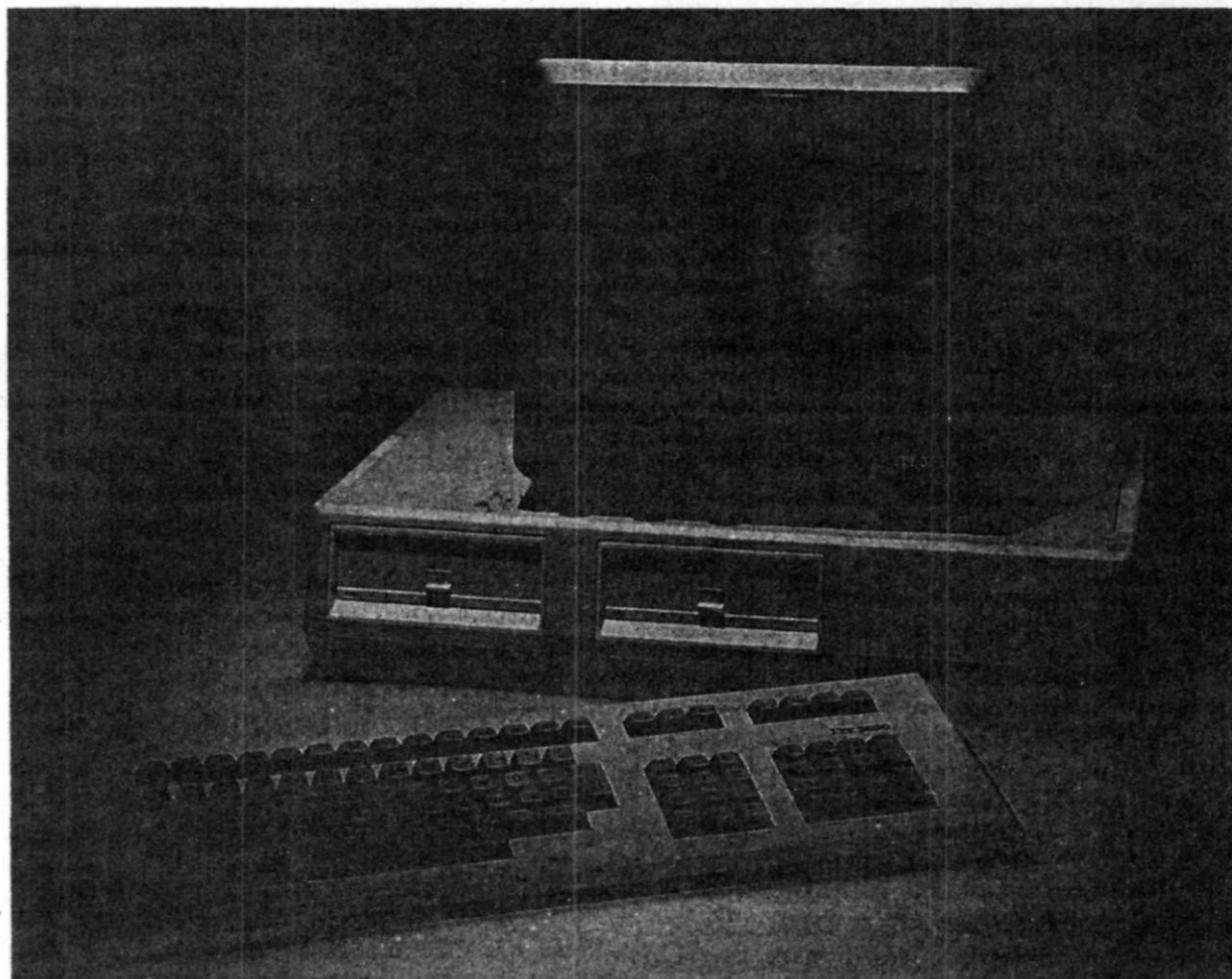




ITT 3030:

Wer mit einem neuen Tischcomputer zu nennenswerten Stückzahlen kommen will, muß längst mehr bieten als »nur« eine verlässliche Standard-CP/M-Maschine. Sowa basteln sich heutzutage talentierte Hobbyisten auf dem Küchentisch zusammen. Aufsehen erregen allenfalls noch solche Maschinen, die der Konkurrenz Wesentliches voraus haben. Als die 3030 erstmals angekündigt wurde, geriet sie deshalb publizistisch ein wenig in den Windschatten spektakulärerer Computer wie beispielsweise des BASIS 108, der von Anfang an Zwei-Prozessor-Betrieb zuließ und auf dem nicht nur CP/M-Programme, sondern gleichzeitig auch die Unmenge heute verfügbarer Apple-II-Software eingesetzt werden konnte. SEL dagegen brach mit der Vergangenheit. Man setzte die mit der 2020 begonnene Entwicklung nicht fort.

Apple und 2020 sind kleine Hochleistungsmaschinen für Leute mit »Computergefühl«, für engagierte Hobbyisten, für Techniker und Wissenschaftler, aber keine »Arbeitspferde«. Genau auf solche Computer jedoch, auf unverwüsthche Gebrauchsgeräte mit einem breiten Spektrum billiger Software, werden die Massenumsätze der Zukunft zukommen. Vor diesem Hintergrund sind die SEL-Entscheidungen in Pforzheim und Zuffenhausen zu sehen, auf einen Neubeginn zu setzen. Doch was dabei mit dem ITT 3030 herauskam, war nach der bei der Ankündigung bekannt gewordenen technischen Einzelheiten kaum dazu angetan, sonderliches Interesse von Mitbürgern mit einem Gespür für technische Kreativität zu wecken. Zweiter Prozessor? Irgendwann später. Farbe? Irgendwann später. Feingezeichnete Grafik? Irgendwann später. Pascal? Irgendwann später. Als die Erprobungsmaschine eintraf, war die Erwartungshaltung dementsprechend zwiespältig: Einerseits der hervorragende Ruf des Anbieters aus alten Tagen, andererseits das Vorwissen über die Technik einer neuen Maschine, die vieles noch nicht konnte, was die Vorgängerin in Perfektion zeigte. Jetzt, nach ein paar Wochen Arbeit mit der 3030, ist man klüger geworden. Es wurden Mängel gefunden, einige davon so schlimm, daß sie so nicht bleiben können (und sie werden so nicht bleiben: Erprobt wurde hauptsächlich eine Betriebssystem-



Mit dem ITT 3030 setzt man bei SEL auf einen Neubeginn

Prototyp kritisch erprobt

Wenn ein Unternehmen von der Bedeutung der Standard Elektrik Lorenz ein neues Mikrocomputer-System anbietet, dann erwartet man von einem solchen Produkt mehr als von Erzeugnissen kalifornischer Jungfirmen oder deutscher Zwölf-Mann-Betriebe. Schließlich baute die SEL unter Professor Karl Steinbuch den ersten Transistor-Großrechner der Welt, und die berühmte ER 56 aus Zuffenhausen lebt, obwohl längst verschrottet, in der Literatur zur Technikgeschichte weiter fort. Darüber hinaus war der von einer SEL-Firma vertriebene Apple-Abkömmling ITT 2020 nach dem PET 2001 der zweite wirkliche Massencomputer diesseits des großen Teiches. Hält die ITT 3030, was der Name SEL verspricht? Hier finden Sie die Ergebnisse einer ausgedehnten Betriebserprobung.



Prototypversion, deren Macken bei Erscheinen dieses Berichtes wohl längst ausgebügelt sein werden): Doch alles in allem kommt man zu einem ausnehmend günstigen Urteil. Warum und wie, das soll hier belegt werden.

