C3PO

VOL 3 NO 5
JUNE 1980

NOTICE IS HEREBY GIVEN OF THE ANNUAL GENERAL MEETING OF THE CAPE COMPUTER CLUB TO BE HELD ON WEDNESDAY, 04 JUNE 1980, AT 20H00 IN THE MAIN HALL, HERZLIA HIGH SCHOOL.

AGENDA

1. APOLOGIES
2. CONFIRMATION OF MINUTES OF 1979 AGM
3. CHAIRMAN'S REPORT
4. TREASURER'S REPORT
5. ELECTION OF OFFICE BEARERS
6. GENERAL

REFRESHMENTS: CHEESE AND WINE

OFFICE BEARERS WHO MUST AUTOMATICALLY STAND DOWN: ROB BERNSTEIN, ALAN PARRY, DONALD COOK.

WE URGE YOU TO ATTEND TO NOMINATE THOSE WHO YOU FEEL YOU WOULD LIKE TO RUN YOUR CLUB.

Cape Computer Club Printout

P O BOX 6829, ROGGEBAAI, 8012

CHAIRMAN/SOFTWARE	:	ROB BERNSTEIN	: 72-9418
VICE CHAIRMAN/ADVERTISING/LIBRARY	:	ALAN PARRY	: 72-4773
SECRETARY	:	COLIN RUDOLPH	: 53-1460
TREASURER/SUBSCRIPTIONS	:	NEIL WALSH	: 65-7415
PUBLICATIONS/C3PO/Z80 GROUP	:	DONALD COOK	: 46-3615
COMPONENTS/HARDWARE	:	NIKKI PARKYN	: 66-8464
6800 GROUP	:	PIERRE WILTER	: 55-4176

Contents

VOL 3 NO 5 JUNE 1980

Editorial

Rob Bernstein

ON RELIABILITY

A worrying aspect of computers today, and this applies also to larger more expensive machines as well as micros, is their reliability.

What makes a machine reliable? Initially, obviously good design, the use of components well within their specification ranges, well tried and tested techniques.

All of the above one takes for granted. But no matter how well one builds a machine one must accept that components do fail, that mains power fluctuations do take place, that operators do press the wrong buttons or keys and nobody ever reads instruction manuals!

All the above accepted, what then? Well, if something is going wrong, the minimum that can take place is to detect the situation. This being done, one can proceed to find the problem.

When one looks at the micro end of the market one finds that NONE of the more popular brands of machine have memory parity checking, let alone memory error correction circuitry. So what? Well, I think that all of us who make regular use of these machines find that a program crashes for no good reason and upon rerunning works correctly. Well, one cause could be a dropped or picked-up bit in memory - without some way of detecting the error you will never have any way of finding out what went wrong.

Some machines, particularly those with floppy disks, proceed to get very sick when power fluctuations take place. The inclusion of a constant voltage transformer together with transient suppressors often works wonders.

I have recently heard of a problem occurring on a large mini/mainframe where, due to a component failure, a disk drive proceeded to randomly add 2 bits to data written. As the disk was used in a large data base, it sometimes took more than a couple of months before the data was again accessed and an error found!

Another problem, one which can be subtle in appearance but devastating in action, is the ability of lack thereof of checking the checking circuitry/procedures - a simple case and one which one comes across every day - when an error is detected, light a lamp to tell somebody. How do you know if the lamp is working? The fact that it is not burning may be due to the fact that the error warning lamp is broken rather than the fact that there is no error! Two simple solutions spring to mind:

(1) That all lamps are lit on a regular basis, i.e. once per minute.
(2) That the lamp is ON when NO error exists and that it goes out when an error does exist!

How often have you seen this implementation?

All of the above pertains to hardware, but the same can be said of software. Many programs written have insufficient error detection procedures, but how many of those that include the procedures have been thoroughly tested to check that the error procedures are in fact working?

I hope that all of us involved with computers will give the above comments some thought as reliability of operation is the most important aspect of computing.

