

Systembeschreibung

Auf den Seiten dieses Fachgebiets wird das System, das Computersystem oder genauer gesagt, die Hardware beschrieben. Das wird eine recht kurze Sache, denn die eigentliche Hardwarebeschreibung gehört ja bereits zum Lieferumfang der einzelnen Leiterplatten und Geräte. Hier im Lehrgang Assemblerprogrammierung wird ergänzend noch über das gesprochen, was zum Verständnis des Zusammenwirkens von Hardware und Software notwendig ist.

Die Inbetriebnahme Ihres Computers brauchen wir Ihnen nicht zu erklären. Nach dem Erscheinen des Start-Menüs auf dem Bildschirm wird das Christiani-Betriebssystem durch Eingabe der Ziffer 3 aufgerufen. Bei diesem Kaltstart werden alle erforderlichen Speicherinitialisierungen vorgenommen, der Kommando-Manager (Sie lernen ihn noch kennen) aktiviert und die folgende Systemmeldung auf dem Bildschirm veranlaßt:

Christiani

Z80

Editor

Assembler

Tester

>

Diese Meldung des Betriebssystems mit dem Promptzeichen und dem blinkenden Cursor (Schreibmarke) sagt aus, daß der Computer bereit ist und auf Ihre Anweisungen wartet. Lassen Sie ihn noch eine kurze Weile warten -es ist gut, wenn Sie vor dem praktischen Umgang mit dem Betriebssystem noch erfahren, an welcher Stelle im Speicher dieses steht und wie der Speicher im ganzen aufgeteilt ist.

Speichereinteilung

Sie wissen, daß die Befehle des Z80-Mikroprozessors aus einem Opcode und eventuell einem oder zwei Operanden bestehen. Diese Operanden können Register-, Port- oder Speicheradressen sein. Es sind die Adressen, die man bei der Programmierung geschrieben hat, man nennt sie auch logische Adressen. Da der Datenbus des Z80 16 Bit breit ist, können 65536 Speicherzellen angewählt werden, anders ausgedrückt: es steht ein Speicherraum von 64 KByte zur

**A****1**

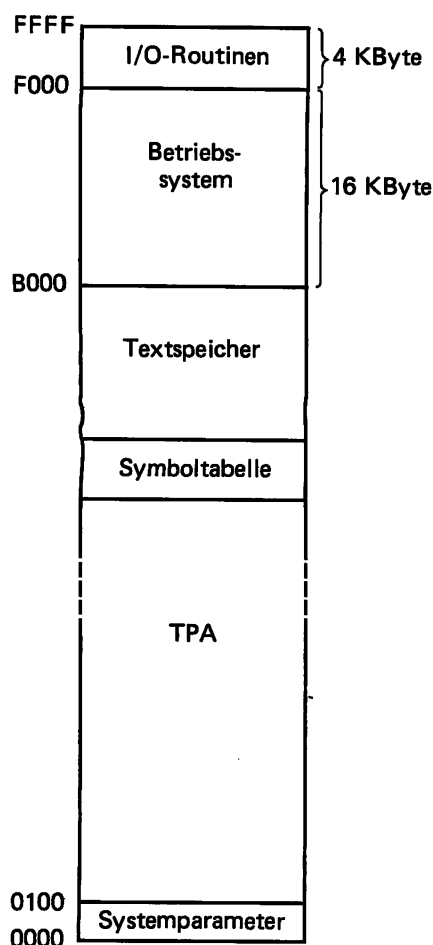


Bild A 2.1
Die schematische Darstellung der Speicherbelegung ist nicht maßstäblich.

Verfügung. Normalerweise gehört zu jeder logischen Adresse eine physikalische Adresse, eine Speicherzelle.

Das muß nicht immer so sein. Bei Ihrem Computer gibt es noch physikalischen Speicherraum außerhalb der logischen 64 KByte in Gestalt einiger EPROMs, deren Inhalt durch das Start-Menü sozusagen angeboten wird. Durch die Eingabe von z.B. der Ziffer 3 wird das Christiani-Betriebssystem in den eigentlichen 64 KByte-Arbeitsspeicher geladen. Es steht also nur dort, wenn es gebraucht wird und verspermt im anderen Fall (etwa bei der Arbeit mit dem Grafiksystem) keinen Speicherplatz.

