

Kennen Sie Ihren C 64?

Was ist das Faszinierende am C 64, dem größten Verkaufserfolg, seit es den Begriff Heimcomputer gibt? Warum schätzen ihn weltweit über vier Millionen Menschen, diesen kleinen, unscheinbaren Kasten mit der großen »64« auf dem Gehäuse? Die Antwort finden Sie in unserer technischen Liebeserklärung.

E ist schon ein Unikum: Sieht aus wie eine Tastatur und ist doch ein vollwertiger Computer, besitzt keine genormten Schnittstellen und kann doch jede Schnittstellen-norm erzeugen. Reizt auf Anhieb so richtig zum Diskette reinschieben und in einem heißen Aktionspiel zu versinken. Doch ein Programmierer muß und will mit diesem Board arbeiten, um anderes als wilde Verfolgungsjagden auf dem Bildschirm zu erleben. Board, das steht dabei für Einplatinencomputer, auf dem alle Baugruppen kompakt vereinigt sind, unabhängig davon, ob man sie braucht oder nicht. Trotzdem muß dieser Computer ein Geheimnis in sich verbergen, denn niemand weiß so recht, warum er zu Tausenden in der Industrie verschwand. Was macht die Industrie mit Computern, deren Standard darin besteht, daß sie keinen haben? Sie baut diese in verschiedenste Meß- und Regeleinheiten ein und aus einem 600-Mark-Computer wird ein 5000-Mark-Steuerungssystem. Möglich wird das alles erst durch das besondere Konzept des C 64. Es fängt beim Aufbau des Betriebssystems an und hört beim Disketten-Aufzeichnungsformat auf. Mit dem Wort »IBM-kompatibel« oder nach »DIN genormte Schnittstellen« kommt man nicht weiter. Und doch gab und gibt es Zeiten, in denen Commodore-Computer nicht in ausreichender Menge produziert werden konnten.

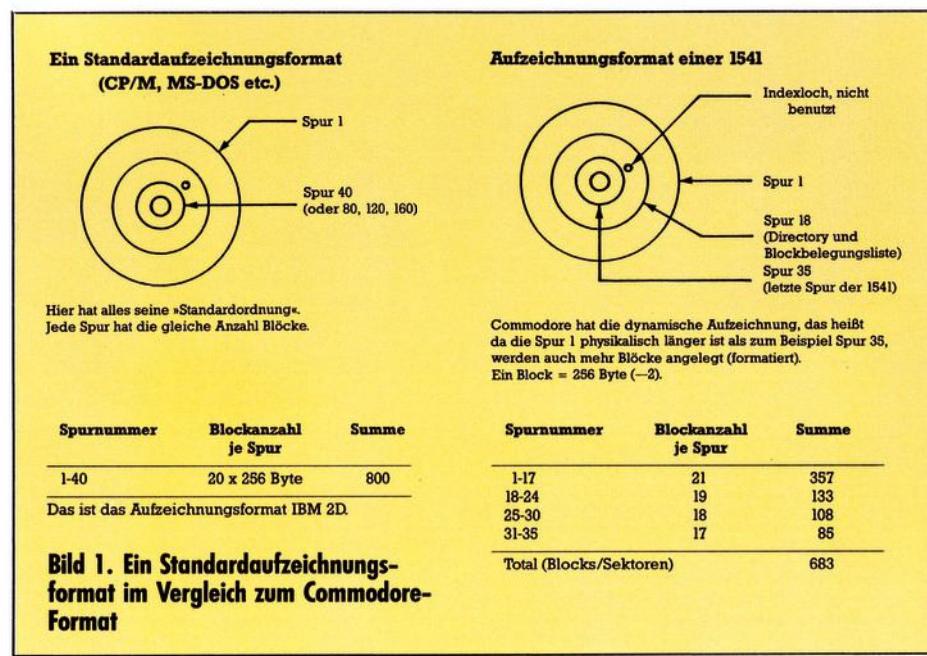
Es ist die Technologie. Eine ge-konnte Mischung aus Frechheit, was den residenten Aufbau des Betriebssystems angeht, Raffinesse, was die Organisation der Systeme betrifft und ein Zusammenspiel zwischen Hard- und Software, das ei-gentlich nicht funktionieren kann, es aber trotzdem vorzüglich tut. Ausge-heckt hat das Ganze der Teil von Commodore, der sich MOS Technologies nennt. Diese Tochterfirma macht alle ICs für Commodore. Der militärische Teil von MOS-Technolo-gies fertigt ICs mit ganz besonderen (und geheimen) Fähigkeiten. Zu ei-ner Zeit, als die beiden Chefent-wickler der legendären CPU 8080