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

A.G.M.
WEDNESAY
04 JUNE 1980

C3PO Scratchpad

Colin Rudolph

Funny stories involving computers abound. Maybe this is because many people are faintly afraid of them. Anyway here are two from my collection.

The first is apparantly a true story and was reported in the Los Angeles Times from a UPI release.

Joseph Begley saved 2000 cigarette coupons and posted them to a British cigarette company in exchange for a watch. When the watch did not arrive he wrote to them and asked them why. Back came three watches. Begley wanted only one so he sent the other two back. Next day 10 parcels arrived. The following day 18. The day after that the post office 'phoned to say there were another 10 parcels waiting. All of them were trade-in gifts given by the cigarette company in exchange for coupons - coupons Begley never had. Among the gifts were three tape recorders, a doll, a golf bag, two electric blankets, a cot, pots, a pressure cooker and long playing records. Begley sat down and wrote a long, pleading letter to the firm, asking it to stop. Back came the reply saying, "It was a computer error." The company gave Begley 10000 coupons in compensation for his troubles. With these Begley ordered tools and a bedspread. He received a plant stand and two stepladders.

The second story may be aprocryphal but seems to me to be the ultimate in computer confrontation.

A computer was programmed to report its own malfunction to a service engineer. The equipment went on the blink one night and duly called the service engineer, who had moved. The telephone company's computer responded with a recorded message that the telephone at that number was disconnected. The computer broke the circuit, redialed the disconected number and reported the malfunction. The telephone replied with the recorded message. The computer broke the circuit, redialed.....

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

Book Reviews

Dataweek, April 1980

"PROGRAMMING THE Z80"

by Rodney Zaks, Sybex Books, 1979, $14,95

Rodney Zaks is known internationally for his teaching and publications in the field of microprocessors, and his experience is clearly reflected in this most valuable text. While almost 200 pages are merely a reprint (if far more clearly presented) of the manufacturers' data sheets, the text does provide the user of the Z80 with a complete reference work which takes him from those first fumbling steps in programming right through to the handling of fairly complex data structures.

The format of the book is well-arranged: starting from basic programming concepts, it goes on to emphasise the hardware organisation from the programmer's point-of-view. These details are often hidden in the manufacturer's data sheets, in between lengthy discourses on timing, number of cycles, etc - information useful to the hardware designer, but generally irrelevant to an understanding of the programming.

Basic programming techniques are covered, and then the full instruction set of the Z80 is studied. The following sections get down to addressing techniques and I/O programming, and then comes the crux - many useful application examples. This is the part of most value - most users are constantly on the lookout for ways of tackling different problems, and well documented examples are of immense value.

The section on data structure is well written and most readable. It explains many concepts which are well known to the computer scientist, but not readily known to the microprocessor user. Zaks has succeeded in really simplifying many of these concepts and showing their uses. Following a look at ideas on programming development and support systems, come the almost statutory appendices listing conversion tables, lists of codes, etc.

This book will, without a doubt, prove to be of the greatest value to the user of the Z80. It presents information well beyond that normally given by the manufacturers, and will largely replace their manuals. At a relatively low price, it is a must for Z80 users.

Address Decoding

Alan Day

Address decoding is one of those areas in microprocessors that gives the beginner sleepless nights at home and sleepless days at work. Last year I went through the difficulties of trying to understand this great mystery.

My explanation of the decoding is based on a 16K, single card system that I built recently. The chip used for the decoding was an Intel 8205 (at about R4) purchased before I discovered that the 74 LS 138 (at about 70 cents) is pin for pin compatible.

The method I used is as follows:
Say, for example, that you wish to start your random access memory (RAM) at address 2800_{HEX}, write 2800 in the four columns as shown in step 1 (Table 1). Next, decode each HEX digit into a 4 bit binary number as in step 2.

If you wish to have some read only memory (ROM) at 3000_{HEX}, the code would be as shown in Table 2.

