Grafikzeichen und mit POKE-16292,0 auf alphanumerische Zeichen umschalten.

Hinweis: Normalerweise ist der TTL-Ausgang AN2 des Computers beim Einschalten auf Low (also Drucker im Normalbetrieb). Bei kurzzeitigem Ein- und Ausschalten des Computers kann er jedoch auf High gehen, Sie müssen dann mit POKE-16292,0 den Drucker auf Normalbetrieb schalten.

Um das Blatt lückenlos mit Grafikzeichen bedrucken zu können, müssen Sie den Drucker auf den Zeilenabstand 7/72 Zoll umstellen (Befehl Esc 1 ≙ CHR$(27); CHR$(49)). Die Grafikzeichen sind in einer 2×3-Matrix angeordnet. Sie können die Codes der Anleitung entnehmen. Es gibt aber auch einen einfacheren Weg, sie zu berechnen. Die *Tabelle* zeigt, wie man das macht.

Dieter Fischlin

Berechnung der MX-80-Grafikcodes

Position des Punktes	Wert
Oben links	1
Oben rechts	2
Mitte links	4
Mitte rechts	8
Unten links	16
Unten rechts	32
Grundwert	32
Beispiel: 	1 + 8 + 16 + 32 = 57 PRINT CHR$(57)