von Intel voller Enttäuschung mein-ten: »Den nächsten Prozessor machen wir besser« und daraufhin die Firma Zilog gründeten, hatten die MOS-Entwickler eine Version eines 16-Bit-Prozessors, basierend auf dem 6500, bereits wieder eingemot-tet. Dem Militär war das alles »zu groß« und zu unhandlich. Sie wollten etwas Kleineres und Schnelleres.

Das ist schon sehr lange her, aber es war dennoch der Grundgedanke für den ROM-residenten Aufbau ei-nes Computersystems. Resident, im-mer präsent, hieß damals die Devise. Denn wenn ein computergesteu-ertes Flugerfassungsradar oder ähnliches erst eine Diskette braucht, um Aktivität zu zeigen, wäre das logischerweise fatal. Bis heute gibt es noch kein einziges Floppy-Laufen-werk, das die Anforderungen des MIL-Standards (nur das Feinste vom Teuren) erfüllt. Dies ist der Grund, warum Commodore-Computer bis-her nie mit einem nachladbaren Dis-ketten-Organisations-System, son-dern immer mit einem fest installier-ten Betriebssystem und selbständi-gen Peripheriegeräten gebaut wurden. Der Amiga scheint hier erst-mals von dieser Leitlinie abzuweichen, aber letztendlich wurde der

Amiga ja auch ursprünglich nicht von Commodore entworfen.

Der zweite Weg, nämlich das Be-triebssystem von einer Diskette (früher waren es Lochstreifen, Lochkar-ten und Magnetbänder) zu laden, wird von vielen anderen Systemen gegangen. Dieser Vorgang, auch »booten« genannt, hat viele Nach- und nur wenige Vorteile. Bestes Bei-spiel hierfür sind Betriebssysteme wie CP/M oder MS-DOS. Leider ba-sieren diese Systeme auf einem fata-lem Rechenfehler. Während näm-lich Großcomputer, deren Leistung von ironischen Zeitgenossen in Tonnen gemessen wird, extrem schnelle (und teure) Speichermedien be-sitzen, plagen sich Millionen von Mi-nisystemen mit lahmen Floppy-Laufen-werkern und verschiedenen Be-triebssystem-Versionen herum. Selt-samerweise halten sich solche Sy-steme für besonders fortschrittlich, obwohl sie dem Anwender das Le-ben unnötig schwer machen. Das al-les geschieht, um möglichst flexibel zu bleiben, denn es könnte ja sein, daß nach der Betriebssystem-Version 2.2 die Version 2.3 folgt. Un-glücklicherweise beinhaltet das aber das Risiko der Inkompatibilität in sich, denn Programme für die



Version 2.2 müssen nicht notwendigerweise auch mit der Version 2.3 funktionieren.

Doch zurück zu Commodore oder besser gesagt zu MOS-Technologies. Am Beispiel des Commodore 64 soll gezeigt werden, was aus diesem Computerkonzept geworden ist und wie man aus der Sicht eines Entwicklers so einen Computer nutzen kann. Es geht nicht nur darum, das riesige Heer von Freaks mit neuen Tricks zu versorgen. Vielmehr sollen gerade Sie einen Einblick in die sogenannte Hexenküche eines Entwicklers bekommen. Denn nach dem weisen Zitat von Cäsar: »Man gebe mir ein Land und ich mache was draus«, versuchen Entwickler eine Basistechnologie zu nutzen, um Systemlösungen jeglicher Art zu schaffen. Das erfordert jedoch harte Auseinandersetzungen mit der Sache. Wenn jemand zum Beispiel über einen Computer schimpft, weil die Floppy am C 64 ausgefallen und/oder trotz aller Turbo, Speedy,

Hyper etc. immer noch zu langsam ist, dann reizt es den Entwickler, neue Wege zu suchen, zu erkennen und zu realisieren.