Software legt Tastenbedeutung fest

Monitor und Tastatur sind abgesetzte Einheiten, die durch Kabel mit dem Rechner verbunden sind, der zwei Philips-Laufwerke für Fünf-Zoll-Disketten enthält. Design und Farbgestaltung sind so beschaffen, daß der 3030 als ein ausgesprochen schöner Computer angesehen werden kann, durchaus geeignet beispielsweise als Zierstück eines von Innenarchitekten gestalteten Büros. Mobile Flach-Tastatur und ein scharf gezeichnetes Schriftbild auf dem 28-cm-Bildschirm machten stundenlange Texteingabe in einem Maße vergleichsweise ermüdungsfrei möglich, wie ich das bisher nur bei sehr wenigen anderen Rechnern erlebt habe — dies freilich erst dann, nachdem man sich an einige Eigenheiten von Tastenbelegung und Schirmdarstellung gewöhnt hatte, die zwar bei der Textverarbeitung nicht stören, dafür jedoch um so mehr beim Programmieren: Darüber später mehr. Die Längen der Verbindungskabel sind so bemessen, daß man zwar zu einiger Freizügigkeit auf der Schreibtischplatte kommt, doch auch nicht zu mehr. Wer den Rechner gern »ein Stockwerk tiefer« im Schreibtisch hätte und auf der Tischplatte nur Monitor und Tastenfeld dulden möchte, scheitert an den zu kurzen Kabeln.

Das Schirmbild ist, wie gesagt, auch über längere Zeit ermüdungsfrei lesbar, doch problematisch war, daß der Cursor nicht blinkte: Sobald Inversdarstellung zum Einsatz kam, wußte man oft nicht mehr so recht, »wo« man eigentlich »war«,

und in Normaldarstellung waren Buchstaben wie kleines »h« und kleines »n« immer dann kaum zu unterscheiden, wenn links von diesen Zeichen das statische Cursor-Leuchtquadrat stand. Vermutlich bedarf es lediglich der Änderung eines einzigen Bytes im Betriebssystem, genauer: Die Initialisierung des Video-Controllers müßte modifiziert werden, um dem ITT 3030 das Blinken oder eine andere Art der besseren Cursor-Identifikation beizubringen.

Die Tastatur macht das Schreiben zum Genuß, geht es nicht gerade um Pascal-Text oder C-Source-Code: Hier macht sich höchst ärgerlich bemerkbar, daß zwar eine »Shift-Lock«-Taste vorhanden ist, die viel wichtigere »Caps-Lock«-Taste jedoch nicht. Wer in Großbuchstaben schreiben will und deshalb die »Shift-Lock« betätigt, kann die häufigsten Interpunktionszeichen und alle Zahlen ohne Lösen dieser Taste nicht mehr benutzen. Das mag für Textautomaten oder reinrassige Basic-Computer tolerierbar sein, für eine CP/M-Maschine ist es das keinesfalls.

Die Control-Taste ist an den Rand der Welt verbannt, ans linke obere Eck: So ziemlich der ungünstigste Platz für CP/M-Programmierung und die Benutzung vieler CP/M-Anwendersoftware. Freilich ist den Tastenfeld-Gestaltern hier zugute zu halten, daß eine Fülle von Control-Zeichen durch zahlreiche Zusatztasten unmittelbar erreichbar sind; diese Tasten sind mit Bezeichnungen wie »HOME«, »ROLL« und »LOE« versehen und zur Verwendung mit speziell konfigurierten Editoren gedacht, was der Bedienung durch Laien ohne Zweifel sehr entgegenkommt: Allerdings nur dann, wenn man diese speziell für den ITT 3030 konfigurierte Software kauft.

Da längst nicht alle ITT 3030-Käufer Computer-Novizen sein dürften, sondern sich unter ihnen auch Zeitgenossen befinden, die bereits ei-

niges Geld für CP/M-Software ausgaben und diese nun weiterverwenden möchten, unternahm ich den Versuch, mein gewohntes

Liebblings-Textverarbeitungsprogramm auf den ITT 3030 zu überspielen und dort »umzuinstallieren«. Dies gelang prächtig nach ein paar Stunden Vorarbeit, welche der Erkundung bislang unveröffentlichter System-Parameter des Rechners galten (Tabelle 1).

Installations - Parameter für ITT - 30 30	
MicroPro-Produkte wie Wordstar, Datastar usw. (erprobt für MOS-Version 5.2 DT)	
Install-Adresse	Werte (Hexnotation)
HITE	18
WID	50
CLEAD1	02 1B 16
CLEAD2	00
CTRAIL	00
CB4LFG	00
LINOFF	01
COLOFF	00
ASCUR	00
HIBIV	FF

Tabelle 1. Wer Software auf dem ITT 3030 installieren möchte, dem helfen diese maschinenspezifischen Parameter, hier mit den von MicroPro gewählten symbolischen Namen bezeichnet.

Zurück zum Tastenfeld: Ich hatte vor Erprobung dieses Computers längere Zeit mit einer CS-2000 gearbeitet und vermißte die Autorepeat-Funktion von deren Ampex-Terminal auf dem SEL-Computer sehr. Statt dessen hat der ITT 3030 eine gesonderte Repeat-Taste, die sich höchst merkwürdig benahm: Sollte beispielsweise eine Folge von Control-X-Zeichen wiederholt werden, dann gab die Repeat-Funktion nach anfänglicher Wiederholung von Control-X unversehens eine Reihe von schlichten »X«-Zeichen von sich. Die Behebung dieser Prototyp-Mängel dürfte in

Fortschritte im Detail bestimmen die Zukunft.

Wie z.B. der cw-Wert den Luftwiderstand beim Fahrzeug, der k-Wert die Dämmfähigkeit beim Mauerwerk

oder der „e-Wert“ die ergonomische Softwaregestaltung . . .





der nun anstehenden Serienfertigung schon deshalb vergleichsweise unproblematisch sein, weil die Tastenfunktionen nicht durch Hardware festgelegt sind, sondern durch Tabellen gesteuert werden, die mit dem Betriebssystem von Diskette geladen werden. Durch schlichten Disketten-Wechsel kommt man blitzschnell in den Genuß einer völlig »neuen« Tastatur mit amerikanischer, englischer oder französischer Tastenanordnung (wobei die Original-Gravur natürlich bleibt).

SEL-Cheftechniker Lehmann kündigte an, daß in Kürze sogar der Zeichengenerator für die Video-Ausgabe auf nichtresidente Tabellen umgestellt wird. Das würde bedeuten, daß man sich die unterschiedlichsten Schriftarten für das Schirmbild in die Maschine laden kann: Eine phantastische Sache für Leute, die neben den »Ä« für ihren deutschen Schriftverkehr auch andere Sonderzeichen für Korrespondenz in anderen Sprachen oder die berühmten eckigen Klammern für Pascal-Texte und Anweisungen an CP/M's PIP brauchen. Moderne Drucker haben diese Umschaltbarkeit längst, aber die Schirmbilder halten bei den meisten Mikrocomputern heute leider noch nicht mit.

Ein »zusammengeknöpfter Schnapp-Computer«

Kommen wir zum mechanischen Aufbau dieser neuen Maschine. Unlängst hatte ein Synelec-Techniker gezeigt, wie man lediglich mit einem kleinen Inbus-Schlüssel aus der Geldbörse einen M-Three blitzschnell zerlegt und wieder zusammenbaut. Nun, in noch kürzerer Zeit und dazu völlig ohne Werkzeug schaffte es ein SEL-Mann, alle Karten und Laufwerke aus dem Gehäuse zu holen. Der ITT 3030 ist ein mechanisch höchst sinnreich konstruiertes Gebilde aus »zusammengeknöpften« Komponenten, das trotz des unorthodoxen Aufbaus

gleichwohl einen recht zuverlässigen Eindruck macht.