Wie diese Speicherumschaltungen funktionieren, soll hier nicht interessieren. Aber wohin das Betriebssystem geladen wird und wie es überhaupt im Arbeitsspeicher aussieht, das müssen Sie wissen. Bild A 2.1 zeigt in schematischer Darstellung, daß sich das Betriebsprogramm ganz oben im Speicher lädt, direkt unter einen 4 KByte breiten Bereich, der für I/O-Routinen reserviert ist.

Das Betriebssystem benötigt 16 KByte Speicherraum, reicht also bis zur Adresse B000H. Ganz unten im Speicher liegt die sogenannte Zero-Page (Nullseite), die für Systemparameter reserviert ist, sie reicht bis zur Adresse 00FFH. Nach dem Einschalten des Computers oder nach einer Betätigung des RESET-Schalters wird die Adresse 0 angesprungen, von wo aus weitere Adressen in der Zero-Page die Anweisungen für die notwendigen Speicherinitialisierungen und Sprünge ins Betriebssystem enthalten.

Der dazwischen liegende Speicher von 0100H bis AFFFFH steht für Benutzerprogramme (TPA= transient programming area) und Textspeicher zur Verfügung. Beim Gebrauch des Assemblers legt dieser noch die Symboltabelle an. Der Textspeicher, mit dem Sie gleich auf den nächsten Seiten zu tun bekommen, wird direkt unterhalb des Betriebssystems angelegt.

Bei den Bereichen für Textspeicher, Symboltabelle und TPA sind in unserer Darstellung keine Adressen angeschrieben. Das geschieht aus gutem Grund: diese Adressen sind nämlich variabel. Sie werden noch erfahren, daß der Benutzer die Bereiche nach seinen speziellen Anforderungen anlegen kann.

Zusammenfassung:

Der Aufruf des Christiani-Betriebsprogramms nach dem Start-Menü erfolgt mit Eingabe der Ziffer 3. Das Betriebsprogramm wird aus einem außerhalb des 64 KBYTE-Arbeitsspeichers liegenden Speicher geladen.

Schnittstellen

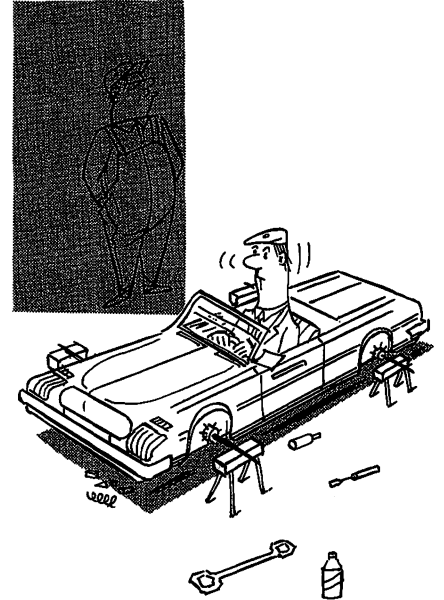
Ein Computer ohne Möglichkeiten zur Ein- und Ausgabe von Daten ist etwa so wertlos wie ein Auto ohne Räder- da nützte auch die schnellste CPU mit dem größtem Speicherraum nichts. Die notwendige Schaltung zur Ein- und Ausgabe von Daten nennt man Schnittstelle (im allgegenwärtigen englischen Sprachgebrauch Interface, was etwa das gleiche bedeutet).

Die beiden wichtigsten Schnittstellen Ihres Computers haben Sie schon ausgiebig benutzt: die Schnittstelle zur Tastatur und diejenige zum Bildschirm. Man kann in der Betrachtung noch einen Schritt weiter gehen und Tastatur sowie Bildschirm wiederum als Schnittstellen zwischen Mensch und Maschine ansehen. Diese müssen auf den Menschen zugeschnitten sein: der Bildschirm muß Licht einer Wellenlänge aussenden, das unsere Augen wahrnehmen können und die Tastatur muß so beschaffen sein, daß sie mit unseren Fingern bedient werden kann. Wir bleiben bei den ersten Schnittstellen vom Computer aus gesehen, also den Schaltungen, die den Anschluß von Tastatur und Bildschirm herstellen. Diese Schaltungen müssen natürlich auch zu ihren Partnern, den Schnittstellen von Tastatur und Bildschirm, passen.