Der Weg zum Standard

Wenn dies der erste Gedanke war, so muß der zweite Gedanke der Suche nach einem existierenden Standard gelten — und dieser heißt derzeit noch immer IBM. Genauer gesagt, die Art wie IBM-Computer Daten auf Diskette schreiben, nämlich nach Industriestandard durch einen FDC-(Floppy Disk Controller-)Chip und unselbstständigen Peripheriegeräten. Auf diesem FDC steht aber nicht IBM, sondern Western Digital, SMC oder Rockwell. Der Verwendung eines solchen ICs stellte Commodore beim C 64 allerdings einige nicht unwesentliche Hindernisse in den Weg. So verwendet man beispielsweise die virtuelle Aufzeichnung (Bild 1), das heißt, auf der Diskette, bestehend aus Spuren

und Sektoren, wurde nicht gleichmäßig jede Spur mit einer festen Anzahl von Sektoren belegt (wie das der FDC macht), sondern auf fast jeder Spur befinden sich unterschiedlich viele Sektoren. Das erscheint auf den ersten Blick unsinnig, denn wer soll das alles organisieren und verwalten? Doch hier ist ja nicht CP/M oder MS-DOS, wo der Computer jede Kleinigkeit machen muß, am Werke. Bei Commodore verwendete man Computer, die erwarten, daß das Floppy-Laufwerk »mitdenkt«. Man nehme also für so ein Laufwerk einen eigenen Computer, packe ihn in einen eckigen Kasten, schreibe ein autarkes DOS (Disketten-Organisations-System) dazu und lasse das Ganze als »Slave« (Knecht) arbeiten. Wenn also der Computer von der Floppy irgend etwas will, ruft er über die Verbindungsleitung (Bus) den Floppy-Computer. Hat der gerade zu tun, wartet der Computer, bis das Laufwerk bereit ist. Andererseits macht das Floppy-Laufwerk in den Zeiten, in denen es nicht angesprochen ist, nichts anderes, als auf den Computer zu warten. Hat man vergessen, das Laufwerk einzuschalten, das heißt, das Laufwerk meldet sich auf Anruf des Computers nicht, so geschieht etwas, das Sie sicherlich schon kennen: Es wird die Fehlermeldung »Device not present« ausgegeben.

Jedes Peripheriegerät, wie beispielsweise ein Floppy-Laufwerk, wird mit seiner Nummer (0 bis 15) gerufen. So ist es möglich, viele Peripheriegeräte auf einem Bus zu verwalten. So ein Hin und Her auf den Leitungen nennt man übrigens Bus-Management und das wurde auch genormt und zwar in Chicago 1975 und 1978. Diese Norm nennt sich IEEE 488 und wird von den verschiedenen Firmen (Tektronix, HP, etc.) für Meß- und Regelzwecke verwendet. Bei Commodore wird der IEEE 488-Standard aus dem Jahre 1975 verwendet, alle anderen benutzen den von 1978. Wenn man dieses Hintergrundwissen und noch dazu die Schaltungsunterlagen hat, kann man mit einem Computer für 600 Mark einen Plotter für 100 000 Mark perfekt steuern.

So gesehen war die Schöpfung von Commodore schon ein Vorteil: Der Computer setzt nur ein paar Bytes an die Peripheriegeräte ab. Der durchforstet sein Operating-System und weiß, was er zu tun hat, während der Computer wieder frei für andere Aufgaben ist. Das kann man so weit treiben, daß man dem Floppy-Laufwerk riesige Aufgaben über-

Anschlußnummer	Bezeichnung	Erklärung
1	Serial SRQ IN	Eingang für Bedienungsanfrage
2	GND	Dieser Anschluß wird nicht benutzt Massepotential für alle Signale
3	Serial ATN IN/OUT (ATTENTION)	Ein-/Ausgang für Bedienungsauftrag. Meldet sich das gerufene »DEVICE« innerhalb 1 ms nicht, meldet der C 64 »DEVICE NOT PRESENT ERROR«
4	Serial CLK IN/OUT	Takt zur Übergabe eines Bits
5	Serial Data IN/OUT	serieller Kanal für Daten und Befehle
6	nicht angeschlossen	—

