

Im Junior-Computer-Buch 1 wurde der Computer mit dem Menschen verglichen: Die Hardware des Menschen ist dessen Fleisch und Blut, die Software ist sein Geist. Nach Erreichen eines bestimmten Lebensalters kommt bekanntlich das körperliche Wachstum des Menschen zum Stillstand, das geistige Wachstum kennt dagegen keine Altersgrenze. Übertragen auf den Junior-Computer bedeutet dies, daß der Junior nun die Schwelle zum Erwachsenenalter erreicht hat. Seine Hardware ist mit der Interfacekarte vollständig, während die Software weiter wachsen kann. Zum Ausbau der Software für den Junior hier gleich folgende Anmerkung: Wer

ein Leitungsbündel, den Bus, in der Weise verbunden, daß jeder Anschluß einer Karte mit dem entsprechenden Anschluß auf den anderen Karten in Verbindung steht. In der Praxis werden dafür entweder einzelne Drähte oder aber eine weitere Platine, die Buskarte, verwendet. Da die Länge des Bus im Prinzip nicht begrenzt ist, läßt sich ein Bussystem auf den beabsichtigten (festen) Verwendungszweck hin maßschneidern.

Bei Einplatinensystemen wurde vom Entwickler buchstäblich alles auf eine Karte gesetzt. Hier entscheidet hauptsächlich die Bestückung und damit das äußere Format der Platine, welche

## Letzte Lötarbeiten

# Junior-Computer: Hardware vollendet

Ein Projekt geht seiner Vollendung entgegen: Nun endlich wird der Junior-Computer zum ausgereiften Senior. Die brandneue Interfacekarte verhilft dem Junior zum Eintritt in die Welt der "erwachsenen" Computer.

Die Interfacekarte ist das Bindeglied zwischen dem Prozessor und dem Elektrominial (Datensichtgerät mit ASCII-Tastatur) sowie zwischen dem Arbeitsspeicher (RAM) und zwei Kassettenrekordern. Ferner nimmt die Interfacekarte einen zusätzlichen I/O-Baustein, zwei RAM- und zwei EPROM-ICs und auch die Buspuffer auf, die für eine Erweiterung der Speicherkapazität über 8K hinaus notwendig sind.

Der Artikel gibt gleichzeitig einen Vorschau auf die Junior-Computer-Bücher 3 und 4. Um alle für den Bau wichtigen Einzelheiten ausführlich und vollständig darstellen zu können, muß sich jedoch der vorliegende Beitrag auf die Beschreibung der Hardware beschränken.

sich mit Anwenderprogrammen in Maschinensprache und mit Systemsoftware nicht beschäftigen möchte, der wird nach einer höheren Programmiersprache Ausschau halten. Der Vorzug gilt sicherlich Basic, denn diese Sprache bietet dem Anfänger viele Vorteile. Hier kann man jedoch leicht auf bereits Vorhandenes zurückgreifen und den Basic-Interpreter eines anderen 6502-Computers an den Junior anpassen. Elektor wird deshalb für den Junior diese Möglichkeit berücksichtigen und keinen eigenen Basic-Interpreter entwickeln.

Aus dem Elektor-Labor ist noch eine abrundende Ergänzung der Hardware zu erwarten; eine Speicherkarte mit dynamischen RAMs (Kapazität 16K), die eine preiswerte Alternative zu zwei 8-K-RAM/EPROM-Karten darstellt. Doch zurück zum Gegenstand dieses Artikels, der Interfacekarte für den Junior-Computer. Welche Funktionen wurden auf ihr untergebracht, wie wurden diese realisiert, und welche Eigenschaften und Möglichkeiten gewinnt der Junior-Computer hinzu? Diese und andere Fragen sollen nachfolgend ausführlich beantwortet werden.

### Bus, Karte und Buskarte

Hobbycomputer für den Selbstbau haben entweder die Gestalt eines Bussystems (z.B. SC/MP-System) oder eines Einkartensystems (Junior-Computer in seiner Grundausbaustufe). Ein Bussystem besteht aus mehreren Karten (Platinen), die meistens die gleichen äußeren Abmessungen haben. Verbreitet ist auch hier das Format 16 x 10 cm, das sogenannte Eurokartenformat. Die einzelnen Karten sind miteinander über

Möglichkeiten der Computer letztendlich bietet und wo seine Grenzen sind. Diese Grenzen lassen sich mit vertretbarem Aufwand meistens nicht weiter hinausschieben.

Einkartencomputer erfüllen ihre Aufgabe als Übungs- und Lehrcomputer sowie bei der Steuerung von einfachen Prozessen. Der Junior-Computer wurde so konzipiert, daß er in seiner Basisversion ein Einkartencomputer mit den genannten Möglichkeiten ist. Tausende von Lesern lassen ihn in dieser Gestalt arbeiten. Ihr Interesse am Computerhobby wurde häufig durch die zahllosen Veröffentlichungen und Beiträge in den Medien geweckt, die sich mit Chips, Mikros und ähnlichen Wundern moderner Technik beschäftigten. Nicht selten bleibt es aber bei einfachen oder einseitigen Anwendungen des zusammengebauten Juniors: beim einen dient er als A/D-Wandler; im anderen Fall, auch das wurde bekannt, leistet er Hilfe bei der Steuerung von Prozessen in der Halbleiterherstellung. Natürlich ist generell nichts dagegen einzuwenden, daß der Junior-Computer in seiner Grundausbaustufe bleibt und eine stets gleichbleibende Aufgabe erfüllt. Auch als Einkartencomputer bietet der Junior unzählige Möglichkeiten; sein Erweiterungskonnektor gerät deshalb manchmal in Vergessenheit.

Diese Kontaktleiste an der Schmalseite der Hauptplatine ist jedoch weniger als Verzierung, sondern vielmehr als Tor zur Realisierung von durchaus sinnvollen Erweiterungen gedacht: einer Einrichtung, die das Speichern von Programmen auf Bandkassetten ermöglicht, einem Puffer für die Datenleitungen, so daß die vom Prozessor maximal adressierbare Speicherkapazität (64K) ausgeschöpft werden kann, und einer Anschlußmöglichkeit für ein

Datensichtgerät, das den Dialog zwischen Mensch und Computer wesentlich erleichtert. Alles dies und noch einiges mehr befindet sich auf einer einzigen Karte, der Interfacekarte.

Nach Anschluß einer weiteren Platine an den Erweiterungskonnetktor gehört der Junior nicht mehr zur Gruppe der Einkartencomputer. Das bedeutet jedoch nicht, daß er nun zu einem "endlosen" Bussystem ausgebaut wird. Auch seine äußere Erscheinung spricht dagegen, ihn als Bussystem zu bezeichnen. Die Interfacekarte, auf der sich die Hardware der genannten Funktionen befindet, wird unter die Hauptplatine geschraubt. An die Interfacekarte kann eine Buskarte angeschlossen werden; die Buskarte dient ihrerseits als Träger für eine oder mehrere Speicherkarten.

Trotz der Option für eine Buskarte ist der erweiterte Junior-Computer kein Bussystem im üblichen Sinn. Im Hinblick auf die Sandwich-Konstruktion von Basiskarte und Interfacekarte kann man ihn eher als Doppel-Einkartencomputer bezeichnen. Wichtiger als die Klärung der Familienzugehörigkeit erscheint an dieser Stelle jedoch die Beschreibung dessen, was den Junior-Computer zum Doppel-Einkartencomputer macht.