Verbesserungswürdig ist der Lüfter, für meinen Geschmack ein bißchen laut, zumindest was den Betrieb außerhalb gewöhnlicher Büro-Atmosphäre etwa zur Nachtzeit in Wohnräumen angeht (und welcher Programmierer arbeitet nicht auch mal nachts in Wohnräumen?), und von den Disketten-Laufwerken waren oft Geräusche zu hören, als würde jemand eine knarrende Tür hin- und herschwingen lassen. Neben der akustischen »Umweltverschmutzung« fiel noch eine »elektromagnetische« Umweltbeeinträchtigung auf: Würden Computer und Heimfernseher an der gleichen Hauptsteckdose betrieben, gab's deutlich wahrnehmbare Bildstörungen (Empfänger: ITT Ideal Color 3200). Auch der Monitor zeigte sich, elektromagnetisch betrachtet, durchaus als störend. Er brachte, wenn er leicht gedreht rechts auf dem Rechner stand, den Computer durch Störstrahlung außer Tritt. Mit der üblen Folge, daß durch doppelten Disketteninhaltsverlust einmal ein paar Stunden Arbeit verloren gingen. Nachdem der Monitor nicht mehr länger auf dem Rechner, sondern nur noch daneben aufgestellt wurde, gab es keine einzige zerstörte Aufzeichnung mehr. Kommentar aus Pforzheim: »Diesen Prototyp-Fehler haben wir auch schon bemerkt, das kommt bei künftigen Handelsgeräten nicht vor.« Ich hatte, dies beiläufig bemerkt, anfänglich eine Vorserienmaschine in Betrieb, die problemlos lief. Wegen des für die Arbeit mit Pascal MTPPlus etwas geringem Fassungsvermögens der ursprünglichen Standard-Laufwerke wurden die Laufwerke auf eine höhere Kapazität (96 tpi/rund 560 KByte) umgerüstet, doch war diese Konfiguration durch die Qualitätskontrolle damals noch nicht freigegeben. Insofern sind diese Systemzusammenbrüche dem Anbieter nicht anzulasten.

Zwischen dem BIOS und der Hardware sitzt »MOS«

Moderne CP/M-Maschinen haben heutzutage einen Urloader, der sich nach dem Booten »ausknipst«, so daß der volle RAM-Bereich für CP/M und das Anwenderprogramm zur Verfügung steht. Die Kommunikation mit der Hardware übernimmt dabei der maschinenspezifische Betriebssystem-Teil von CP/M, das BIOS (Basic I/O System). Bei dem ITT 3030 kommt dem BIOS nur Briefträger-Funktion zu; es reicht weiter an ein von Standard Elektrik Lorenz »MOS« (Machine Operating System) genanntes Dienstprogramm heran, das über genormte Schnittstellen verfügt und zusammen mit CP/M beim Booten geladen wird. Fachleute wollen übrigens in der zur Erprobungszeit jüngsten MOS-Version Betriebssystem-Eigenheiten eines nicht sehr erfolgreichen anderen deutschen Mikrocomputers wiedererkennen, und es gibt Gerüchte, wonach diese Systemteile nach hausinternem Knatsch bei SEL nun durch einen effizienteren Code ersetzt werden. Vorzug einer solchen Konzeption: Der ITT 3030 ist von CP/M unabhängig; man kann ihn auch mit einem Betriebssystem »BOS« (Business Operating System) haben, laut SEL bald auch mit UCSD und laut Branchengerüchten sogar mit Oasis.

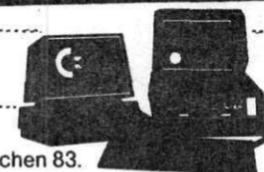
RAM-Aufrüstung bis zu mehreren hundert KByte ist heute schon möglich, der Steckplatz für eine 16-Bit-CPU ist bereits »verdrahtet«. Derlei Flexibilität kostet ihren Preis: Eine solche »zukunfts-feste« Konfiguration nimmt mehr Speicherraum für den Eigenbedarf in Anspruch, als dies ein hochgezüchteter reinrassiger CP/M-Computer tut. Und dieser Mehrbedarf ist gewaltig: Der ITT 3030 bietet dem Benutzer, wenn er sich nach dem Booten mit der stolzen Meldezeile »64 KByte CP/M« bemerkbar macht, nur ein einziges

SM IBIS. Das integrierte kaufmännische Programmpaket für Commodore- und Sirius-Computer

mit höchstem Bedienungskomfort durch optimale Abstimmung von Hardware und Software. Ein Musterbeispiel

für praktizierte Software-Ergonomie. Problemloser Einsatz auch für den EDV-Laien.

Näheres erfahren Sie bei Ihrem Commodore- oder Sirius-Fachhändler oder direkt bei
SM SOFTWAREVERBUND MICROCOMPUTER GMBH, Scherbaumstraße 33, 8000 München 83.





KByte mehr Netto-Speicher als eine 2020 mit Soft Card und Language Card an (wobei die sich dann allerdings, ein bißchen ehrlicher, mit »Apple CP/M 56K« meldet).

Auch für die »speicherplatzfressende« Betriebssystem-Variante, die zur Erprobung zur Verfügung stand, gilt: Ein Prototyp; man kann wohl guter Hoffnung sein, daß gewisse MOS-Passagen demnächst effizienter formuliert vorliegen, doch viel Spielraum bleibt dabei nicht: Die Übergangsstelle zwischen MOS und dem jeweiligen Betriebssystem ist so festgelegt, daß die oberen 6 KByte für das MOS reserviert werden. Es gibt Hinweise für die bevorstehende Einführung einer CP/M-Neuversion mit Bank-Switching: Dafür wäre der ITT 3030 dann allerdings hervorragend geeignet. Solange es freilich noch nicht soweit ist, gilt die Faustformel: Das angebliche 64-KByte-CP/M auf dem ITT 3030 ist faktisch nicht viel mehr als ein schlichtes 57-KByte-CP/M; den Beweis für diese Behauptung entnehme man Tabelle 2 in Zusammenschau mit Bild 1.

I/O: Ziemlich flott für fünf Zoll — CPU: ausgesprochen fix

Wie steht es mit der Arbeitsgeschwindigkeit von CPU und Disk-I/O? Hier verdient der ITT 3030 glänzende Noten, wenn man in Betracht zieht, daß 5¼-Zoll-Laufwerke generell nicht die gleiche Datendurchsatz-Geschwindigkeit erlauben wie 8-Zoll-Drives. Zur Messung

wurde wieder, wie bereits mehrfach in der Vergangenheit, unser unterdessen schon ein bißchen »genormtes« Versuchspaket herangezogen, das verschiedene Geschwindigkeitskriterien berücksichtigt. Programme und Ergebnisse finden sich in den Bildern 2 und 3 sowie in Tabelle 3. Meßanordnung: Eichquarz und Zähler, über ein Tor verbunden, das von einem Akustik-Schalter geöffnet und geschlossen wird. Den wieder steuert der Beeper (bei SEL heißt er »Hupe«) im Erprobungs-Rechner. Einschwingvorgänge kompensieren sich bei diesem Verfahren; mögliche Differenzen zwischen Einschaltverzögerung und Ausschaltverzögerung schlagen sich im Ergebnis innerhalb der verwerteten Stellenzahl nicht nieder. Ergebnis: Unter CP/M ist der ITT 3030 eine ausgesprochen fixe Maschine (wie es unter BOS steht, wurde nicht untersucht), und dies macht sich beim Sortieren größerer Datenmengen und bei der Arbeit mit dem Wordstar angenehm bemerkbar.

Beim TIMTEST hätte man aufgrund der vom Hersteller angegebenen System-Takt-Frequenz wohl ein um mindestens zehn Prozent besseres Ergebnis erwartet — wo die »verschundene Zeit« geblieben ist, entzieht sich meiner Kenntnis, doch mag es wohl so sein, daß die CPU gelegentlich auch dann durch Interrupts von der Arbeit abgehalten wird, wenn dies nicht erkennbar ist. Ob Zusammenhänge mit der Schirmbedienung oder der Tastaturabfrage bestehen, ob irgendwo gelegentlich WAIT-Zyklen

»Sand ins Getriebe streuen«, das mögen Experten beantworten. Halten wir zum Thema Geschwindigkeit fest: Der ITT 3030 ist bei der Schirmausgabe blitzschnell, beim Rechnen ausgesprochen flink und beim Disk-I/O noch immer beachtlich flott.