Vor den Überlegungen über passende Schnittstellen muß jedoch geklärt sein, wie man überhaupt aus dem Computer herauskommt, wie man eine Schnittstelle so einrichtet, daß sie per Software zur Ein- oder Ausgabe von Daten benutzt werden kann. Im Prinzip ist das recht einfach: man macht bestimmte Speicherstellen von außen zugänglich. Bei einer Schnittstelle zur Ausgabe von Daten werden die Ausgänge der einzelnen Flipflops der Speicherstelle nach außen geführt. Im Programm können dann Anweisungen stehen, bestimmte Werte in diese Speicherstelle zu schreiben. Diese Werte stehen an den Ausgängen der Flipflops als 0- oder 1-Signale an und können elektrisch weiterverarbeitet werden.

Bei einer Eingabe-Schnittstelle müssen die Flipflops von außen gesetzt bzw. gelöscht werden können, so daß von außen ein bestimmter Wert gesetzt und per Programm weiterverarbeitet werden kann. Für den Prozessor macht es keinen Unterschied, ob er mit einer Speicheradresse eine "normale" oder eine nach außen verbundene Speicherstelle anspricht.

Allerdings gibt das noch keine fertige Schnittstelle, wenn die Signale der Flipflops einer Speicherstelle von außen zugänglich sind. Die Schnittstelle soll auch zu



A

3

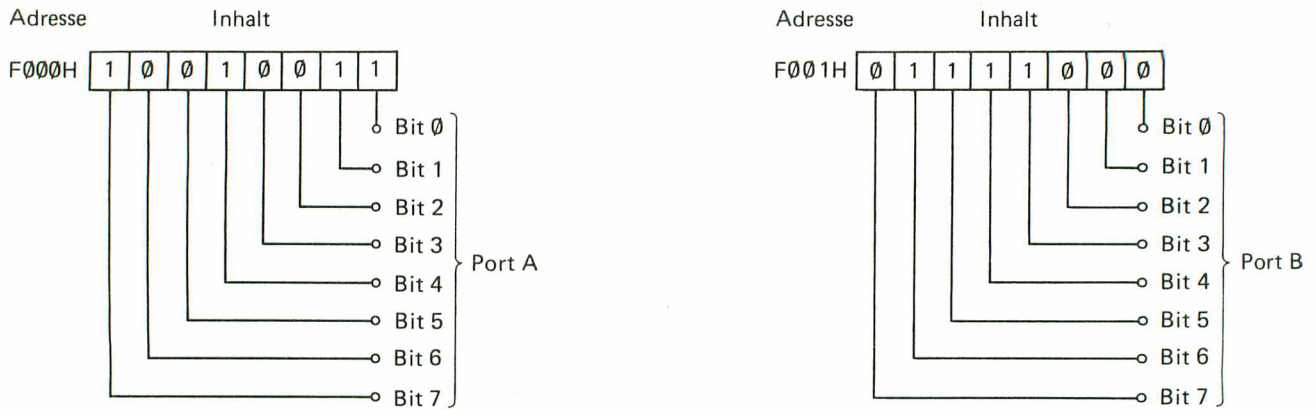


Bild A 4.1

Pro Port können 8 Daten-Bits ein- oder ausgegeben werden.

ihrem jeweiligen Gegenstück passen, es sind also noch Trennstufen und Pegelumsetzer notwendig, um die Schnittstelle komplett zu machen. Der Port als Teil der Schnittstelle ist nach dem vorher gesagten ein Baustein, den der Prozessor als Speicherstelle anspricht und dessen Inhalte von außen zugänglich sind. Bild A4.1 zeigt ein Beispiel dazu.