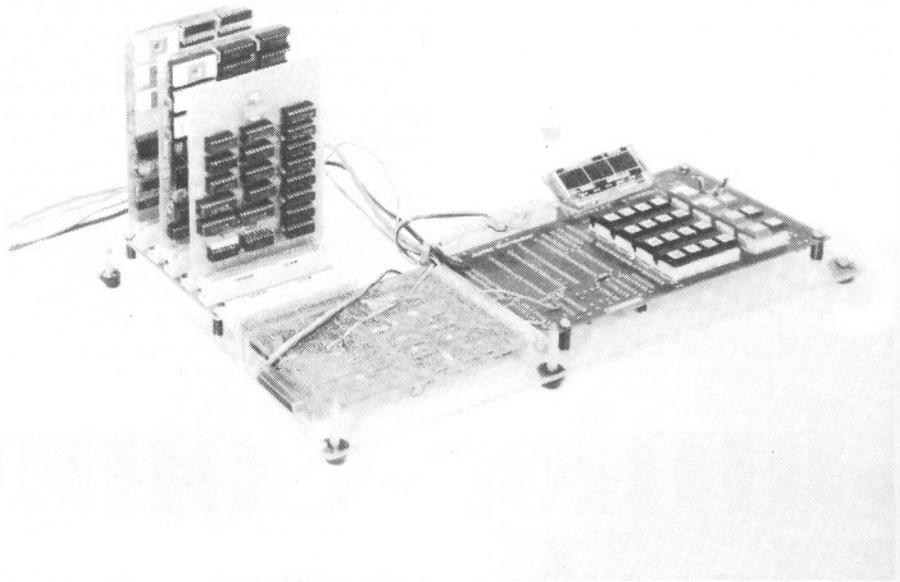
## Die Interfacekarte

Interface bedeutet so viel wie Kopplung, Anpassung oder Bindeglied. Die Interfacekarte des Junior-Computers stellt die Verbindungen zu zwei Kassettenspeichern und zu weiteren peripheren Geräten entsprechend der Norm RS-232 her; sie enthält ferner einen Datenbuspuffer für die Basiskarte, der als internes Interface betrachtet werden kann. Die außerdem noch vorhandenen Speicherbausteine ergänzen die Speicherkapazität der Hauptplatine.

Die Bilder 1 und 2 zeigen die Schaltung der Interfacekarte. Obwohl die Elektronik im Vergleich zur Hauptplatine des Juniors umfangreicher ist, haben beide Platinen die gleichen Abmessungen. Es war nicht ganz einfach, dieses Ziel zu erreichen, aber schließlich hatte die Mühe des Platinenlayouts doch Erfolg. Nachfolgend sollen die Schaltungsdetails der Interfacekarte der Reihe nach besprochen werden.

### Stoßstellen: Die Datenbuspuffer

In Bild 1 sind links die Anschlüsse des Eingangskonnettors (INPUT CONNECTOR) gezeichnet. Über die Stifte dieser Kontakte sowie über fünf weitere Verbindungen mit dem PORT CONNECTOR (siehe Bild 2) verläuft die Kommunikation mit der Basisplatine des Junior-Computers. Bis auf die Leitungen EX und K1...K7 (diese enden auf der Interfacekarte) finden sich sämtliche Leitungen auf dem Ausgangskonnetktor (OUTPUT CONNECTOR) wieder, der in Bild 1 rechts oben dargestellt ist. Der Ausgangskonnetktor verbindet die Interfacekarte mit der Buskarte, die



ihreits die Verbindungen zu einer oder mehreren Speicherkarten herstellt. Die Adreßleitungen A0...A15 und die Datenleitungen D0...D7 des Eingangskonnettors erscheinen auf dem Ausgangskonnettor als AB0...AB15 und DB0...DB7. Der Buchstabe B steht für Buffer (Puffer); die Adreß- und Datenleitungen werden nämlich auf der Interfacekarte gepuffert. Das geschieht aus zwei Gründen: Zum einen verhindert die Pufferung ein Überschreiten der zulässigen Last an den einzelnen Leitungen, die beim Anschluß vieler Bausteine an den Daten- und Adreßbus auftreten kann. Zum anderen läßt sich der Datenfluß zwischen der Basisplatine und dem restlichen Teil des Computers (z.B. den zusätzlichen Speicherplatinen) mit Hilfe der Puffer blockieren. Es wird später noch deutlich werden, wann dies notwendig ist.

Die Puffer der Adreßleitungen sind mit N1...N16 bezeichnet. Da die Adressierung stets vom Prozessor ausgeht, reichen hier unidirektionale, also stets in gleicher Richtung arbeitende Puffer aus. Anders verhält sich dies beim Datenbus. Für jede der acht Datenleitungen sind zwei parallelgeschaltete, jedoch entgegengesetzt gerichtete Puffer vorhanden. Die Gruppe der Datenbuspuffer kann abhängig von den Erfordernissen drei verschiedene Zustände annehmen:

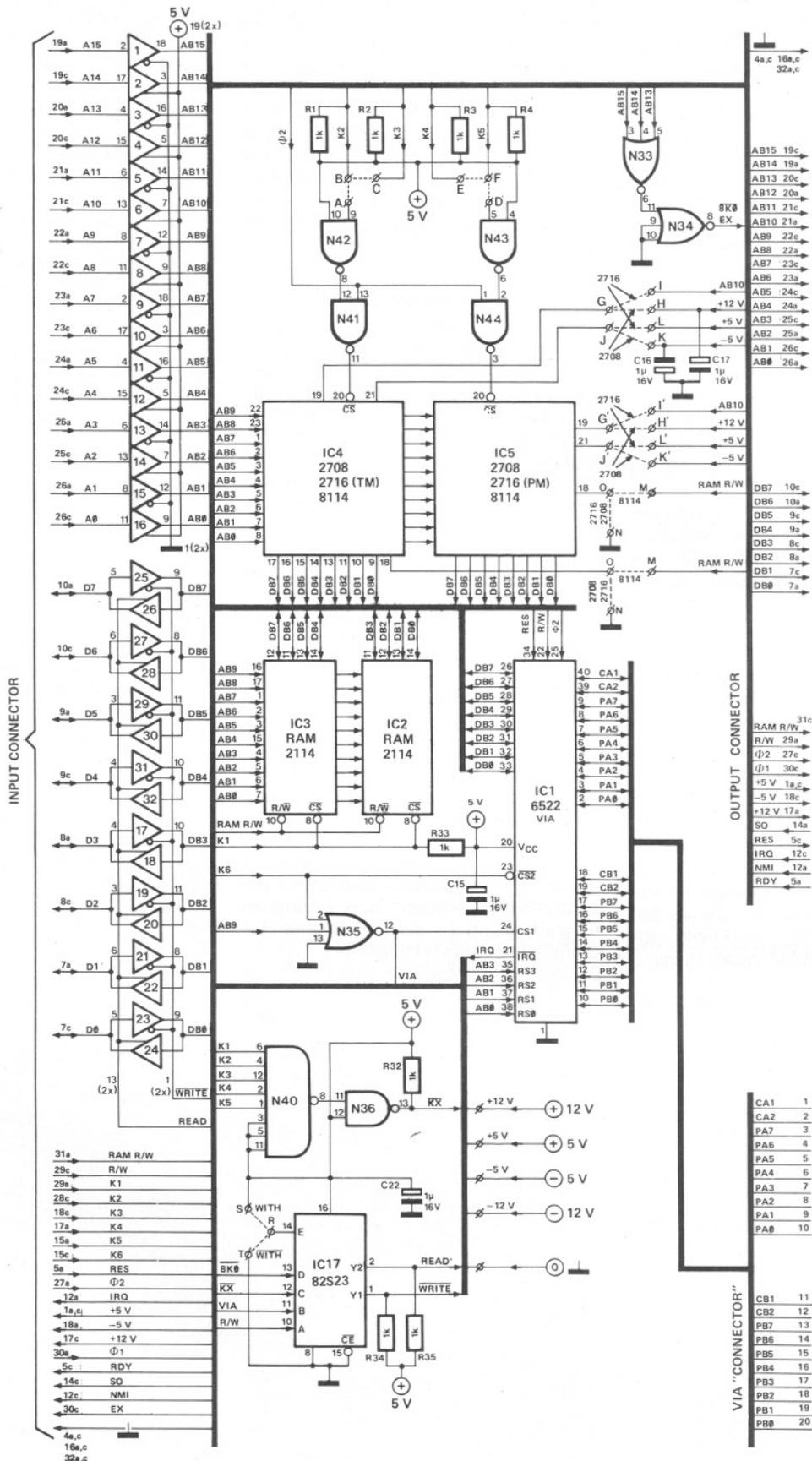
1. Die Puffer mit ungerader Numerierung (N17, N19 usw. ... N31) sind aktiv, während die übrigen Puffer inaktiv sind. Dieser Fall tritt dann ein, wenn die Leitungen WRITE und READ auf logisch 0 gehen. Der Datenfluß ist jetzt von der Basiskarte nach außen zu den übrigen Funktionseinheiten des Computers gerichtet.
2. Die Puffer mit ungerader Numerierung sind inaktiv (Tri-State-Zustand; die Ausgänge sind hochohmig), während die Puffer mit gerader Numerierung (N18, N20 usw. ... N32) die aktive

Rolle übernehmen. Dazu müssen die Leitungen WRITE und READ logisch 1 sein. In diesem Zustand können Daten von außen (z.B. von der Interfacekarte) zur Basiskarte fließen.