Bild 2. Die Signalbelegung des seriellen IEC-Busses

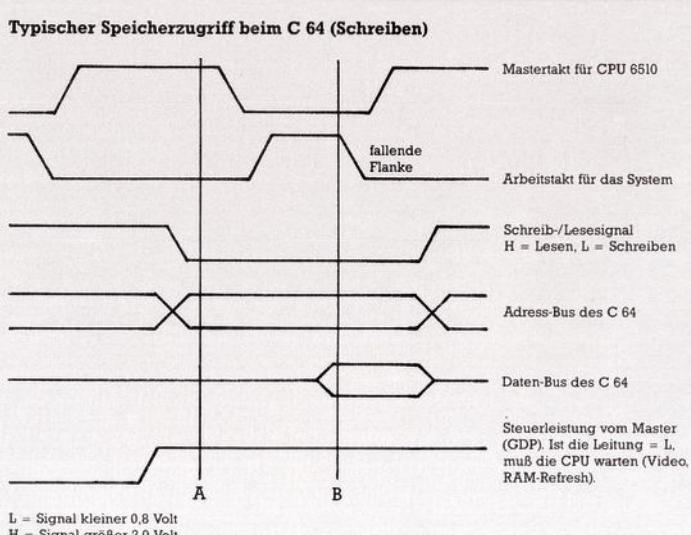


Bild 3. Typischer Speicherzugriff beim C 64

mittelt. Die wiederum schreibt sich alle Daten in einen eigenen Puffer (Zwischenspeicher) und sortiert, organisiert, sucht, ordnet und bearbeitet, wohlgemerkt, ohne daß der Computer dadurch behindert wird.

Während der parallele IEEE 488-Bus nur bei den »größeren« Commodore-Computern zum Einsatz gekommen ist, besitzt der C 64 eine serielle Variante dieses Busses (Bild 2). Dabei werden also die Bits hintereinander mit einem Takt (Clock) durch die Leitung gejagt. Damit aber auch hier das übliche Übertragungsprotokoll weiterverwendet werden kann, hat man noch eine zusätzliche Leitung gelegt: das Attention-Signal, die sogenannte Rufleitung. Der parallele IEEE-488 Bus hat aber 24 Leitungen, also ist diese Lösung ein echtes »Nadelöhr«. Das Problem Nadelöhr zu beseitigen, ist aber nicht der Kernpunkt der Pro-