Sie sehen, daß Port A die Speicheradresse 0F000H und Port B die Adresse 0F001H hat. Man nennt diese Art der logischen Einbindung von Ports "Memory-Mapped I/O". Das läßt sich mit "Ein/Ausgabe im Speicherplan" übersetzen; jedem Port ist eine Adresse im Speicher zugeordnet. In die Speicheraufteilung in Bild A2.1 wären in diesem Fall noch die Adressen einzutragen, an denen sich die Ports befinden.

Der große Vorteil von Memory-Mapped I/O liegt darin, daß die Ports von der Software wie jede andere Speicherstelle angesprochen werden können. Ein Nachteil ist der große schaltungstechnische Aufwand. Um eine Portadresse zu decodieren, müssen alle 16 Adreß-Bits des Prozessors verknüpft werden.

Der Port A im Beispiel Bild A 4.1 darf nur angesprochen werden, wenn der Prozessor über den Adreßbus die Adresse F000H ausgegeben hat. Dazu werden 15 Gatter mit je 2 Eingängen gebraucht, oder, wie in Bild A 4.2 gezeigt, 5 Gatter mit je 4 Eingängen. Da der Port B einen Teil der Decodierung des Port A mitbenutzt, wird im angeführten Beispiel der Aufwand noch etwas kleiner.

Zusätzlich zur eigentlichen Decodierung ist bei jedem Portbaustein noch eine Schaltung notwendig, die den Hauptspeicher abschaltet bei denjenigen Adressen, die einen Port angehen. Eine Adresse darf ja nicht gleichzeitig eine Schreib/Lesestelle und einen Port ansprechen, es gäbe einige Kurzschlüsse auf dem Datenbus, wenn beide Bausteine gleichzeitig Daten liefern wollten.

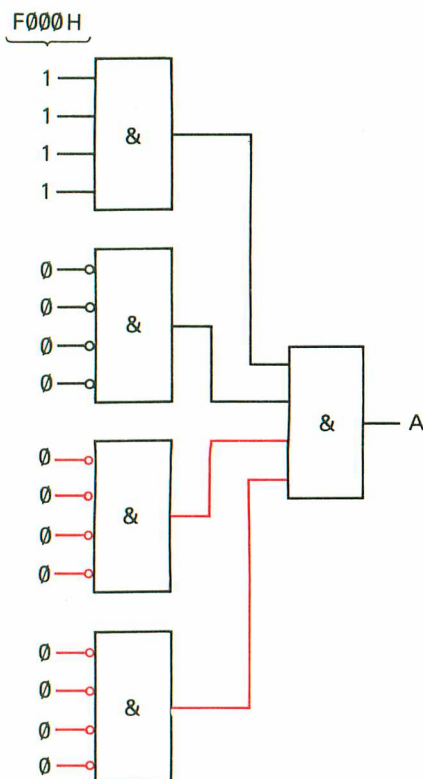


Bild A 4.2

Durch Weglassen der rot gezeichneten Leitungen verringert sich der Schaltungsaufwand, dafür wird mehr Adressenraum benötigt.

Man kann den Aufwand für die Decodierung reduzieren, indem man die Adressen nicht voll decodiert. Im Beispiel Bild A 4.2 würde der rot gezeichnete Teil einfach wegfallen. Die entsprechenden Adreßleitungen werden ignoriert, alle Adressen, bei denen die noch angeschlossenen Leitungen den richtigen Wert annehmen, sprechen den Port an, ohne Rücksicht auf die Werte der nicht angeschlossenen Leitungen. Was das bedeutet, ist klar: im gezeigten Beispiel würde der Port A durch alle Adressen von 0F00H bis 0F0FFH angesprochen, das sind 256 Adressen. Die Schaltungsvereinfachung wäre teuer erkaufte: wegen eines Ports gehen 256 Speicherplätze des Arbeitsspeichers verloren.