3. Sämtliche Puffer sind inaktiv, wenn die WRITE-Leitung logisch 1 und die READ-Leitung logisch 0 ist. Ein Austausch von Daten mit der Basisplatine kann nicht stattfinden. Dieser Zustand ist immer dann notwendig, wenn Daten nur intern zwischen den auf der Basisplatine selbst befindlichen Bausteinen zu transportieren sind. Die dritte Möglichkeit, bei der alle Datenbuspuffer gleichzeitig aktiv sind, ist nicht sinnvoll; von ihr wird deshalb kein Gebrauch gemacht.

### Zweiter I/O-Baustein

Das Innenleben von IC1 in Bild 1, einem VIA (Versatile Interface Adapter) vom Typ 6522, ist ein Kapitel für sich; ihm wurde deshalb ein ganzes Kapitel im zweiten Teil des Junior-Computer-Buchs gewidmet. Hier soll nur angemerkt werden, daß dieser Baustein den PIA (Peripheral Interface Adapter) des Typs 6532 auf der Basisplatine an Vielseitigkeit noch erheblich übertrifft. Die wichtigen Verbindungen mit der Außenwelt sind auf den VIA-"CONNECTOR" gelegt (siehe Bild 1). Da dieser Konnetktor kein echter Konnetktor ist, wurde "CONNECTOR" in Anführungszeichen gesetzt. Der 6522 ist mit den Adreßleitungen AB0...AB3 und mit verschiedenen Leitungen des Steuerbus verbunden. Natürlich sind am 6522 auch die acht Leitungen des Datenbus sowie die IRQ-Leitung angeschlossen. Über letztere wird ein Interrupt ausgelöst, wenn zum Beispiel ein im 6522 integrierter Timer den "Time Out" erreicht. Ebenso wie der 6532 benötigt auch der 6522 zwei "Chip select"-Signale. Die Eingänge CS2 beider Bausteine liegen gemeinsam an Ausgang K6 des Adressendekoders IC6, der sich auf der Basis-



81033-1

Bild 1. Der Junior wird erwachsen: Auf der Interfacekarte sind unter anderem der Datenbuspuffer, zusätzliche Speicher-ICs und ein weiterer I/O-Baustein untergebracht.

platine befindet. Der Eingang CS1 des 6522 ist dagegen mit dem Ausgang von NOR-Gatter N35 verbunden. Den Eingängen von N35 werden die Signale K6 und AB9 zugeführt. Der 6522 ist nur dann aktiv, wenn  $\overline{CS2}$  logisch 0 und gleichzeitig CS1 logisch 1 ist. Letzteres bedeutet, daß AB9 und K6 logisch 0 sein müssen (der 6532 ist dagegen bei A9 = "1" aktiv). Berücksichtigt man ferner, daß Leitung A8 (bzw. AB8) weder mit dem 6522 noch mit dem 6532 verbunden ist, dann ergibt sich für den VIA (6522) auf der Interfacekarte folgender Adressbereich:

1800 = 1900 ... 18FF = 19FF

(AB8 = X; AB9 = 0; K6 = 0)

Der PIA (6532) auf der Basisplatine hat dagegen den Adressbereich

1A00 = 1800 ... 1AFF = 18FF

(A8 = X; A9 = 1; K6 = 0).

Läßt man die Doppeladressierung durch A9 = X außer Betracht, so sind sowohl für den PIA als auch für den VIA jeweils 256 Adressen verfügbar. Doch ebenso wie beim PIA (6532), der 19 verschiedene, mehrfach adressierbare "Speicherplätze" besitzt, sind auch beim VIA nicht sämtliche 256 Adressen belegt. Wie aus Bild 1 hervorgeht, stehen die Adreßleitungen AB4 ... AB7 mit dem VIA nicht in Verbindung; beim VIA können somit nur 16 Adressen angesteuert werden.

#### Mehr Speicherkapazität

Auf der Interfacekarte ist so viel Platz für zusätzliche Speicher-ICs reserviert (maximal 5K, selektiert durch K1 ... K5), daß der Standard-Adressbereich von 8K ausgefüllt ist. Die RAMs IC2 und IC3 (2 x 2114) haben zusammen eine Kapazität von 1K. Sie werden durch  $\overline{CS} = K1 = "0"$  selektiert; die Adressen dieses 1-K-RAM lauten daher: 0400 ... 07FF.

Die hier gespeicherten Seiten 04 ... 07 schließen unmittelbar an die Seiten 00 ... 03 der Basisversion an; längere Anwenderprogramme können daher zusammenhängend gespeichert werden.

Für IC4 und IC5 lassen sich wahlweise 1-K-RAMs (8114), 1-K-EPROMs (2708) oder 2-K-EPROMs (2716) einsetzen. Diese Chips werden über die Gatter N42 ... N44 von den Signalen K2 ... K5 sowie vom Taktsignal  $\Phi 2$  aktiviert. Das Taktsignal  $\Phi 2$  sorgt für das richtige Timing beim Lesen und, sofern erforderlich, beim Schreiben in bzw. aus diesen Speicher-ICs.

Die Adressbereiche von IC4 und IC5 sind wie folgt festgelegt:

K2 = 0 Adressen 0800 ... 0800 0,5 bzw. 1 x IC4

K3 = 0 Adressen 0C00 ... 0CFF 0,5 bzw. 1 x IC4

K4 = 0 Adressen 1000 ... 13FF 0,5 bzw. 1 x IC5

K5 = 0 Adressen 1400 ... 17FF 0,5 bzw. 1 x IC5

Setzt man für IC4 und IC5 1-K-RAMs oder 1-K-EPROMs ein, dann gibt es für beide ICs jeweils zwei identische Adressbereiche. Bei den 2-K-EPROMs (2716) werden dagegen jeweils zwei K-Signale kombiniert: K2 & K3 für IC4 (Adressbereich 0800 ... 0FFF) und K4 & K5 für IC5 (Adressbereich

1000 ... 17FF). Um die Signale K2 ... K5 in Abhängigkeit von der gewählten IC-Bestückung an die Gatter N42 und N43 legen zu können, sind auf der Platine die Lötunkte A ... F angebracht. Dort müssen Drahtbrücken eingelötet werden, deren Lage sich aus der Schaltung in Bild 1 ergibt. Dies gilt auch für die Lötpunkte G ... O und G' ... O'; die Lage der Brücken hängt hier von den erforderlichen Speisespannungen und davon ab, ob zusätzlich die Adreßleitung AB10 oder die RAM-R/W-Leitung benötigt wird.