»HEXSEND«-Verfahren zum Software-Transfer im Intel-Format

Vom Hersteller wurden verschiedene Programmpakete zur Erprobung mitgegeben, auf deren Einsatz jedoch aus guten Gründen verzichtet wurde: Es sollte nicht aus der Perspektive eines Benutzers geurteilt werden, der neben den Kosten für den Computer noch ein paar Tausender zusätzlich für Software ausgeben kann; statt dessen sollte auf »Altbestände« zurückgegriffen und diese für den ITT 3030 gegebenenfalls umgebaut werden. Damit trug man der Tatsache Rechnung, daß viele Kunden in ähnlicher Lage sind: Man kauft einen neuen Computer, möchte die gewohnte und bezahlte »Altsoftware« jedoch möglichst weiterverwenden.

Deshalb wurde ein neuartiges Übertragungsverfahren entwickelt, das gegenüber den bisher publizierten Transfer-Arten zwei gewichtige Vorteile bietet: Außer der Standard-CP/M-Grundsoftware (hier PIP, SUBMIT und LOAD) sollte die Empfangsmaschine keinerlei gesondertes Empfangsprogramm benötigen, und die Übertragung sollte gleichwohl mit einem zuverlässigen Prüfsummenverfahren gesichert werden. Dies gelang auf

Speicherraum fuer Anwenderprogramme (TPA Transient Program Area unter CP/M)		
Maschine	BS-Variante, Meldezeile	TPA (Byte)
ITT 2020 mit Language Card und Soft Card	APPLE II 56 k CP/M VER. 2.20B	51 968
CS-2000 mit Computershop-Bios	CP/M Vers 2.2/1.9 62k+8"+M10+Mult-I0/2	56 064
CS-2000 mit Economy-Bios	CP/M 64k Vers 2.2/Joebios4	58 112
ITT 3030	64K CP/M VERSION 2.2 - 5.2 DT	52 992
	u 5.27 ST/W	49 932

Tabelle 2. Das angebliche »64 Kilobyte CP/M« des ITT 3030 bietet nur 1 KByte mehr an Benutzerspeicher als der ITT 2020 mit »ehrlichem« 56 KByte-CP/M, verriet das Testprogramm aus Bild 1. Hier weitere Vergleichswerte.

Ermittlung des Benutzer-Anteils an Schreib-Lese-Speicher	
(Erprobt mit Compiler BASCOM Vers 5.24 und Interpreter Basic 80 Vers 5.21)	
2 PRINT	
4 PRINT "	TPA SIZE (Joe/Sep 82)"
6 PRINT "	_____"
8 PRINT	
10 LO=PEEK(6)	
12 HI=PEEK(7)	
14 NUM=LO+256*HI	
16 NUM=NUM - 256 - 6 'subtract page zero and jmp table bytes	
18 PRINT "TPA Size in Bytes: "	
20 PRINT NUM; " (";	
22 PRINT HEX\$(NUM); "H),-" ; PRINT	

Bild 1. Für Benutzer ist unwichtig, wieviel an Speicher ihre Computer absolut haben — es kommt allein darauf an, wieviel Speicherplätze das Betriebssystem dem Kunden letztendlich läßt. Und das ermittelt dieses kleine Basic-Programm zuverlässig. Über die 3030 hatte es insoweit wenig Schmeichelhaftes zu sagen.


```
program hexsend ( 4-9-82, derived from unload 1027 30-6-82 , xperimental );
```

```
const delay = 7 (hardware dependent; this will do on all machines (?));
```

```
type feld      = array [1..16] of byte;
     spt      = ^string;
```

```
var namp      : spt;
    showcom   : boolean;
    c         : char;
    choice    : (punch, lister);
    buf       : feld;
    infil     : file of feld;
    nam       : string;
    i (index), result : integer;
    ty (type),
    da (data),
    cs (compls),
    s (sum)   : byte;
    la (load address) : word;
```

```
external function @cmd: spt; (So we get the address of COMMAND ARGUMENT)
external function @dos (func, parm: integer): integer; (Let's call CP/M)
```

```
procedure pput (c: char) (put char to output channel);
var n: integer; mydelay: integer (for testing purposes);
begin if choice = punch
      then begin mydelay:= delay;
              while mydelay > 0 do mydelay:= mydelay - 1;
                  n:= @dos (4, ord(c))
              end
      else n:= @dos (5, ord(c))
end;
```

```
procedure adjust (var i: integer) (convert nibble to char);
begin i:= i + $30; if i > $39 then i:= i + 7; end;
```

```
procedure pwb (b: byte) (write byte as two hex chars to output channel);
var hi, lo: integer;
begin hi:= ord (b) div 16; lo:= ord (b) mod 16;
      adjust (hi); adjust (lo);
      pput (chr (hi)); pput (chr (lo))
end;
```

```
procedure pws (s: string) (put string to output channel);
var n, l: integer; c: char;
begin l:= length (s);
      for n:= 1 to l do begin c:= s[n]; pput (c) end
end;
```

```
procedure pwcrlf (put cr/lf to output channel);
begin pput (chr($0d)); pput (chr($0a)) end;
```

```
procedure pwdd (i: word) (write integer as four chars in hex);
var b1, b2 : byte;
begin b1:= hi (i); b2:= lo (i); pwb (b1); pwb (b2) end;
```

```
procedure wrdat ( write data to ofil: 16 bytes ->
```

```
32 chars, updt s ); begin
s:= $10 + hi (la) + lo (la); i:= 0;
while i < 16 do
begin
i:= i + 1; da:= buf[i];
s:= s + da; pwb (da)
end;
end;
```

```
procedure wrrec ( write hex record to ofil );
```

```
begin
pws (':10'); pwdd (la); pws ('00');
wrdat; cs:= -s; pwb (cs); pwcrlf
end;
```

```
procedure wrend ( write end hex record to ofil );
```

```
begin pws (':0000000000'); pwcrlf end;
```

```
begin ( ----- HEXSEND ----- )
```

```
writeln ('program hexsend (Joe/4-09-82)');
writeln ('-----'); namp:= @cmd; nam:= namp^;
if (nam = '') or (nam = ' ') (id est, we go no argument directly:)
then
begin
write (' name of file to be sent? '); readln (nam);
repeat writeln; write (' output to Punch or to Lister (P/L)? '); read (c)
until c in ['P', 'p', 'L', 'l'];
if c in ['l', 'L'] then choice:= lister else choice:= punch; writeln;
if choice = punch
then begin repeat writeln;
            write (' Showing progress at work (y/n) ? '); read (c)
            until c in ['N', 'n', 'Y', 'y'];
            showcom:= (c = 'y') or (c = 'Y');
            end;
else begin choice:= punch; showcom:= true; writeln;
        writeln (' to send: ', nam); writeln
        end;
writeln; assign (infil, nam); reset (infil);
if eof (infil) then begin writeln (' >>> ', Nam, ' not found.-'); exit end;
la:= wrd ($100); ( --> If your tpa doesn't start at 100H, change address)
writeln;
repeat
buf:= infil^; get (infil); wrrec ( write data hex record );
if showcom then write (' ');
la:= la + wrd ($10);
until eof (infil);
wrend ( write end hex record ); writeln;
la:= la - wrd (1);
if choice = punch then pput ( chr ($1a) ); ( Sending EOF char to receiver)
writeln; write (' XXXX Ready; last load address: ');
writehex (output, la, 2); writeln('H.-')
end.
```

Bild 4. Wollen Sie mit Prüfsummen-Sicherung Programme in einen Computer übertragen, der keinerlei Sonder-Software für Datenempfang und Datenprüfung hat? Mit dem »Hexsend«-Programm in der Sendemaschine geht es — hier formuliert für den Compiler Pascal MTPlus, doch läßt sich Hexsend auch in ältere Pascal-Varianten wie UCSD übertragen. Mit dieser Neuentwicklung kann der ITT 3030 softwaremäßig gewaltig aufgerüstet werden.

aus der Zeit der Streifenleser stammt. Die so von ASM geschriebenen Files tragen die Extension »HEX«. Sie werden durch das Dienstprogramm »LOAD« in vollziehbaren Code zurückverwandelt.