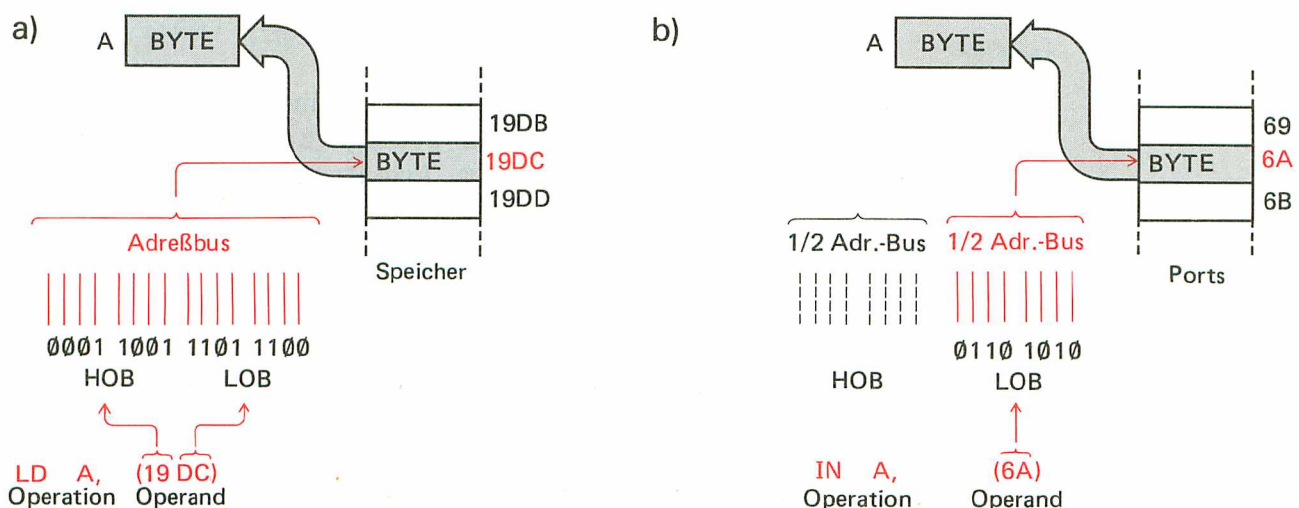
Aus diesem Grund ist man bei der Entwicklung der Prozessoren vom Typ 8080, 8085 und Z80 einen anderen Weg gegangen: man hat für die Ein- und Ausgabe einen separaten Adreßbereich von 256 Adressen vorgesehen. Für einen solchen getrennten Adreßbereich sind zwei Dinge notwendig. Erstens braucht man besondere Befehle, die ein Byte vom A-Register in eine Speicherstelle dieses gesonderten Adreßbereiches bringen und umgekehrt. Es sind dies die IN- und OUT-Befehle.

Zweitens müssen in der Hardware Wege für die Ein- und Ausgabeoperationen vorhanden sein. Dies wären ein 8 Bit breiter Datenbus und ein 8 Bit breiter Adreßbus (für 256 Adressen) zusammen mit den Schreib/Lese-Steuersignalen. Nun ist es aber so, daß eine Ein/Ausgabe-Operation nie gleichzeitig mit einer normalen Schreib/Lese-Operation vorkommt. Man kann also ohne weiteres den schon vorhandenen Datenbus und die unteren 8 Bit des Adreßbus benutzen und lediglich getrennte Steuersignale vorsehen.

Bei den Prozessoren der 80-er-Familie gibt es deshalb nur einen Datenbus und einen Adreßbus, jedoch außer den Steuersignalen, die Schreib- oder Leseoperationen kennzeichnen, zusätzlich noch solche, die angeben, ob eine

Bild A 5.1

Vergleich der Operationen eines Ladebefehls (a) und eines Portbefehls (b).



Speicherstelle des Arbeitsspeichers (LD-Befehl) oder eine der 256 Speicherstellen des Portbereichs (IN/OUT-Befehl) betroffen sind.

A**6**

In Bild A 5.1 wird die unterschiedliche Arbeitsweise des Prozessors bei der Adressierung einer Speicherstelle und der eines Ports noch einmal deutlich gemacht. Im Teilbild a) ist die Ausführung des Befehls LD A,(19DC) als Beispiel dargestellt. Es wird die im dritten und zweiten Befehlsbyte enthaltene Adresse der anzusprechenden Speicherstelle auf den Adreßbus geschaltet und anschließend das Byte aus dieser Speicherstelle über den Datenbus in das A-Register kopiert.