#### Vollständige Adressendekodierung

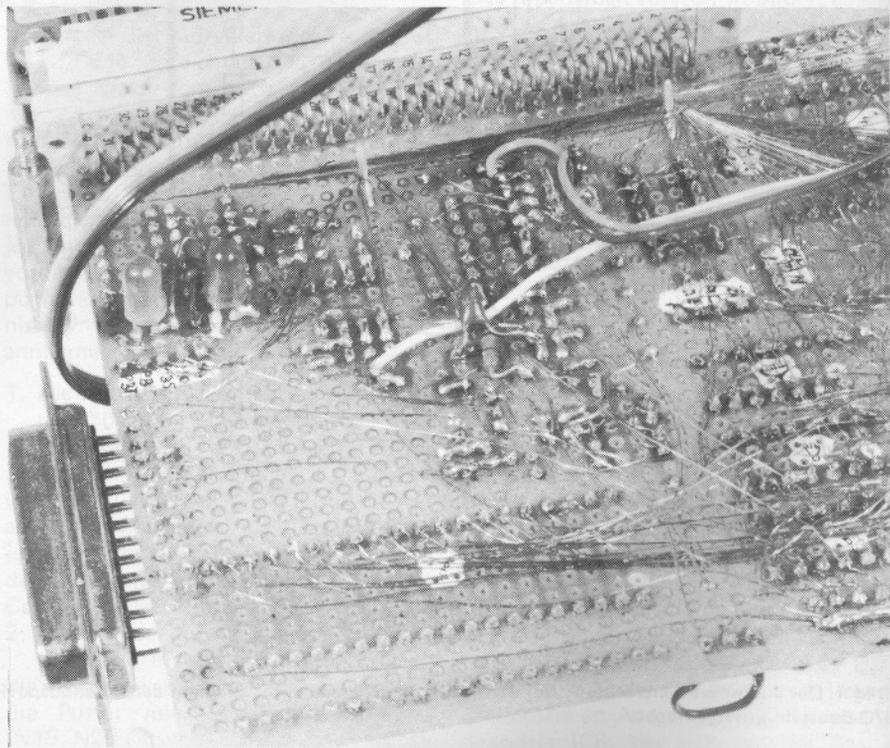
Beim Junior-Computer in der Basisversion werden nur 8K des insgesamt 64K umfassenden, vom Prozessor adressierbaren Bereichs dekodiert; die Leitungen A13, A14 und A15 des Adreßbus enden nämlich auf dem Expansionskonnektor. Jede Adresse ist bei der Basisversion mit sieben weiteren Adressen identisch: Der Inhalt der Seiten 0X mit X = 0 ... F wiederholt sich auf den Seiten 2X, 4X, 6X, 8X, AX, CX und EX, während der Inhalt der Seiten 1X mit X = 0 ... F auch auf den Seiten 3X, 5X, 7X, 9X, BX, DX und FX erscheint. Eine Speichererweiterung um mehr als 5K erfordert eine vollständige Adressendekodierung, da dann auch der Bereich 2000 ... FFFF zumindest teilweise belegt werden muß. Beschränkt man sich dagegen auf eine Erweiterung um maximal 5K (dafür ist unmittelbar auf der Interfacekarte Platz vorhanden), so ist eine vollständige Adressendekodierung nicht unbedingt notwendig. Erst wenn auf die Interfacekarte zusätzliche Speicherkarten gesteckt werden, ist der Zeitpunkt gekommen, zum Lötcolben zu greifen und an der Basisplatine eine kleine Änderung vorzunehmen.

Die Adressendekodierung wird nämlich

wie folgt erweitert: Als erstes ist Eingang D des Adressendekoders IC6 auf der Basisplatine von Masse abzulöten und an Punkt EX zu legen. Eingang D von IC6 liegt dann am Ausgang von Inverter N34; dieser wird von NOR-Gatter N33 gesteuert, dessen Eingänge mit den Leitungen AB13, AB14 und AB15 des Adreßbus verbunden sind. Sobald eine dieser drei Adreßleitungen logisch 1 ist, wird auch die Leitung EX und damit der Eingang D von IC1 logisch 1. Das hat zur Folge, daß alle Ausgänge (K0 ... K7) auf logisch 1 gehen. Sind dagegen alle drei Adreßleitungen AB13 ... AB15 logisch 0, dann ist der Eingang D von IC6 logisch 0, was gleichbedeutend mit der Situation vor Umlöten der Brücke ist. Nur in diesem Fall kann eine der Leitungen K0 ... K7 logisch 0 werden und einen der hier angeschlossenen Bausteine aktivieren.

Ein zu adressierender Speicher- oder I/O-Baustein, der sich auf der Basisplatine oder der Interfacekarte befindet, ist nach Umlöten der Drahtbrücke nur noch auf den Seiten 0X und 1X mit X = 0 ... F erreichbar, während die Seiten 2X ... FX den auf der Buskarte aufgesteckten Karten zugeteilt sind. Um diesen Bereich (Adressen 2000 ... FFFF) zu erschließen, genügt natürlich das Umlöten der Drahtbrücke von Masse nach EX allein nicht, sondern es sind zusätzliche Adressendekoder erforderlich.

Das Signal EX gelangt unter dem Namen 8K0 auch zu einem Adresseneingang des PROM IC17. In diesem PROM sind 32 Byte gespeichert; von jedem der aus 8 bit bestehenden 32 Byte sind jedoch nur die Bits Y1 (= WRITE) und Y2 (= READ) von Bedeutung. Dies sind die Signale, die die Datenbusbuffer N17 ... N32 steuern. Die Adressierung der 32 Byte erfolgt über die PROM-



Eingänge A...E; die hier anliegenden Signale werden später noch ausführlich besprochen.

Warum ist hier ein PROM notwendig? Schließlich muß dieses programmiert werden, und außerdem könnte man meinen, daß sich die Signale READ und WRITE für die Datenbusbuffer einfach aus dem Signal R/W des Prozessors ableiten lassen. Weshalb also umständlich, wenn es anscheinend auch einfach geht?

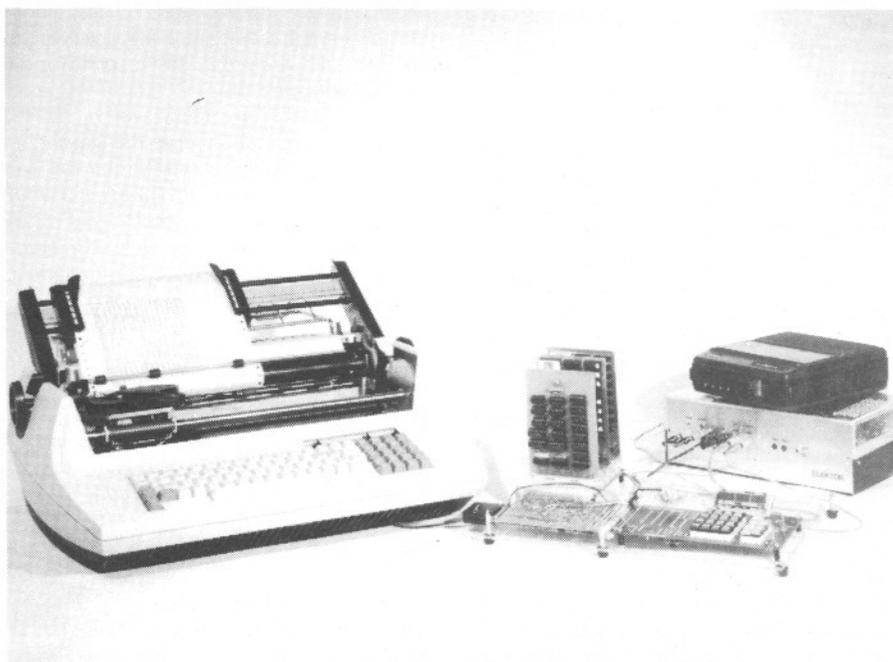
Aus zwei Gründen. Der erste ist folgender: Gesetzt den Fall, daß Daten aus einem der Bausteine gelesen werden sollen, die sich auf der Basiskarte des Junior-Computers befinden und deshalb am ungepufferten Datenbus liegen. Mit ihm sind auch die Ausgänge der Datenbuspuffer N18, N20 usw. ... N32 verbunden. Wenn das R/W-Signal des Prozessors die Puffer direkt steuern würde, wären diese Puffer jetzt aktiv. Der Datenbus würde von ihren Ausgangssignalen "beherrscht", so daß eine Verfälschung der zu lesenden Daten unvermeidlich wäre. Auf dem Datenbus würden nämlich infolge der niederohmigen Ausgänge der genannten Puffer nicht die gewünschten Daten, sondern irgendwelche anderen Signale erscheinen. Um dies zu verhindern, müssen im beschriebenen Fall sämtliche 16 Datenbuspuffer blockiert werden, was einer Abtrennung aller (vom Prozessor aus gesehen) hinter den Puffern liegenden Bausteine gleichkommt. Beim Schreiben in Bausteine der Basiskarte sind die Dinge dagegen anders gelagert: Hierbei treten derartige Probleme nicht auf, da die nicht adressierten Bausteine die an ihren Dateneingängen ankommenden Signale ignorieren.