Das Verfahren ist nun dadurch gekennzeichnet, daß die Sendemaschine die zu übertragenden Programme in einem »UNLOAD-Prozeß« ins Intel-Hex-Format übersetzt und wahlweise über den LST-Kanal oder den PUN-Kanal ausgibt. Empfangen wird von PIP über RDR oder einen anderen elektrisch passenden Kanal, wobei PIP als Extension der Files die Bezeichnung »HEX« vorgegeben wird, dem Peripheral Interface Processor also »weismacht« wird, man nähme eine Lochstreifenübertragung vor.

PIP prüft dabei auf korrektes »Lochstreifen-Format« und nimmt somit bereits schon eine erste Übertragungssicherung vor. Das von der Empfangsmaschine aufgezeichnete »HEX-File« wird nun dem Dienstprogramm LOAD übergeben. Das nimmt eine »Feinkontrolle« anhand der Prüfsummen vor und übersetzt ins »COM-Format«. Trug das Ausgangsfile eine andere Extension, so ist abschließend nur noch durch Umbenennung des empfangenen Files der Endzustand herzustellen.

Dieses Hexsend-Verfahren ist wesentlich langsamer, als es Übertragungen mit speziellen Transferprogrammen sind, die mit einem besonderen Software-Empfänger arbeiten — doch wie einen solchen

Empfänger erst mal in die Zielmaschine schaffen, wenn die Disketten inkompatibel sind? Die Lösung bisher: Das »Empfänger-Programm« von Hand eintippen und dann kompilieren — keine sehr effektive Arbeit. Sie wird durch Hexsend überflüssig. Das Sendeprogramm ist in Pascal für den Compiler MTPlus geschrieben und wird, weil es für die Erprobung des ITT 3030 entwickelt wurde, hier gleich mit vorgestellt (Bild 4). Der Aufruf erfolgt auf der Senderseite durch Namensangabe und wahlweise dem nachgestellten Namen des zu übertragenden Programmes als Argument. Wie dieses Programm im einzelnen arbeitet, lesen Sie im zweiten Teil dieses Testberichtes. (Hans-Georg Joepgen)

Auf Seite 86 in Ausgabe 24 von Computer persönlich war das Programm »Hexsend« zur Übertragung von Daten von Rechner zu Rechner in Bild 4 abgebildet, das dem Benutzer die Wahl läßt, ob über den PUN oder den LST-Kanal gesendet werden soll. Das »Hexsend« fragt an, ob kontinuierliche Unterrichtung des Operators zum Fortschritt der Arbeit erwünscht sei oder ob »stumm«, dafür aber schneller, übertragen werden soll. Mit nachgestelltem Argument erfolgt Übertragung immer über PUN, diese Betriebsart ist batchtauglich und erlaubt damit Stapelbetrieb, den SUBMIT-Files steuern. Ein Beispiel: Es sollen die Files EINS.COM, ZWEI.TXT und DREI.OVR übertragen werden. Auf der sendenden Maschine wird mit ED oder einem anderen Editor ein Submit-File HEXSEND.SUB folgenden Inhaltes

```
HEXSEND EINS.COM
HEXSEND ZWEI.TXT
HEXSEND DREI.OVR
```

generiert und die Übertragung mit der Anweisung SUBMIT HEXSEND ausgelöst. Auf der Empfangsmaschine muß ein Submit-File HEXGET.SUB bestehen, das folgenden Inhalt hat:

```
PIP EINS.HEX = RDR:
LOAD EINS
PIP ZWEI.HEX = RDR:
LOAD ZWEI
REN ZWEI.TXT = ZWEI.COM
PIP DREI.HEX = RDR:
LOAD DREI
REN DREI.OVR = DREI.COM
ERA *.HEX
```

Der Start erfolgt hier mit der Tastenfeld-Anweisung »SUBMIT HEXGET«. Ordentliches Funktionieren des Quittungsbetriebes zwischen den I/O-Ports der beiden Rechner vorausgesetzt, läuft die Übertragung nun mit hochwirksamer Prüfsummierung selbsttätig ab, wobei — dies sei noch einmal ausdrücklich betont — auf der Empfängerseite nur die zur CP/M-Standardlieferung gehörenden Dienstprogramme PIP.COM, SUBMIT.COM und LOAD.COM benötigt werden, die man mit dem ITT 3030-CP/M ohne Aufpreis bekommt. Bei der Rechner-Erprobung war vor allem wesentlich, daß man das gesamte Sprachsystem Pascal MTPlus (Version 5.5) auf den ITT 3030 herüberholen und dort höchst zufriedenstel-



ITT 3030: Prototyp kritisch erprobt

Nachdem wir uns in der letzten Ausgabe von Computer persönlich mit der Entstehungsgeschichte des ITT 3030 befaßt und die Hardware — vom Systemteil über den Monitor bis hin zur Tastatur — eingehend beschrieben haben, wollen wir nun im vorliegenden zweiten Teil dieses Testberichtes auf den Datentransfer von Rechner zu Rechner und die Schnittstellen eingehen.

lend damit arbeiten konnte, zumal es noch kein anderes Pascal im ITT 3030-Format gab. Compiler, Linker, Debugger, Librarian und Disassembler taten, mit einer Ausnahme, wie von anderen Maschinen gewohnt ihre Pflicht. Die Ausnahme betraf das Rechnerverhalten im seltenen Falle eines vorzeitigen Kompilations-Abbruchs. Wo andere CP/M-Maschinen einen Warmstart auslösen und den »Console Command Prozessor« CCP des DOS (Disk Operating System) zu-

rück in den Speicher holen, blieb der ITT schlicht hängen und war nur durch den RESET-Knopf (mit Recht auf der Gehäuse-Rückwand versteckt) wieder zu aktivieren: Unschädlich, weil zu diesem Zeitpunkt ohnehin alle Files wieder geschlossen sind. Da MTPlus den ohnehin knappen Speicherraum des Rechners beim Kompilieren nahezu voll nutzt, nehmen wir an, daß es das ITT-Betriebssystem MOS nicht »mag«, wenn ihm jemand so »dicht auf den Pelz rückt«.