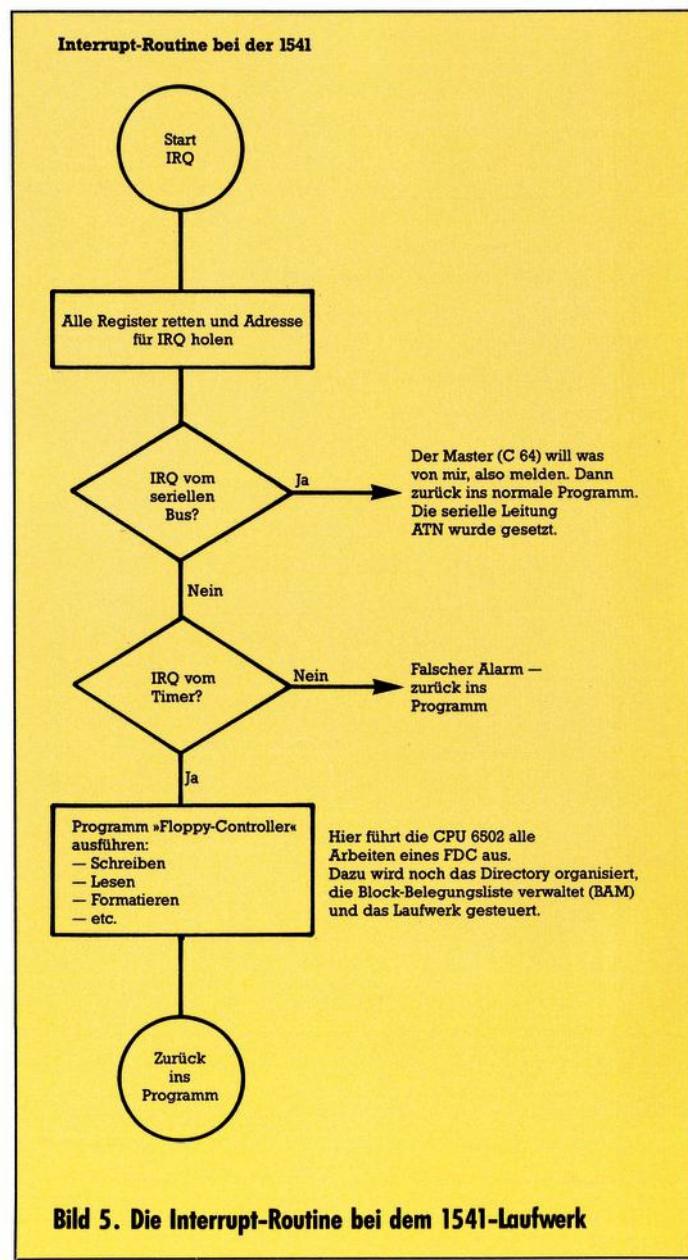
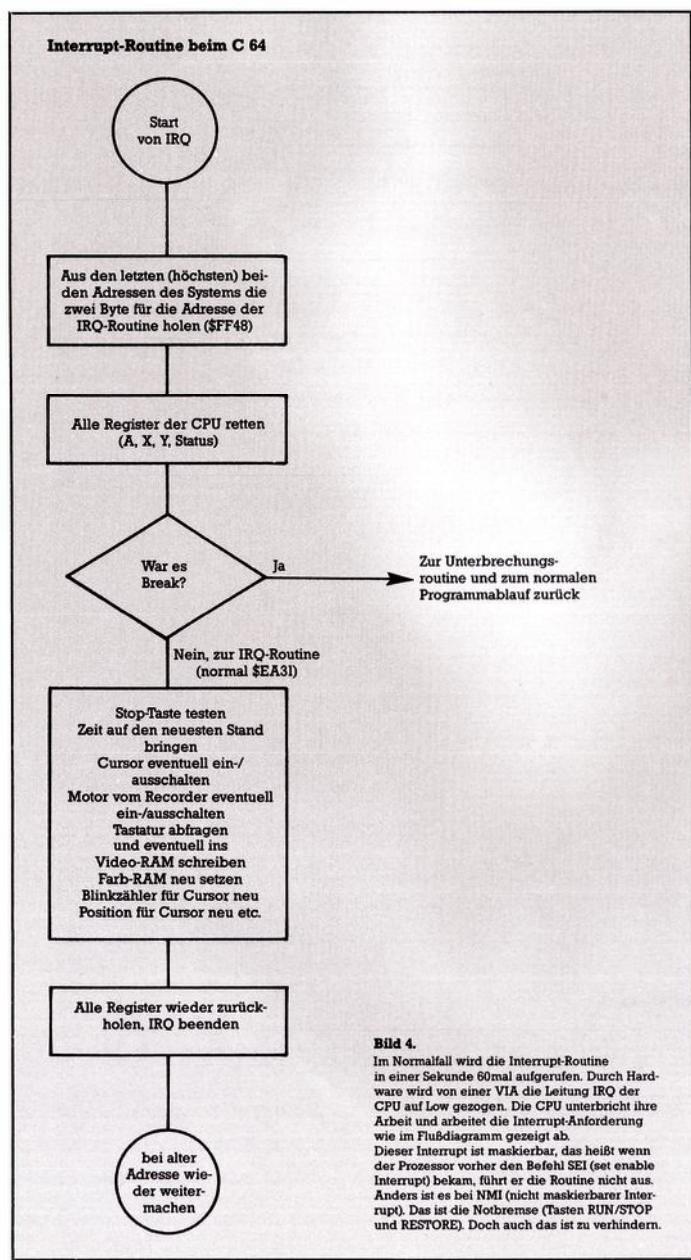
bлемstellung. Vielmehr geht es um das Laufwerk. Commodore benutzt ein Laufwerk (ohne Controller-Platine), auf dem fast alles fehlt, was nach Elektronik aussieht. Ein Standardlaufwerk hat dagegen die gesamte Steuerelektronik »eingebaut«: die Motorsteuerung, die Umwandlung der analogen Signale in digitale (Bits) und vieles mehr. Dieser Standard heißt Shugart (nicht zu verwechseln mit dem Hersteller von Laufwerken). Egal, ob 3-, 3½-, 5¼- oder 8-Zoll-Laufwerke: Überall findet man eine genormte Steckerleiste, Schalter oder Brücken und Anpassungsmöglichkeiten (Ausnahmen bestätigen die Regel). Commodore hat auch das nicht nötig, denn ein eigener Prozessor steuert ja das Ganze. Was tut man aber, wenn die Industrie ein Standardlaufwerk anschließen möchte, weil Sicherheit und Standardformat gefragt sind,

man aber trotzdem dieses wirklich gute DOS nicht missen möchte?

Zunächst nimmt man einfach einen C 64, ein 1541-Laufwerk, einen FDC (Floppy Disk Controller), einige Schaltungsunterlagen und eine gehörige Portion Mut.