Noch ein zweiter Grund macht das PROM IC17 notwendig: Die Vektoren NMI, RES und IRQ können bei vollständiger Adressendekodierung unter den echten Adressen FFFA...FFFF anstelle von 1FFFA...1FFF des Basisversion-EPROMs abgelegt werden. Möglich ist dies natürlich nur, wenn Seite FF tatsächlich zu einem (E)PROM gehört. Ist Seite FF nicht mit einem (E)PROM belegt, so müssen die Vektoren NMI, RES und IRQ wie bisher aus dem EPROM der Basiskarte geholt werden. Das aber kann nur geschehen, wenn alle 16 Datenbuspuffer inaktiv sind, so daß die auf der Basiskarte nicht angeschlossenen Leitungen A13, A14 und A15 des Adreßbus keinen Einfluß haben. Auf der anderen Seite ist es zwingend notwendig, Seite FF mit einem diese Vektoren enthaltenden (E)PROM zu belegen, sobald eine oder mehrere zusätzliche Speicherkarten über die Buskarte auf die Interfacekarte gesteckt werden. Die Vektoren können dann nämlich nicht mehr ersatzweise aus der Seite 1F gelesen werden, weil die Abtrennung der Buskarte durch die Puffer den ungehinderten Zugang zu den Speicherkarten verhindern würde. Zurück zur Adressierung des PROM IC17. Folgende fünf Signale liegen an

Tabelle 1

PROM-Adresse (hex)	E = WITH oder WITH	D = 8K0	C = KX	B = VIA	A = R/W	Y8	Y7	Y6	Y5	Y4	Y3	Y2 = READ	Y1 = WRITE	PROM-Daten (hex)
00	0	0	0	0	0	0	0	0	0	0	0	0	0	00
01	0	0	0	0	1	0	0	0	0	0	0	1	1	03
02	0	0	0	1	0	0	0	0	0	0	0	0	0	00
03	0	0	0	1	1	0	0	0	0	0	0	0	0	00
04	0	0	1	0	0	0	0	0	0	0	0	0	1	01
05	0	0	1	0	1	0	0	0	0	0	0	0	1	01
06	0	0	1	1	0	0	0	0	0	0	0	0	0	00
07	0	0	1	1	1	0	0	0	0	0	0	1	1	03
08	0	1	0	0	0	0	0	0	0	0	0	0	0	00
09	0	1	0	0	1	0	0	0	0	0	0	0	0	00
0A	0	1	0	1	0	0	0	0	0	0	0	0	0	00
0B	0	1	0	1	1	0	0	0	0	0	0	0	0	00
0C	0	1	1	0	0	0	0	0	0	0	0	0	1	01
0D	0	1	1	0	1	0	0	0	0	0	0	0	1	01
0E	0	1	1	1	0	0	0	0	0	0	0	0	0	00
0F	0	1	1	1	1	0	0	0	0	0	0	0	0	00
10	1	0	0	0	0	0	0	0	0	0	0	0	0	00
11	1	0	0	0	1	0	0	0	0	0	0	1	1	03
12	1	0	0	1	0	0	0	0	0	0	0	0	0	00
13	1	0	0	1	1	0	0	0	0	0	0	0	0	00
14	1	0	1	0	0	0	0	0	0	0	0	0	1	01
15	1	0	1	0	1	0	0	0	0	0	0	0	1	01
16	1	0	1	1	0	0	0	0	0	0	0	0	0	00
17	1	0	1	1	1	0	0	0	0	0	0	1	1	03
18	1	1	0	0	0	0	0	0	0	0	0	0	0	00
19	1	1	0	0	1	0	0	0	0	0	0	0	0	00
1A	1	1	0	1	0	0	0	0	0	0	0	0	0	00
1B	1	1	0	1	1	0	0	0	0	0	0	0	0	00
1C	1	1	1	0	0	0	0	0	0	0	0	0	0	00
1D	1	1	1	0	1	0	0	0	0	0	0	1	1	03
1E	1	1	1	1	0	0	0	0	0	0	0	0	0	00
1F	1	1	1	1	1	0	0	0	0	0	0	0	0	00

Tabelle 1. Inhalt des PROM IC17. Die Programmierung kann mit dem PROM-Programmierer aus Elektor Juli/August '80 (Schaltung 95) vorgenommen werden. Pinkompatible PROMs (z.B. 74S188) lassen sich zwar in der Schaltung verwenden, sie sind jedoch nicht auf die gleiche Weise programmierbar.



seinen Adreßeingängen und nehmen Einfluß auf die Steuersignale WRITE und READ für die Datenbuspuffer:

A. Das Signal R/W oder genauer  $R/\bar{W}$  (der Invertierungsstrich über dem W wurde bisher meistens weggelassen; er bedeutet: aktiver Zustand bei logisch 0). Dieses Signal gibt an, ob eine Lese- oder eine Schreiboperation ausgeführt werden soll.

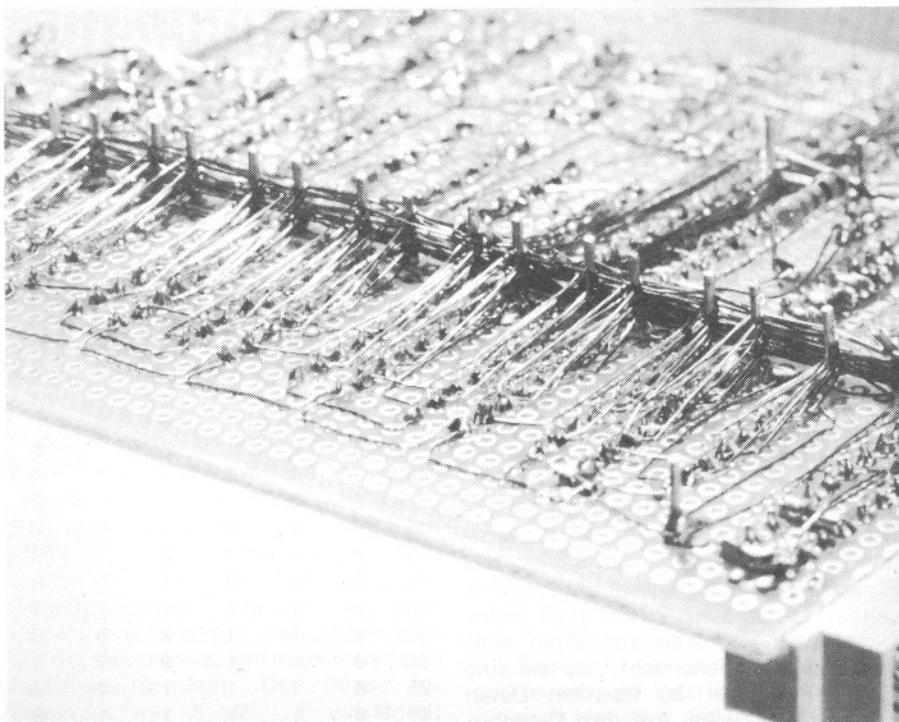
B. Das Signal VIA, das von N35 kommt und gleichzeitig als CS1-Signal für den VIA IC1 dient. Es ist logisch 1, wenn K6 und AB9 logisch 0 sind, also wenn der VIA aktiviert ist. Bei der Adressierung des 1-K-Bereichs, bei dem  $K6 = 0$  ist, liegt der PIA vor und der VIA hinter den Datenbuspuffern.

C. Das Signal  $\bar{KX}$ . Dieses liefert das als Inverter arbeitende Gatter N36, das seinerseits von NAND-Gatter N40 gesteuert wird.  $\bar{KX}$  ist logisch 0, wenn eins der Signale K1...K5 logisch 0 ist. Das aber ist bei der Adressierung eines Bausteins auf der Interfacekarte der Fall.

D. Das schon besprochene Signal  $\bar{8K0} = EX$ . Es ist logisch 0, wenn der erste 8-K-Bereich (Speicher und I/O auf der Basiskarte und der Interfacekarte) adressiert wird, und logisch 1, wenn Speicherbausteine auf der Buskarte adressiert werden.