Im Teilbild b) sehen Sie die Funktion des Befehls IN A,(6A) mit der Adressierung eines Ports (auf den nächsten Seiten des Fachgebiets ASSEMBLER werden Sie diesen und ähnliche Befehle noch näher kennenlernen). Der Unterschied zwischen dem LD-Befehl und dem IN-Befehl besteht darin, daß bei letzterem der Operand aus einem einzigen Byte besteht, das die Port-Adresse darstellt und dessen Bits bei der Ausführung des Befehls auf die niederwertigen acht Leitungen des Adreßbus geschaltet werden. Beim LD-Befehl besteht der Operand aus 2 Bytes, deren Bits auf alle 16 Leitungen des Adreßbus geschaltet werden.

Wichtig ist, daß bei den Port-bezogenen Befehlen von der CPU ein Steuersignal an den Adreßdecodierer geschickt wird, so daß der für die Port-Adressen vorgesehene Speicherbereich angesprochen und der Arbeitsspeicher abgeschaltet wird. Auch für dieses vom Z80-Prozessor benutzte System der Ein/Ausgabe gibt es selbstverständlich einen englischen Ausdruck in der Fachsprache, es wird als "Isolated I/O" bezeichnet. Sinngemäß läßt sich das mit "getrennter Ein/Ausgabe" übersetzen.

Zusammenfassung

Beim Z80-Prozessor werden die Ports über einen getrennten Adreßbereich adressiert. Das bedingt zusätzliche Steuersignale, spart aber Platz im Arbeitsspeicher.

Frage:

1. Warum werden bei der Ausführung der Portbefehle des Z80 keine zusätzlichen Busleitungen benötigt?

(Die Antwort zu dieser Frage finden Sie auf Seite G5.)

Tastaturschnittstelle

Was ein Port ist, wissen Sie inzwischen, von Grundlagen-Lehrgängen her oder von der kurzen Darstellung in den vorangegangenen Seiten. Wie wird aber nun aus einem Port eine Schnittstelle, z.B. eine Tastaturschnittstelle?

Es soll eine Tastatur mit einem 7-Bit-Parallelausgang an ein Mikroprozessorsystem angeschlossen werden: der erste Schritt ist klar, es wird ein 8-Bit-Eingangsport mit einer bestimmten Portadresse eingerichtet. An sieben der acht Eingangsleitungen wird die Tastatur angeschlossen, damit kann ein von der Tastatur geliefertes (ASCII-) Byte jederzeit in den Computer eingelesen werden.

Soweit, so gut, aber so funktioniert die Sache noch nicht. Die Tastatur liefert nämlich immer irgendein Byte (und sei es 00) an ihrem Ausgang, auch dann, wenn gerade keine Taste gedrückt ist. Es ist also noch eine Leitung notwendig, auf der ein Signal dem Computer mitteilt, wenn eine Taste gedrückt und somit das Byte am Ausgang der Tastatur gültig ist. In der Tat gibt es diese Leitung bei jeder Tastatur mit Parallelausgang, sie wird "Strobe" genannt und liefert meist ein kurzes 0-Signal, wenn ein Datenbyte am Ausgang steht.

Der Port muß also um eine Schaltung ergänzt werden, die dafür sorgt, daß ein von der Tastatur kommendes Byte in das Portregister übernommen wird, wenn das Strobe-Signal auf Null geht. Außerdem muß durch das Strobe-Signal ein Flipflop gesetzt werden (und damit Strobe=0 festgehalten werden), dessen Ausgang z.B. auf das noch freie 8.Bit des Ports führt. Dann kann beim Einlesen des Portinhalts geprüft werden, ob das Strobe-Bit gesetzt ist. Wenn ja, dann enthält das Portregister ein gültiges Byte, das weiterverarbeitet werden kann. Danach muß das Strobe-Bit wieder zurückgesetzt werden, sonst wird bei den folgenden Abfragen immer wieder das gleiche Byte eingelesen und weiterverarbeitet.