Das Problem: Der Kampf der Zeiten

Den Mut braucht man vor allem dazu, sich nicht von den verschiedenen lautenden Datenblättern des FDC und der 1541 aus der Ruhe bringen zu lassen. Im Datenblatt des FDC steht, daß der Controller-Chip mit 8 Megahertz arbeitet, wohingegen der 6502-Prozessor der 1541 lediglich mit 1 Megahertz getaktet wird. Der FDC-Prozessor ist also achtmal schneller, als die Zentral-



einheit (6502) des 1541-Laufwerks. Das kann normalerweise nicht gut gehen, aber wir haben es dennoch probiert. Der 6502 versuchte den FDC auf dem Bus anzusprechen. Wider Erwarten funktionierte das Ganze auf Anhieb. Der 6502 konnte dem FDC zum Beispiel befehlen: Formatiere eine Spur — mit Erfolg. Doch da der FDC mindestens die »Intelligenz« einer 6502 besitzt und dazu noch achtmal schneller ist, wird das Signal, das der FDC nach erfolgreicher Befehlsausführung abgibt, vom 6502 nicht richtig interpretiert, mit der Folge, daß das Statusregister des FDC nicht ausgelesen wird. Dieser fatale Fehler wird nur noch dadurch übertragen, daß der 6502 beim Lesen aus dem FDC die dort angebotenen Daten einfach nicht übernehmen kann und diese damit verlorengehen. So geht es also nicht; eine neue Lösung muß gefunden werden.

Wie eine Zentraleinheit aus einem Speicherbereich Daten übernimmt, ist im Prinzip bei allen Computersystemen der Welt gleich: Der Prozessor setzt auf dem Adressbus diejenigen Bits, um den entsprechenden Baustein (RAM, ROM, PROM, VIA, PIO, etc.) zu adressieren. Dazu muß der betreffende Baustein noch durch das Setzen des »Chip selekt«-Signals selektiert werden. Bei

Schreib-/Lese-Speichern muß der Baustein natürlich auch noch wissen, ob er die Daten anbieten oder annehmen soll. Dafür gibt es zusätzlich noch das »Read/Write«-Signal. Nachdem diese Signale alle »gesetzt« sind, muß dem Baustein eine bestimmte Zeit gegeben werden, um die Daten hervorzukramen und auf dem Datenbus anbieten zu können (Bild 3). Wer bestimmt aber nun diese Zeit? Die Zentraleinheit (CPU)? Nein, es ist der Systemtakt. Dazu ein Beispiel: Auf einer spanischen Galeere gab es Sklaven, die immer, wenn der Trommelschläger den Takt schlug, die Ruder durchziehen mußten. Sie ruderten also im Takt. Taten sie das nicht, ruderten sie meistens nicht mehr lange. Im Computersystem bestimmt der Systemtakt wie die Trommel der spanischen Galeere das Timing des ganzen Systems. Wird während eines Taktes die »Arbeit« nicht getan, stürzt das System ab und nichts geht mehr. Was während eines Taktes geschieht, bestimmt die CPU. Es wird entweder gelesen, geschrieben oder innerhalb der CPU eine Operation ausgeführt. Beim C 64 macht die CPU 6510 diese Arbeit. Sie kann jedoch nicht uneingeschränkt über das Bussystem regieren, sondern muß noch auf den GPD (Grafik Display Prozessor) oder Videochip Rücksicht nehmen (dazu später mehr). Im Normalfall arbeitet die CPU des C 64 mit einem Systemtakt von knapp einem Megahertz pro Sekunde (PAL-Version für Europa), was ungefähr 500000 Taktzyklen entspricht. Wenn man nun in einem Floppy-System einen achtmal

schnelleren Prozessor dazu nimmt und der verlangt, daß die langsame CPU die Daten auch annimmt, braucht man einen Meldeeingang für die CPU. Dieser Meldeeingang kann aber nicht von der 6502-CPU abgefragt werden, weil das ja wieder Zeit braucht. Der richtige Zeitpunkt wäre vorbei, wenn die CPU abfragen würde — ein Teufelskreis, den es zu durchbrechen heißt. Zunächst bieten sich die Eingänge der CPU an, die Unterbrechungen auslösen; die Interrupt-Request-Eingänge. Wird so ein Signal gesetzt, »rettet« die CPU alle momentanen Parameter (Adresse, Daten und Status), holt die IRQ-Adresse, verzweigt zu dieser Routine, führt diese aus, holt alle Parameter wieder zurück und macht an der alten Stelle weiter. Doch allein beim Retten der Daten vor der Interruptroutine (Bild 4) wäre die Zeit schon wieder vorbei und die vom FDC angebotenen Daten könnten nicht angenommen werden. Man braucht also einen Eingang, der direkt in dem Statusregister der CPU eine Veränderung verursacht, unabhängig davon, was die CPU gerade macht (Bild 5).