Der Parallel-Port: Trauerspiel mit Happy End

Ob man Druckerdaten seriell oder parallel überträgt, ist ein bißchen Glaubenssache. Was mich angeht, so meine ich, Computer stellen Bytes parallel zur Verfügung, Drucker brauchen Bytes letztendlich parallel, und für eine Übertragungstrecke von vielleicht lediglich einem Meter parallele Daten in serielle zu verwandeln

und dann wieder zurück, sei bei Licht betrachtet technisch Schwachsinn. Überdies müssen für eine serielle V.24-Schnittstelle oft extra Spannungen produziert werden, die für Computer aus modernen Bauteilen systemfremd sind, drittens sind serielle Sender und Empfänger oftmals wesentlich teurer als einfache Parallel-I/O-Bausteine. Hinzu kommt noch: Bei sonst gleichen Randbedingungen »bla-

sen« serielle Übertragungsleitungen wegen der höheren Spannungen und der höheren Pegelwechsel-Frequenz rund zwölfmal mehr Energie in Form elektromagnetischer Umweltverschmutzung ins Umfeld, als dies Parallel-Druker-Leitungen tun. Vier gute Gründe also, die für Parallelbetrieb sprechen, aber es gibt es nun ein-

```
=====
+++++ LST.MAC (Joe/2138 011082) +++++
=====

Druckertreiber für Zweidraht-Quittungsbetrieb

;
; Anpassung eines Druckers mit Parallel-Übertragung an die
; PIO-Schnittstelle der ITT 3030 unter Benutzung der Signale
;
; STROBE (Aktiv Low, verlängerte Impuls-Standzeit)
; als Aufforderung zur Datenübernahme an den Drucker, und
;
; +++ BUSY (Aktiv High) als Drucker-Rückmeldung für fehlende Bereitschaft.
; (Adern für Acknowledge, Init, Demand und Error unbeschaltet)
;
; ACHTUNG: Zur Wahl der I/O-Basisadresse 0A0H am DIP-Schalter der PIO-Karte
; nicht, wie in der uns vorliegenden ITT-Dokumentation verlangt, die
; Teilschalter 1 und 3 in "ON-Position" (niedergedrückt) bringen,
; sondern statt ihrer die Teilschalter 2 und 4.
; (Fehler: Teilschalterbezeichnung Karte: 1..4, Dokumentation: 0..3)
;
; Assembler: Macro 80 (Microsoft)
;
title LST (Parallel-Drucker-Treiber für 3030/Hans-Georg Joepgen)
;
aseg ; Wir brauchen hier keinen relokierbaren Code
org 100H ; CP/M lässt uns keine andere Wahl. Apple-DOS
; wäre da insofern allemal flexibler.
; I/O - Basis-Adresse des PIO
basis equ 0a0h ; zugleich Steuer- und Statusregister Port B
bctrl equ basis + 0 ; Datenregister Port B (Hier: Quittungs-Signale)
bdata equ basis + 1 ; Steuer- und Statusregister Port A
actrl equ basis + 2 ; Datenregister Port A (Hier: Daten-Adern 0..7)
adata equ basis + 3 ; Betriebsart-Steuerbyte
mode3 equ 0cfH ; Datenrichtung: Adern 0..7 sind Ausgang
allout equ 0 ; Datenrichtung: Nur Ader 1 (Busy) ist Eingang
bctrlin equ 2 ; Ausgabebyte: Strobe Low (Aktiv) setzen
b1 equ 0 ; Ausgabebyte: Strobe High (Inaktiv) setzen
strobh equ 10H ; Startadresse serieller Original-LST-Treiber
moslst equ 0fe7aH

anzahl equ ende - treibr ; Blocklänge für späteres Blockkopieren

send macro was, wohin
ld a, low was
out (low wohin), a
endm

puls macro
nop ; Drucker zur Datenübernahme veranlassen:
send strobl, bdata ; Einschwingpause; Data vielleicht noch instabil.
nop ; Strobe-Leitung auf Low gleich Aktiv.
nop ; Hier schlagen wir für ein paar Millionstel
nop ; Sekunden die Zeit tot. Das gibt ein norm-
nop ; widrig langes Signal, bekommt aber dem
; Einschwingverhalten. Echo's stören nicht.
send strobh, bdata ; ... und die Strobe-Strippe wieder auf High.
endm

warten macro
local retour ; Wir fragen die Busy-Leitung ab und tun dies
retour: in a, (bdata) ; solange, bis sie mit LOW Drucker-
bit 1, a ; Bereitschaft anzeigt. Hier könnte eine
jr nz, retour ; Zeitschleife rein. Jetzt! Hängenbleiben,
endm ; wie oft "schöner Brauch" in CP/M.

ist: send mode3, actrl ; PIO definitionsgemäß programmieren.
send allout, actrl ; - mit Makros ganz schön übersichtlich.
send mode3, bctrl
send busyin, bctrl

ld hl, treibr ; Was wird kopiert?
ld de, (moslst) ; Wohin wird kopiert? (Mitten hinein ins MOS)
ld bc, anzahl ; Wieviel wird kopiert? (M80 hat's errechnet.)
ldir ; Und ab die Post! Das ist insoweit erst
; einmal alles.-

send 0, adata ; Alle Datenadern jetzt bitteschön auf Null -
puls ; Wir feuern einen ersten Impuls ab.
in a, (bdata) ; Was antwortet der Drucker -
bit 1, a ; hat er die Busy-Leitung auf High gesetzt?
jr z, fehler ; Nein! Dann soll's der Computer sagen.
; Ja! Nimmt er "Busy" später auch zurück?
; Dafür müssen wir ihm etwas Zeit lassen.
; Ein Byte kürzer als "LD A, 00".
xor a ; Jetzt lassen wir Zeit zum Reagieren.
pause: inc a ; Rücksprung, bis Akkustand erneut Null.
jr nz, pause ; Das reicht! Es waren immerhin 256 Zyklen.
; Busy vom Drucker nun brav nach Low gezogen?
in a, (bdata) ; Prüfen wir's mal.
bit 1, a ; Ja? Dann verabschiedet sich LST hier.
ret z ; Nein - dann ist Drucker aus, weg oder hin.-

fehler: ld de, auweia ; Da sollte man Alarm schlagen -
ld c, 9 ; was hier geschieht: Fehlermeldung auf Schirm,
call 5 ; vorgenommen durch CP/M höchstpersönlich.

weiter: warten ; Die Welt jetzt wieder in Ordnung, Busy Low?
; Wenn nicht, dann prüfen wir halt ad infinitum.
; Wenn ja! Dann adieu miteinander, den Rest macht
; TREIBER bei Bedarf, also bei Drucker-
; Anforderung, vom MOS aus. LST ist nun fertig
; und kann überschrieben werden.-
; Andernfalls sieht's nicht gut aus!
; Irgend etwas ist faul.-

auweia: db "Drucker leider " ; Wenn's schon Kummer gibt, dann soll der
db "nicht bereit.-" ; Computer wenigstens die Form wahren und
db 0dH, 0aH, "$" ; sich höflich mitteilen. Hier der Text.

treibr: di ; Hier wird - von MOS aus - endlich gedruckt.
ld a, c ; Zeichen kommt in C an, wir brauchen's in A.
out (adata), a ; Legen wir's mal auf Verdacht auf die Leitung.
puls ; ... und feuern einen Strobe-Impuls ab.
warten ; Hier bleiben wir, bis bereit ("Nicht-Busy").
; Alles bestens, der Drucker ließ uns laufen.
ei ; Geben wir also die Interrupts wieder frei
scf ; und sorgen für ein gelöschtes Carry-Flag,
ccf ; damit das MOS erfährt, hier ging alles
ret ; gut. So long bis demnächst.-

ende: end
```

Bild 5. Über diesen Drucker-Treiber lief das Manuskript des Beitrages, den Sie gerade lesen. Das Programm wurde umgeschrieben als Ersatz für die nur bedingt taugliche SEL-Software, die (als Assembler-Quelltext) mit den ersten Parallel-I/O-Karten für den ITT 3030 in Umlauf kam. Ausführlich abgedruckt wird »LST.MAC«, weil es auch für Benutzer anderer Mikrocomputer einiges an Anregung bietet.