E. Eingang E von IC17 ist entweder mit der Speisespannung (Drahtbrücke R-S) oder mit Masse (Brücke R-T) verbunden. Dieser Eingang muß logisch 0 sein, wenn keine Buskartenspeicher vorhanden sind (WITH = WITHOUT = ohne); im anderen Fall liegt er an +5 V. In Tabelle 1 ist der Inhalt von PROM IC17 angegeben. Die Bits Y3...Y8 sind ständig logisch 0; sie spielen hier keine Rolle. Von den 32 möglichen Kombinationen der Eingangssignale A...E werden nur acht benutzt, da die Datenbuspuffer in Abhängigkeit von acht unterschiedlichen Situationen gesteuert werden müssen:

1. Schreiben oder Lesen in bzw. aus Bausteinen auf der Interfacekarte (IC2...IC5). Es sind die acht Datenbuspuffer aktiviert, in deren Richtung gelesen bzw. geschrieben werden soll (PROM-Adressen 00 und 01).
2. Lesen aus dem EPROM und Lesen oder Schreiben aus bzw. in das RAM oder den PIA auf der Basiskarte. Sämtliche Datenbuspuffer sind blockiert (PROM-Adressen 04 und 05).
3. Schreiben oder Lesen des VIA. Er befindet sich auf der Interfacekarte, so daß die acht Datenbuspuffer aktiviert werden müssen, die in die Schreib- bzw. in die Leserichtung zeigen (PROM-Adressen 06 und 07).
4. Schreiben oder Lesen von Adressen, die bei vollständiger Adressendekodierung auf den Buspufferkarten liegen würden; Buspufferkarten sind jedoch (wie bei 1. bis 3.) nicht vorhanden. Eingang E von IC17 wurde wie beschrieben an logisch 0 gelegt. Die in Leserichtung zeigenden Datenpuffer werden blockiert; die Vektoren NMI, RES und IRQ werden automatisch aus



dem EPROM auf der Basisplatte geholt (PROM-Adressen 0C und 0D).

5. Wie 1., jedoch Buspufferkarten vorhanden (PROM-Adressen 10 und 11).
6. Wie 2., jedoch Buspufferkarten vorhanden (PROM-Adressen 14 und 15).
7. Wie 3., jedoch Buspufferkarten vorhanden (PROM-Adressen 16 und 17).
8. Wie 4., jedoch Buspufferkarten vorhanden. In diesem Fall (Eingang E von PROM IC17 liegt an logisch 1) müssen die betreffenden Datenbuspuffer beim Lesen aktiviert werden. Die Vektoren NMI, RES und IRQ werden wie beschrieben aus Seite FF geholt; diese Seite muß deshalb mit einem (E)PROM belegt sein, das unter den Adressen FFFA...FFFF diese drei Vektoren enthält. Von den 32 in PROM IC17 gespeicherten Bytes wird nur die Hälfte verwertet, da die zur anderen Hälfte gehörenden Kombinationen von  $D = \bar{8K0}$ ,  $C = \bar{KX}$  und  $B = VIA$  normalerweise nicht auftreten. Doch auch wenn das geschehen würde, bliebe es unschädlich: Da hier ausschließlich logische Nullen im PROM stehen, zeigen in einem solchen Fall die aktiven Datenbuspuffer in Schreibrichtung; eine Verfälschung der Datenbuspuffer ist somit ausgeschlossen.

### Kassetten-Hardware

Der größte Teil der Schaltung in Bild 2 gehört zum Kassetteninterface, das zusammen mit der zugehörigen Software das Speichern von Daten auf normalen Tonbandkassetten ermöglicht. Dazu müssen die zu speichernden Daten vorher in Töne umgesetzt werden; umgekehrt ist beim Lesen der Daten die Rückwandlung der aufgezeichneten

Töne in die vom Computer lesbare Datenform notwendig. Wie dies im einzelnen geschieht, soll nun betrachtet werden.

Als Datenleitung zwischen Prozessor und Kassetteninterface wird die Leitung PB7 des Portkonnektors benutzt. Während der Aufzeichnung von Daten auf Band ist PB7 ebenso wie PB5 als Ausgang programmiert. Bei dieser Operation liegt PB5 ständig auf "0" und PB6 ständig auf "1". Die "0" am linken Eingang (Pin 8) von Gatter N38 hat eine "1" an seinem Ausgang zur Folge; dieser Ausgang mit offenem Kollektor ist dann hochohmig. Infolge der "0" an Leitung PB5 leitet der PNP-Darlingtontransistor T3, so daß der Kontakt von Relais Re2 geschlossen ist und gleichzeitig die rote LED aufleuchtet. Der Relaiskontakt verbindet die beiden Anschlüsse von Buchse J4 und schaltet über den Fernbedienungsanschluß den Motor des mit OUTPUT bezeichneten Kassettenrekorders ein. Vorausgesetzt, daß bei diesem Rekorder die Aufnahmetaste gedrückt ist, läßt sich so der Aufnahmevorgang durch Software starten und stoppen.

Die während des Schreibens von Daten auf Band an PB6 liegende "1" sorgt dafür, daß T2 sperrt und dadurch Relais Re1 sowie die grüne LED stromlos sind. Der Motor des Kassettenrekorders INPUT ist daher ausgeschaltet.

Über Leitung PB7, Inverter N39 und Buchse J2 gelangen die bereits mit Hilfe von Software in Töne umgewandelten Datenbits zum Eingang des Kassettenrekorders OUTPUT. Sie passieren dabei das Trimpoti P2, mit dem sich die Höhe der tonfrequenten Eingangsspannung einstellen läßt.

Sollen Daten von einer Kassette gelesen werden, dann wird PB7 als Eingang programmiert, PB5 und PB6 bleiben dagegen als Ausgang geschaltet. Jetzt ist jedoch PB5 logisch 1 und PB6 logisch 0,

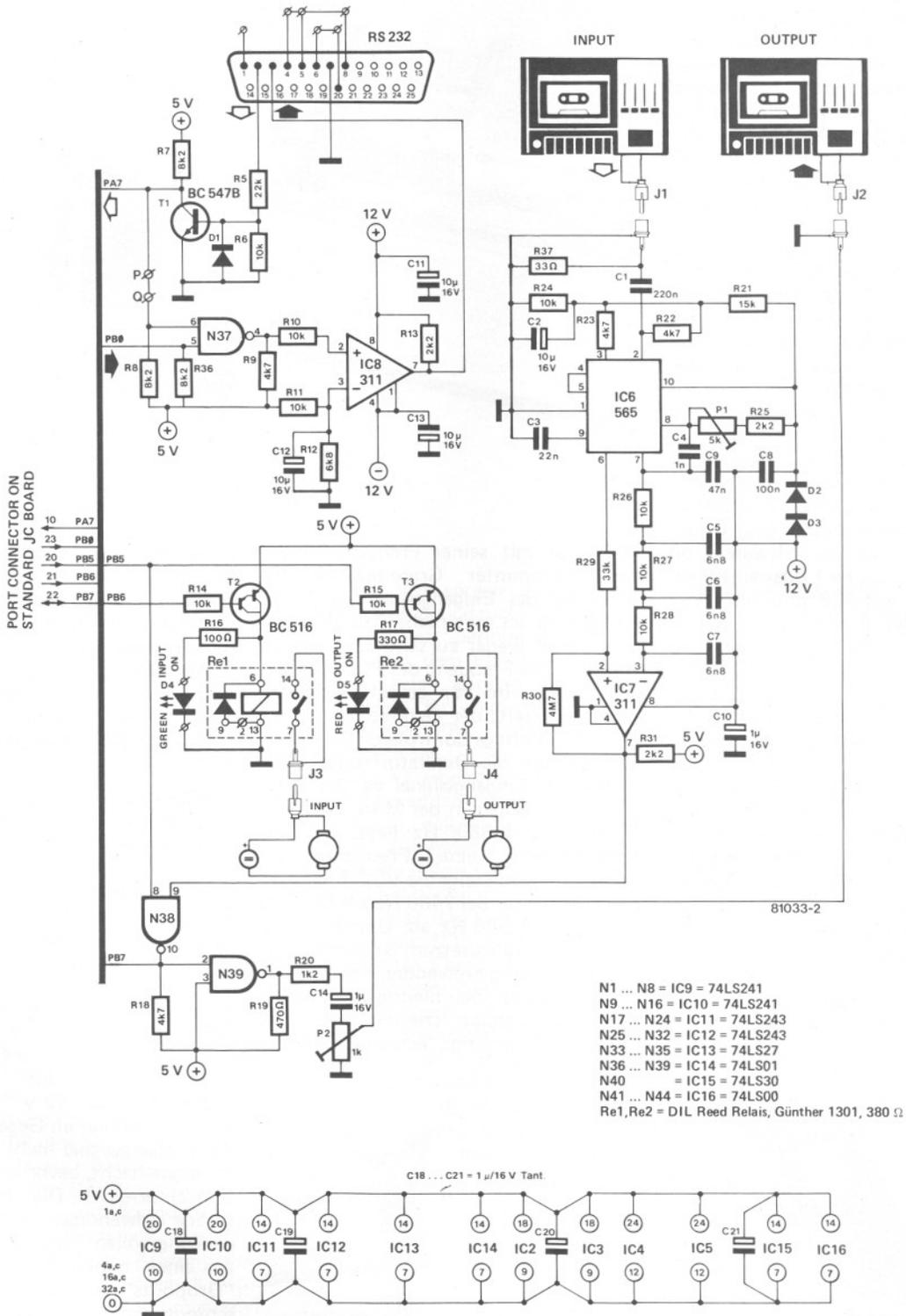


Bild 2. Das Kassetteninterface für zwei Rekorder und das RS-232-Interface zum Anschluß peripherer Geräte befinden sich ebenfalls auf der Interfacekarte.