Zu einer Schnittstelle gehört neben dem Portbaustein demnach irgendein Verfahren, das den zeitlichen Ablauf der Ein- und Ausgabeoperationen steuert. Im Fall der parallelen Tastaturschnittstelle besteht das Verfahren in der Ausgabe eines Datenbytes und eines Strobe-Signals durch die Tastatur. Letzteres zeigt an, wann ein gültiges Datenbyte ansteht. Die dazu passende Schnittstelle im Computer besteht aus ein oder zwei Ports, einigen Logikbausteinen und einer Software, die den Einlesevorgang dem Strobe-Signal entsprechend steuert. Ein solches Verfahren, das die Übergabe von Daten zwischen zwei Schaltungen

steuert, nennt man "Handshake", also Signalisation durch Händeschütteln, wobei die Hände im vorliegenden Beispiel die Strobe-Signale sind.

A**8**

Es ist einzusehen, daß jede schaltungstechnische Realisierung einer solchen Tastaturschnittstelle eine andere Software erfordert. Deshalb gehört diese Software zum Betriebssystem, das seinerseits den Anwenderprogrammen mit seinen Systemfunktionen erlaubt, die Schnittstelle zu benutzen, ohne sich um die Einzelheiten der jeweiligen Schaltung kümmern zu müssen.

Wenn in den Betriebssystemen unterschiedlicher Computer die gleichen, vereinheitlichten Systemfunktionen zur Verfügung gestellt werden, dann werden Programme sehr leicht zwischen diesen Computern austauschbar. Das gilt ebenso für Tastaturen mit gleicher Schnittstelle. Das Verfahren der Systemaufrufe in Ihrem Computer bewirkt softwaremäßig genau das gleiche wie ein normierte Tastaturschnittstelle in der Hardware. Man spricht deshalb analog zur Hardware-Schnittstelle auch von einer Software-Schnittstelle.

Die "Schnittstellen-Philosophie" bringt eine doppelte Unabhängigkeit: zum einen sind die Peripheriegeräte leicht austauschbar, wenn sie über normierte Schnittstellen angeschlossen werden. Zum anderen ist die zugehörige Software leicht übertragbar, wenn auf diese Hardware-Schnittstellen nur über eine normierte Software-Schnittstelle zugegriffen wird. Das bedeutet in der Praxis: es ist sinnvoll, z.B. die Tastaturschnittstelle nie direkt anzusprechen, sondern immer nur über die entsprechenden Systemfunktionen - auch wenn ein IN-Befehl manchmal einfacher wäre als die entsprechende Systemfunktion!

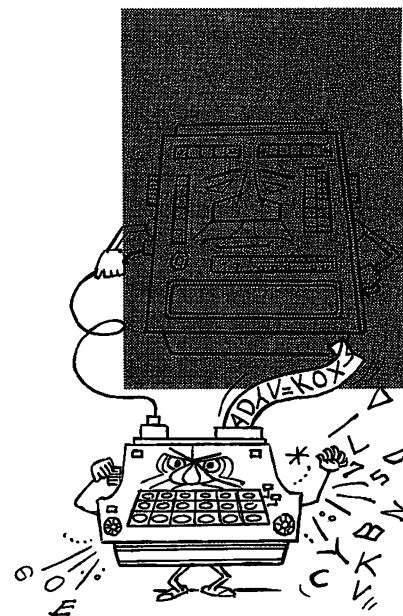
Druckerschnittstelle

Die meisten der bei Mikroprozessor-Systemen gängigen Drucker sind mit einer normierten Schnittstelle ausgestattet, die nach einem Hersteller als "Centronics-Interface" bezeichnet wird. Entsprechend geben die Computerhersteller an, daß im Computer eine "Centronics-kompatible" Schnittstelle vorhanden sei (dies ist auch bei Ihrem System der Fall). Es handelt sich dabei um eine 8-Bit-Parallelschnittstelle mit der umgekehrten Datenflußrichtung wie bei der Tastaturschnittstelle. Die acht Datenbits und das Strobe-Signal sind Ausgangsleitungen des Computers. Der entscheidende Unterschied zur Tastaturschnittstelle ist aber nicht die Richtung des Datenflusses, sondern die Notwendigkeit weiterer Steuersignale. Der Grund ist ein-

Bei der Centronics-Schnittstelle gibt es daher noch zwei weitere Signale, mit denen das Handshake vervollständigt wird: es sind die Signale "Busy" und "Acknowledge". Das Busy-Signal zeigt mit High-Pegel (1-Signal) an, daß der Drucker beschäftigt (busy) ist und daher momentan keine Daten aufnehmen kann. Ist er dann fertig mit drucken und bereit für die Aufnahme der nächsten Daten, dann geht das Busy-Signal auf Low-Pegel (0-Signal). Das Acknowledge-Signal (Bestätigungs-Signal) zeigt durch einen Impuls mit Low-Pegel an, daß der Drucker die mit Strobe übergebenen Daten verarbeitet hat und bereit ist, die nächsten Daten aufzunehmen.