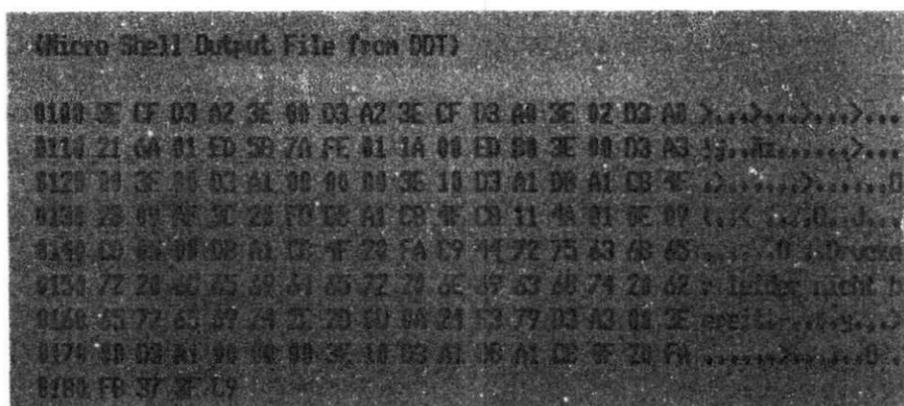
mal, das historisch aus dem elektromechanischen Fernschreiber und später aus Teletype-Maschinen entstandene serielle Verfahren bei Druckern. Und so ist auch der Standard-Druckeranschluß des ITT 3030 wie der vieler anderer Mikrocomputer heutzutage ein solches Unding. Für diesen Rechner gibt es jedoch auch eine Parallel-Karte, die schnell eingebaut ist. Ein Listing, in dem verzeichnet war, was man wie zu assemblieren hatte, um die Karte einzupassen, lag dem Interface bei. Doch geschrieben war das Listing durchaus nicht etwa in der Sprache des mit dem ITT 3030 gelieferten Assemblers ASM (der bekanntlich nur die Intel-Mnemonics für den 8080 versteht), sondern für den Z80 — und auch das ging aus den Papieren nicht hervor. Was da geboten wurde, konnte nicht nur niemals laufen, es ließ sich nicht einmal fehlerfrei assemblieren. Da hatte sich jemand entschlossen, ein Label DRV BGN (wohl Driver-Beginn) zu nennen, dann jedoch seine Meinung geändert und die Bezeichnung TREIB schöner gefunden. Auf TREIB wurde Bezug genommen, doch definiert war DRV BGN. Die Stellung der vier Schalter »Null bis Drei« wurde vorgeschrieben, auf der Karte gab es keinen Schalter Null, dafür aber einen im Papier nicht erwähnten Schalter Vier.

Mit Hilfe eines selbstgeschriebenen Treibers lief schließlich auch der Drucker

Also war der Schalter Vier diese Null? Beileibe nicht: Der Null war der Eins, der Eins war der Zwei und so fort bis zum Drei, der in Wahrheit der Vier war. Oder war nicht vielmehr der Vier der Drei? Also schauten wir ins Schaltbild, doch auf dem Schaltbild stand nur »Schaltbild«, sonst war nichts zu sehen außer unbedrucktem Papier.

SEL schickte schleunigst ein neues Papier, auf dem ebenfalls »Vorläufig« stand, und das mit Recht: Denn das Schalter-Desaster blieb, und ein neues Treiber-Listing hatte es in sich. Wieso und warum — um das zu erfassen, muß man wissen, daß bei paralleler Druckdatenübertragung außer Nebenvarianten hauptsächlich zwei Verfahren üblich sind: Die »Busy-Quittung« und die »Acknowledge-Quittung«. Übliche Drucker hatten bis vor ei-

Bild 6. Wer keinen Makro-Assembler benutzen mag, kann den Treiber »LST« diesem Hexausdruck entnehmen und auf Diskette sichern. Im Text steht, wie.



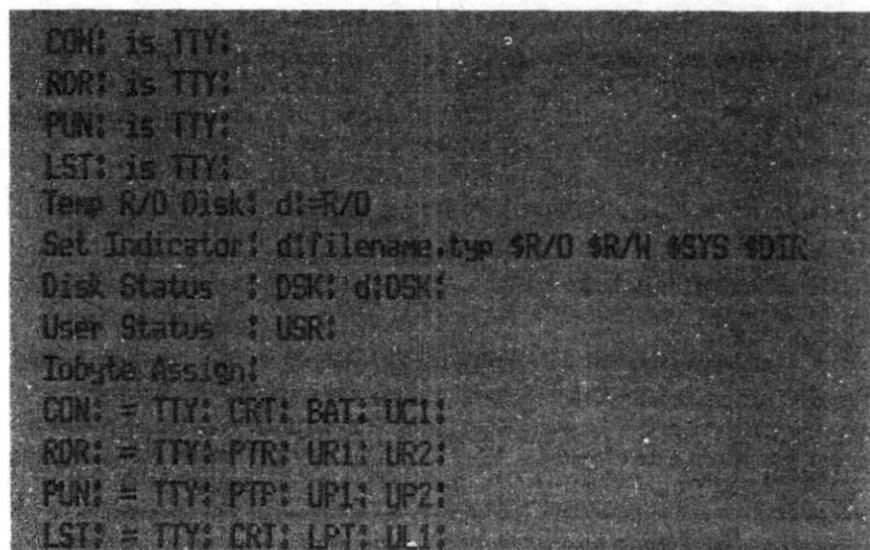
niger Zeit entweder »Busy« oder »Acknowledge«, Computer haben in der Regel »Busy« (semistatischer Pegel) oder Acknowledge (Pulsquittung) — Ausnahmen bestätigen die Regel. Damit ihre Drucker mit möglichst allen Computern ohne Änderung oder Umschaltung zusammenarbeiten können, rüsten nun seit einiger Zeit vor allem japanische Hersteller ihre Printer so aus, daß sie sowohl ein »Busy«- als auch ein »Acknowledge-Signal« abgeben, etwa nach dem Motto: Was der Computer nicht will, braucht er auch nicht abzufragen. Einen solchen Drucker hatten die Verfasser des neuen Treibers wohl in Händen, als sie ihr Werk schufen — der Treiber wertete beides nacheinander aus, Busy und Acknowledge, und fehlte eines von beiden Signalen, dann lief nichts mehr. Erfolg: Drucker, die nur eines der beiden Signale abgaben (mehr braucht man im Normalfall auch niemals), waren ebenso weg vom Fenster wie solche Drucker, die man wahlweise auf »ACK« oder »BSY« umschalten kann. Lassen wir den Zwischenakt weg, in dem es um die Beschaffung des richtigen Kabels (zarter Pfostenstecker für einen Peripherie-Anschluß) oder um invertierte Adreßbits beim PIO ging (für den Rest der Welt gehört die Basis-

adresse eines Z80-PIO üblicherweise zum Port A, bei dem ITT 3030 dagegen ist sie dem Port B zugeordnet, weil ein Bustreiber aus unerfindlichen Gründen Adreßbits »umdreht«) — kommen wir zum glücklichen Ausgang: Das Manuskript dieses Beitrages wurde parallel von dem ITT 3030 auf einem Centronics 737-2-Drucker ausgegeben. Acknowledge und INIT und ERROR hängen in der Luft: Und es geht trotzdem prächtig — mit Hilfe eines hausgemachten Treibers, der in Bild 5 aufgelistet ist.

Wer keinen Makro-Assembler hat, kann auch die Zeichen aus dem Hexlisting des Bildes 6 mit Hilfe von DDT eintippen und dann nach Control-C durch den Befehl SAVE 1 LST.COM auf Diskette schreiben. Der Treiber schreibt einige Bytes in das MOS und fortan funktioniert die Karte auch bei längst überschriebenem LST-Programm — solange kein neuer Kaltstart vorgenommen wird.

Nebenbei bemerkt: Die PIO-Karte ist nicht nur für das Drucken nützlich; sie wurde höchst erfolgreich auch bei Hochgeschwindigkeitsübertragungen im Dialog mit anderen Computern, darunter eine Oasis-Maschine, erprobt. Für solche Transfers ist die Karte auch wirklich notwendig: Die serielle

Bild 7. Alle Kanäle angeblich dem »Teletype-Device« zugeordnet, doch Reader und Lister arbeiten gleichwohl auf der Seriell-Schnittstelle: Da stimmt was nicht mit dem CP/M der ITT 3030. Man sieht die Rechner-Antwort auf



der Eingabe »STAT DEV:«. Schon ein bißchen schizophren: »STAT VAL:« bietet Wahlmöglichkeiten an, vor denen »STAT DEV:« die Augen verschließt. DOS und MOS arbeiteten bei unserer Betriebssystem-Vorabversion nicht gerade harmonisch zusammen.