so daß das Ausgangssignal von Komparator IC7 über Gatter N38 auf die Leitung PB7 gelangen kann. Dieses Signal wird auch von Gatter N39 weitergeleitet, was jedoch bedeutungslos ist. Da PB5 "1" ist, bleibt der an Buchse J4 liegende Relaiskontakt offen, und die rote LED leuchtet nicht auf. PB6 ist dagegen jetzt "0"; nun leitet T2 und läßt Relais Re1 anziehen und die grüne

LED aufleuchten. Über Buchse J3 kann der Strom des Antriebmotors von Kassettenrekorder INPUT fließen. Es ist nicht zwingend notwendig, zwei getrennte Kassettenrekorder für das Lesen und das Schreiben von Daten zu verwenden. Steht nur ein Kassettenrekorder zur Verfügung, dann werden die Kontakte der beiden Relais einfach parallelgeschaltet. Man kann aber auch ohne Relais "im Handbetrieb" arbeiten,

braucht in diesem Fall also gar kein Relais. Die visuelle Information über die Betriebsart, die die beiden LEDs liefern, hat sich übrigens in der Praxis als recht nützlich erwiesen. Auch hier steht die Farbe Rot für "Achtung, Aufnahme!", während die grüne LED "Freie Fahrt" für die Daten vom Band zum Computer signalisiert. Der Teil der Schaltung in Bild 2, der zwischen Buchse J1 und Gatter N38

liegt, wandelt die von Kassettenrekorder INPUT kommenden Tonsignale in computergerechte Signale zurück. Kennern des recht verbreiteten Einplatinencomputers KIM wird die Schaltung mit IC6 und IC7 sicherlich nicht unbekannt vorkommen. Dieser Eindruck ist richtig! Hier wurde auf Altbewährtes zurückgegriffen; lediglich einige kleinere Verbesserungen ergänzen die KIM-Kassetteninterface-Hardware (siehe dazu auch Schaltung 68 in Elektor, Juli/August '80).

Die Daten sind in der Weise auf dem Band gespeichert, daß Töne mit zwei unterschiedlichen Frequenzen einander abwechseln: Auf einen Ton mit einer Frequenz von ca. 3600 Hz folgt ein Ton von ca. 2400 Hz, danach wieder ein 3600-Hz-Ton gefolgt von einem 2400-Hz-Ton und so weiter. Jedes aus einem 3600-Hz-Ton und einem 2400-Hz-Ton bestehende Tonpaar repräsentiert ein Bit. An der Zeitdauer des 3600-Hz-Tons in Relation zur Länge des 2400-Hz-Tons erkennt der Prozessor (mit Unterstützung der Interface-Software), ob dieses Bit eine logische 1 oder eine 0 ist. Die Schaltung mit IC6 und IC7 setzt lediglich ein an Eingang J1 liegendes Tonsignal mit einer Frequenz von 3600 Hz in "1" und ein 2400-Hz-Tonsignal in "0" um. IC6 (ein 565) ist ein sogenanntes PLL-IC (PLL = Phase Locked Loop = phasenverriegelte Schleife). Über die Arbeitsweise derartiger Systeme ließe sich ein kompletter Elektor-Jahrgang füllen; die höhere Mathematik würde dabei eine wichtige Rolle spielen. Im Zusammenhang mit dem Kassetteninterface des Junior-Computers sollen jedoch einige beschreibende Sätze genügen:

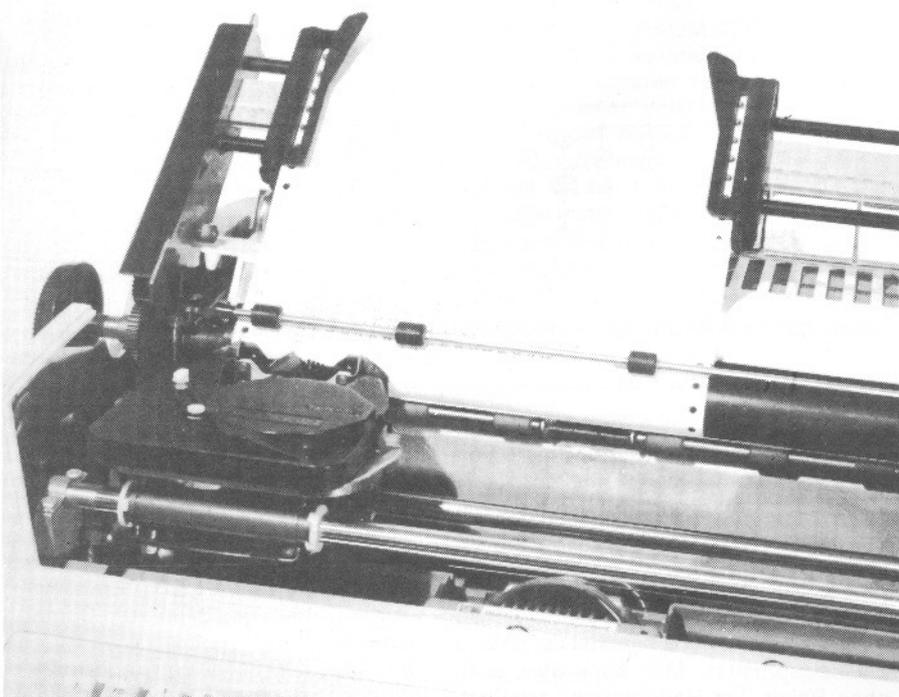
Das PLL-IC kann hier als Frequenzfolger betrachtet werden. Ebenso wie bei einem Emitterfolger die Ausgangsspannung der Eingangsspannung folgt, paßt sich bei dem PLL ein interner

Oszillator mit seiner Frequenz (innerhalb bestimmter Grenzen) an die Frequenz des Eingangssignals an. Voraussetzung ist natürlich, daß das Eingangssignal weder zu schwach noch zu stark ist. Die Frequenz des PLL-internen Oszillators wird von einer Gleichspannung gesteuert; der Oszillator ist daher ein VCO (Voltage Controlled Oscillator). Hier beträgt die Oszillatorfrequenz bei fehlendem Eingangssignal ca. 3000 Hz, so daß sie genau in der Mitte zwischen 2400 Hz und 3600 Hz liegt. Hat das Eingangssignal eine Frequenz von 3600 Hz, dann steigt die VCO-Frequenz um 600 Hz an, bei 2400 Hz am Eingang sinkt sie um 600 Hz ab. Um die VCO-Frequenz herabzusetzen, ist eine höhere Steuerspannung notwendig, während die VCO-Frequenz bei niedrigerer Steuerspannung ansteigt. Die Höhe der im PLL-IC selbsttätig erzeugten VCO-

Steuerspannung ist daher ein Kriterium für die Frequenz des Eingangssignals. Durch Vergleich mit einer Referenzspannung läßt sich eine logische Aussage darüber treffen, ob die Eingangsfrequenz über oder unter 3000 Hz liegt. Man nennt dieses Verfahren der Darstellung von digitalen Signalen auch Frequency Shift Keying, oder abgekürzt FSK.