Beide Signale sagen streng genommen das gleiche aus, die Schnittstelle im Computer muß also eines dieser beiden Signale auswerten, um festzustellen, ob der Drucker zur Datenaufnahme bereit ist. Wie das geschieht und welches Signal dazu benutzt wird, ist gleichgültig und für den Anwender uninteressant, da natürlich auch diese Funktion vom Betriebssystem übernommen wird.

Die Centronics-Schnittstelle des Druckers stellt übrigens noch einige zusätzliche Signale zur Verfügung, die vom Computer bei Bedarf benutzt werden können. Diese Signale sind aber beim normalen Betrieb nicht notwendig und über die Systemfunktionen des Betriebssystems nicht zugänglich. Ein Drucker mit der normierten Centronics-Schnittstelle hat meistens auch einen normierten Stecker, dessen 36 Anschlüsse mit den Signalen wie folgt belegt sind:



Anschluß-Nr.	Signal	Richtung	Beschreibung
1	<u>STROBE</u>	Eingang	Low-Impuls, wenn Daten an DATA 1 bis DATA 8 anstehen
2	DATA 1	Eingang	Daten-Eingang
3	DATA 2	Eingang	Daten-Eingang
4	DATA 3	Eingang	Daten-Eingang
5	DATA 4	Eingang	Daten-Eingang
6	DATA 5	Eingang	Daten-Eingang
7	DATA 6	Eingang	Daten-Eingang
8	DATA 7	Eingang	Daten-Eingang
9	DATA 8	Eingang	Daten-Eingang
10	<u>ACKNLG</u>	Ausgang	Acknowledge, zeigt mit Low-Impuls an, daß Daten (nach Strobe) übernommen wurden.

Anschluß- Nr.	Signal	Richtung	Beschreibung
11	BUSY	Ausgang	Busy, zeigt mit High-Pegel an, daß Drucker nicht bereit ist (Low = bereit).
12	PAPER END	Ausgang	Zeigt bei vielen Druckern mit High-Pegel Papiermangel an.
13	SELECT	Ausgang	Zeigt bei manchen Druckern ON/OFF-Line Status an.
14		Ausgang	Je nach Drucker
16	0 V	-	Signal-Masse
17	CHASSIS	-	Gehäuse-Masse
18	+5 V	-	Bei manchen Druckern mit +5 Volt verbunden.
19...30	GND	-	Abschirm-Masse für Anschluß 1 bis 12
31	INIT	Eingang	Mit Low-Pegel kann bei vielen Druckern ein Hardware-Reset ausgelöst werden.
32	<u>ERROR</u>	Ausgang	Zeigt bei vielen Druckern mit Low-Pegel Fehlerzustand an.
33	GND	-	Wie 19 bis 30.
15,34	NC	-	Nicht benutzt.
35,36		-	Je nach Drucker.

Zusammenfassung

Zu einer Schnittstelle gehört außer dem Port noch ein Verfahren, das den zeitlichen Ablauf der Ein- und Ausgabe von Daten steuert. Die dazu notwendige Hardware und Software ist so ausgelegt, daß sich normierte Schnittstellen ergeben, die über Systemfunktionen problemlos ansprechbar sind. Tastaturschnittstelle und Druckerschnittstelle sind 8-Bit-Parallel-Schnittstellen mit zusätzlichem Steuer-Signalen.

Frage:

1. Warum kann die Druckerschnittstelle nur mit dem Stro-
be-Signal nicht richtig funktionieren?

(Die Antwort zu dieser Frage finden Sie auf Seite G5.)