Standard-Schnittstelle kann laut Dokumentation nur Daten bis 4800 Baud (gegen 500 Byte pro Sekunde) verarbeiten. Zum Vergleich: Wenn die hochbetagte 2020 mit Hilfe eines für 17 Mark erstandenen Interface an einen CS-2000 Daten sendet, dann geschieht das mit genau der doppelten Geschwindigkeit. Schließen wir die Aufzählung der aufgefallenen Schwachpunkte mit der Beobachtung ab, daß der ITT 3030 das CP/M-IOByte offenbar merkwürdig (oder vielleicht sogar gar nicht?) behandelt. Die Kanäle RDR: und LST: sind nach dem Neustart von 5.2 DT einwandfrei dem Combo-Bauteil zugewiesen, das für Standard-I/O zuständig ist, doch die Auswertung durch STAT mit Argument DEV: ergibt anderes (Bild 7). Digital Research bezeichnet das IOByte-Handling ausdrücklich als fakultativ (und spricht dabei vornehm vom »Intel-IOByte«), doch ist eigentlich kein Bios-Autor bekannt, der es heute noch wagen würde, die IOByte-Konvention zu mißachten. Auch Software-Produzenten bauen zunehmend fest auf das besagte IOByte: Immer mehr

Endkunden-Programme bedienen sich dieses eleganten Kanalumschalters.

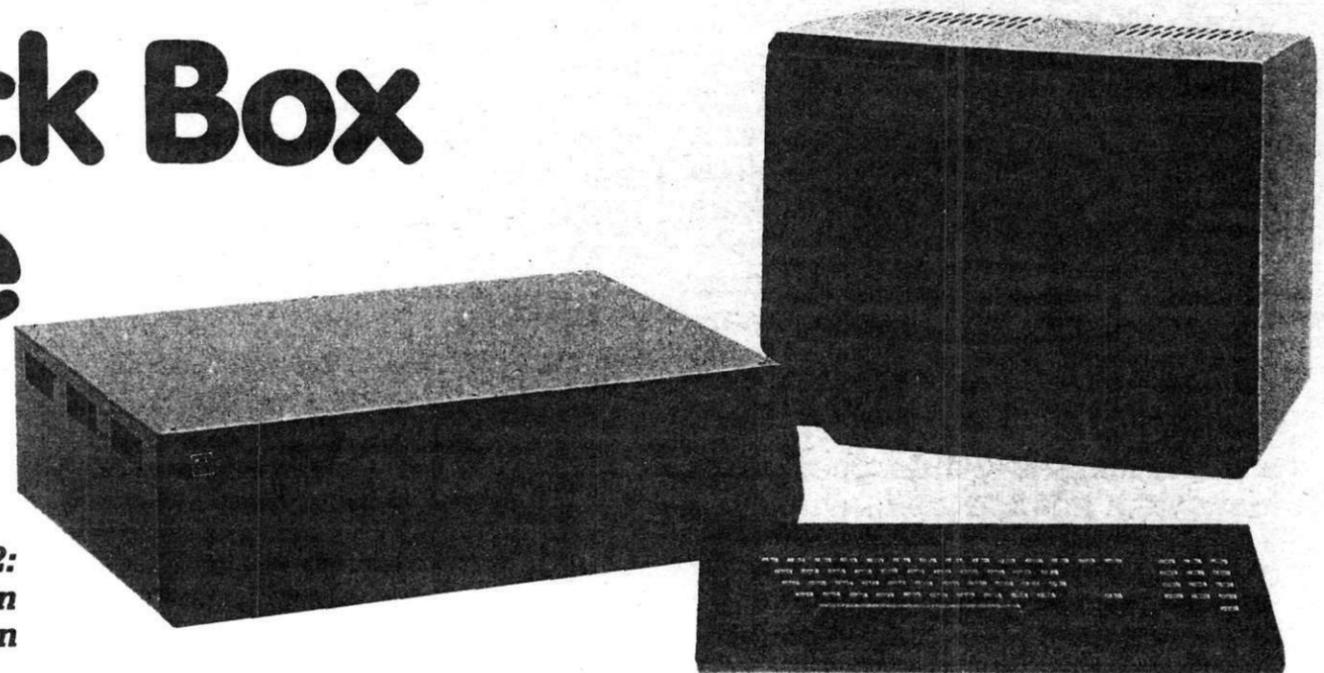
Schlußurteil: Eine Maschine mit Zukunft

Wir haben, dies als Fazit, an unseren Erprobungs-Voraussexemplaren von Computer und Betriebssystem einiges an Mängel gefunden: Nicht nur Dinge, die mit dem persönlichen Geschmack des Benutzers zusammenhängen, sondern auch Eigenschaften, die eher Begriffe wie Stand der Technik und das Ansehen einer Firma mit Weltgeltung berühren. Doch vieles von dem, das uns auffiel, ist damit zu erklären, daß wir kein fertiges Handelsprodukt erprobten, sondern aus Aktualitätsgründen mit einer Betriebssystem-Prototypversion arbeiteten, die sich noch im Feldtest befand. Stellt man dies in Rechnung, so ergibt sich alles in allem ein deutlich positives Bild: Die Maschine zeigte sich (bei seitwärts daneben aufgestelltem Monitor) als unbedingt zuverlässig, wir empfanden sie als handlich, hübsch und ausgesprochen sympathisch. Nach

einer ersten Phase des mißtrauischen Beschnepperns, wobei das Mißtrauen wegen der anfänglichen Systemzusammenbrüche durch Monitor-Störstrahlung bis zur Klärung des Sachverhaltes natürlich erst einmal vertieft wurde, faßte man Vertrauen und wagte es schließlich, dem ITT 3030 auch mal die Früchte tagelanger Arbeit anzuvertrauen. Ich bin in keinem Fall enttäuscht worden. Wenn die angekündigten Neuerungen wie ladbarer Zeichensatz und Bank-Switching unter einem neuen CP/M verfügbar sein werden, die Serienmaschine ein wenig leiser und das Tastenfeld-Layout den Bedürfnissen von Pascal und CP/M noch ein wenig besser angepaßt und die »Caps-Lock-Taste« nicht vergessen wird, dann wird — ordentliches Marketing und attraktive Preise vorausgesetzt — der ITT 3030 ein Computer sein, der sicherlich Zukunft hat und auch einen beachtlichen Platz in den Verkaufslisten wird einnehmen können.

(Hans-Georg Joepgen)

Die Black Box in beige



**Testobjekt ICL 32:
Prinzip der verteilten
Intelligenz unterstützen**

Die Liste der Anbieter von Kompakt- und Kleinrechnern nimmt ständig zu, und ein Newcomer hat es gar nicht leicht, dem interessierten Betrachter ins Auge zu fallen. (Am besten geht es noch mit einem 132spaltigen Bildschirmformat, einer 32-Bit-Rechnerarchitektur und einem Festplattenspeicher zu Floppy-Disk-Preisen.) Oder man schenkt dem Hersteller schon aufgrund seines Rufes die entsprechende Aufmerksamkeit — Beispiele sind da unter anderem IBM oder ITT. Zu dieser Gruppe kann man auch ICL, einen der wichtigsten europäischen Hersteller von Computersystemen, zählen. Seit etwa einem halben Jahr bietet dieses Unternehmen seine Personal Computer der Serie 10 und 30 an. Getestet wurde das Modell 32 mit einer integrierten 5-MByte-Winchester-Platte und einem 5¼-Zoll-Floppy-Disk-Laufwerk. ▶