Das Geheimnis der Hintertür

Dieser fantastische Eingang existiert bei der CPU 6502. Es ist der Eingang »Set Overflow« (SO). Systementwickler benutzen diesen Eingang normalerweise nicht, weil er in der Dokumentation nur als künftige Erweiterung beschrieben wird. Wird der Eingang SO von seinem Normalzustand (High) auf Low-Potential gelegt, wird innerhalb eines Taktes im Statusregister der CPU das Overflow-Bit gesetzt (Bild 6). Hiermit lassen sich also für die CPU sehr schnelle Ergebnisse abfragen. Die CPU löscht das Overflow-Bit mit dem Befehl CLV (Clear Overflow Bit) und hängt in einer Abfrageschleife BVC (Branch on Overflow Clear). Wenn das SO-Signal wieder Low-Potential annimmt, wird die Schleife verlassen und die Daten können angenommen werden (Bild 7). Es ist hiermit möglich, daß die 6502-CPU mit einem achtmal schnelleren Prozessor zusammenarbeitet. Wir haben diesen Eingang die Hintertür genannt.

Sicherlich können Sie jetzt schon erraten, was man hinter dieser Hintertür alles finden, beziehungsweise damit machen kann. Das aber wird Gegenstand des zweiten Teils in der nächsten Ausgabe sein. (Logo/aw)

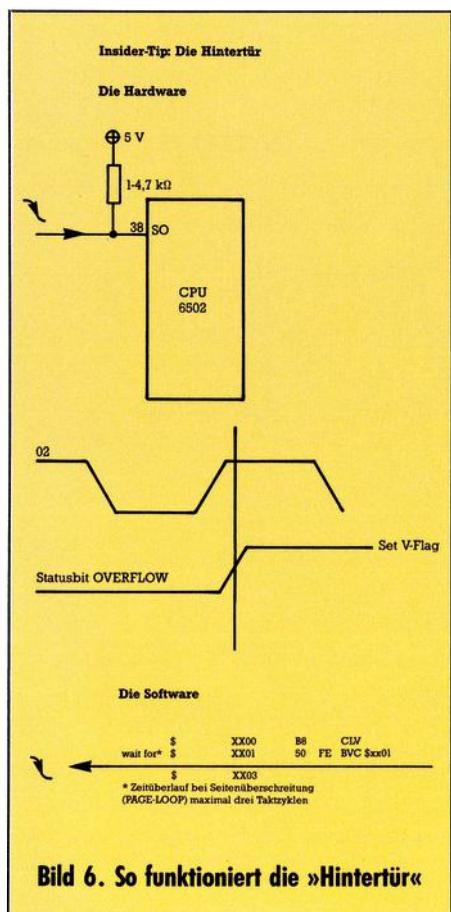


Bild 6. So funktioniert die »Hintertür«

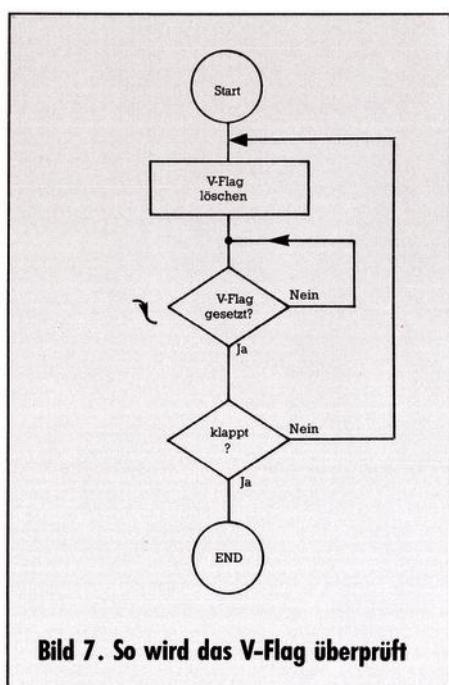


Bild 7. So wird das V-Flag überprüft