Der 565 (IC6) wird mit Speisespannung über die beiden Dioden D2 und D3 versorgt, denen Kondensator C8 parallelgeschaltet ist. Dieses Siebglied unterdrückt wirkungsvoll Störimpulse auf der Speisespannung, die den Betrieb des PLL beeinträchtigen können. Die gereinigte Speisespannung an Pin 10 von IC6 wird auch zur Gleichspannungseinstellung der Eingänge (Pin 2 und 3) herangezogen. Der KIM benutzt hierzu eine ungesiebte 5-V-Speisespannung, obwohl auch dort eine saubere Spannung von 12 V vorhanden ist! Ferner wird hier im Gegensatz zum KIM das Eingangssignal nicht um den Faktor 10 abgeschwächt, bevor es den IC-Eingang (Pin 2) erreicht. Die genannten, nicht gerade aufwendigen, aber trotzdem sehr wirkungsvollen Verbesserungen führen zu einer hohen Störsicherheit gegen "Dropouts" auf dem Band und Verschmutzung oder falscher Einstellung des Tonkopfes.

Von den beiden Eingängen des 565 (Pin 2 und 3) wird nur einer vom Tonsignal gesteuert. Die beiden Eingänge gehören zu einem internen Differenzverstärker; hier liegt also asymmetrische Steuerung vor. Koppelkondensator C1 hat eine wesentlich niedrigere Kapazität als der entsprechende Kondensator des KIM, so daß hier Störspannungen mit niedrigen Frequenzen (z.B. Netzbrummen) besser unterdrückt werden. Widerstand R37 ist beim Anschluß von Buchse J1 an den Lautsprecher- oder Kopfhörerausgang eines Kassettenrekorders als Last bei abge-



schaltetem Lautsprecher notwendig. Die VCO-Frequenz ohne Eingangssignal wird von C3, R25 und P1 bestimmt. Trimpoti P1 ist sorgfältig auf die Mittenfrequenz von 3000 Hz einzustellen; hiervon hängt die Zuverlässigkeit der Datenübertragung maßgeblich ab. In Buch 3 sind mehrere Methoden angegeben, die sicher zum Ziel führen.

Pin 7 von IC6 ist der Ausgang des PLL. An diesen Pin ist die Steuerspannung des internen VCO herausgeführt. Kondensator C9, der zwischen dem Ausgang und +12 V liegt, bildet zusammen mit einem IC-internen Widerstand von 3k6 das sogenannte Schleifenfilter. Von der Dimensionierung des Schleifenfilters hängt das dynamische Verhalten des PLL bei Frequenzänderungen des Eingangssignals ab. Das Ausgangssignal von IC6 gelangt über den Tiefpaß R26... R28/C5... C7 zum invertierenden Eingang des Komparators IC7. Sein nichtinvertierender Eingang liegt über R29 an einer ebenfalls von IC6 gelieferten festen Gleichspannung.

Die Dimensionierung des Tiefpasses steht in Zusammenhang mit der Folgegeschwindigkeit der 3600- und 2400-Hz-Töne, die ihrerseits von der Aufzeichnungsgeschwindigkeit der Daten auf die Kassette abhängt. Diese Geschwindigkeit, auch Baudrate genannt, beträgt beim Junior-Computer 800 Baud; auf diesen Wert ist sowohl die Software als auch die Hardware abgestimmt.

Wie schon beschrieben hat eine hohe Eingangsfrequenz (3600 Hz) eine niedrige Gleichspannung am Ausgang des PLL zur Folge, bei niedriger Eingangsfrequenz (2400 Hz) ist die PLL-Ausgangsspannung dagegen hoch. Da am nichtinvertierenden Eingang von Komparator IC7 die mittlere Spannung liegt und sein Ausgang über R31 mit +5 V verbunden ist, erscheint an letzterem ein TTL-gerechtes logisches Signal: Der Ausgang von IC7 ist "1" bei 3600 Hz an J1 und "0" bei 2400 Hz an dieser Buchse. Das Signal wird von Gatter N38 invertiert und anschließend auf Leitung PB7 gesetzt.

Noch eine letzte Bemerkung zum Kassetteninterface: Der PLL kann nicht unendlich schnell auf Frequenzänderungen des Eingangssignale reagieren. Die Folge ist, daß der Ausgang von Komparator IC7 beim Wechsel von 3600 Hz nach 2400 Hz und umgekehrt nicht einfach von einem logischen Zustand zum anderen springt. Er pendelt kurzzeitig zwischen beiden hin und her, bevor der richtige Zustand stabil bleibt. Dieser sogenannte PLL-Jitter, der weitgehende Ähnlichkeit mit dem Prellen eines mechanischen Kontakts hat, wird von der Kassetteninterface-Software unterdrückt.

### RS-232-Peripherie

Was noch an Hardware übrig bleibt, ist minimal: der Teil der Schaltung, der sich in Bild 2 links oben befindet. T1,

N37 und IC8 sorgen zusammen mit einigen passiven Bauelementen dafür, daß Daten mit peripheren Einheiten, zum Beispiel dem Elekterminal, ausgetauscht werden können. Port PA7 ist der Eingang für ankommende Daten, während über PB0 Daten nach außen gesendet werden.

Transistor T1 arbeitet als Empfänger für die ankommenden Daten. Anzumerken ist dazu nur, daß er das Eingangssignal invertiert. Gesendet werden Daten von IC8: Da Pin 6 von Gatter N37 über R8 an +5 V liegt (Verbindung P-Q sei offen), invertiert N37 das Signal von PB0. Der Ausgang von N37 steuert den nichtinvertierenden Eingang von Komparator IC8, der dieses Signal mit der am Verbindungspunkt von R11 und R12 liegenden Spannung vergleicht. Die Spannung am Komparatorausgang beträgt +12 V, wenn der Ausgang von N37 logisch 1 ist; bei einer logischen 0 am Gatterausgang beträgt sie -12 V. Abgesehen von der Verschiebung der Spannungen für "0" und "1" findet auch zwischen PB0 und dem Ausgang von IC8 eine Invertierung des Signals statt.

Der Eingang des Datenempfängers und der Ausgang des Datensenders sind mit einem genormten 25-poligen D-Konnektor verbunden; er wird auch RS-232-Konnektor genannt. Die Norm RS-232 legt die Verfahrensweise für den Austausch von Daten fest: Die Daten werden seriell (bit für bit) übertragen, wobei jedem Byte ein Startbit vorangeht. Die Spannung, die logisch 0 repräsentiert, muß zwischen +5 V und +15 V liegen; für logisch 1 ist ein Spannungsbereich von -5 V und -15 V vorgeschrieben (Version RS-232-C). Beim Junior-Computer betragen diese Spannungen ca. +12 V und -12 V. Eine logische 1 wird somit durch eine niedrige Spannung und eine "0" durch eine hohe Spannung dargestellt. Diese negative

Logik bedingt, daß das ankommende und das abgehende Signal im Computer invertiert werden müssen, sofern man nicht mit "gespiegelten" Daten arbeiten will.

Am D-Konnektor, dessen Stiftbelegung von der Norm RS-232 ebenfalls festgelegt ist, sind zwischen bestimmten Kontakten Brücken angebracht. Welche Kontakte miteinander verbunden werden müssen, hängt von der Art des angeschlossenen Geräts ab. In Frage kommt vor allem des Elekterminal, ein Datensichtgerät mit ASCII-Tastatur (Schreibmaschinentastatur mit zusätzlichen Steuertasten), das ursprünglich für das Elektor-SC/MP-Mikrocomputersystem entwickelt wurde. Das Elekterminal ist auch für den Junior-Computer hervorragend geeignet, während der im Januar 1980 von Elektor veröffentlichte Metallfoliendrucker (vorläufig) **nicht** verwendet werden kann.

Damit ist die Beschreibung der Interface-Hardware für den Junior-Computer abgeschlossen. Die in ihrer Leistung angepaßte Speisung blieb ebenso wie einige Änderungen auf der Basisplatine unberücksichtigt. Beides wird an anderer Stelle in diesem Heft (und im Junior-Computer-Buch 3 behandelt). Dort sind eine ausführliche Bauanleitung zusammen mit den zugehörigen Platinen enthalten. ✻