The 74 LS 138 is a 3 to 8 line decoder that has the truth table shown in Table 3. (Note that the chip selects are active low.)

TABLE 3

ADDRESS			CHIP SELECT NUMBER							
A0	A1	A2	0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

Consider what would happen if you assigned Ao, A1 and A2 of the 74 LS 138 to A13, A12 and A11 (of your address bus) respectively.

You can now see, at a glance, that your memory map will be as shown in Table 4.

TABLE 4

CHIP SELECT	MAP
0	0000_{HEX}
1	0800 "
2	1000 "
3	1800 "
4	2000 "
5	2800 "
6	3000 "
7	3800 "

Note that each block can contain up to 2048 (2K) bytes, a reasonable map for a single card system.

Note also that chip select "0" is from 0000 to $07FF_{HEX}$. "1" is from 0800 to $0FFF_{HEX}$, etc.

TABLE 1

ADDRESS LINE	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
STEP 1			2				8				0				0	
STEP 2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

TABLE 2

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		3				0				0				0	
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

WRAPAROUND (OR FOLDBACK)

This used to scare the hell out of me. Quite simply wraparound occurs when you "overstep" your address decoding. As long as you stick to your memory map it can do no harm at all.

Take as an example the memory map that we have just worked out. Look at Table 1 and put 4000 into it as shown in the previous examples. Now assign A0, A1 and A2 (74 LS 138) as before. Notice how you find that chip select "0" ($\overline{CS\ 0}$) is active again! It will also be active for addresses 8000_{HEX} to $87FF_{HEX}$ and $C000_{HEX}$ to $C7FF_{HEX}$. In other words you can access the same memory at four different addresses. This is where the only danger lies.

If you have RAM at say $002F_{HEX}$ and you write data to $402F_{HEX}$, $802F_{HEX}$ or $C02F_{HEX}$ you could corrupt the contents of location $002F_{HEX}$. However, as I have said before, no harm can be done if you stick to your memory map.

Foldback can also occur within each 2K block. If for example a 1K chip is configured from 0000_{HEX} to $03FF_{HEX}$, it will also reside from 0400_{HEX} to $07FF_{HEX}$.

Note that unless a more complex method of address decoding is adopted the foldback areas are wasted.

We now have address decoding for 16K of memory (which is ample for a single card system of this nature - ask Intel, they know!)

PORT MAPPING

Most 8 bit CPUs address 256 ports using the lower 8 bits of the address bus.

A "port map" may be created in the same manner as the memory map already described to allow for the decoding of the ports.

To prevent "overlapping" of ports and memory, a separate decoder (74 LS 138) is used for the ports.

If you get hold of a 74 LS 138 (8205 or 3205) data sheet, you will see that the truth table has more to it than I have thus far let on. The chip has 3 enable lines. You will, for the basic system, only use one of these (E_1) on each decoder. E_2 will be tied low and E_3 will be tied high. E_1 on your memory decoder is taken to the $\overline{\text{MEMORY REQUEST}}$ line and E_1 on your port decoder is taken to the $\overline{\text{I/O REQUEST}}$ line. (See Figure 1.)

It can now be seen that when addressing memory, the port address decoder is disabled ($\overline{IORQ}$ high) and when addressing ports the memory address decoder is disabled ($\overline{MREQ}$ high). That's just about all there is to it!

DUEL FUNCTION CHIPS

One last thing I would like to describe is how to decode for a chip that contains (as an example) RAM and ports. This is really very easy. Take the $\overline{CS}$ for your RAM (from memory decoder) and the $\overline{CS}$ for your port (from port decoder), AND them together and connect to your chip as shown in Figure 2.

I hope that this article has been of some use to the budding enthusiast, even if it is just used for making paper jets or starting fires. Good luck.

GUIDE TABLE AND PORT MAP FOR FIGURE 1

A7	A6	A5	A4	A3	A2	A1	A0	HEX	CHIP(PORT)	SELECT AND MAP
0	0	0	0	0	0	0	0	00	$\overline{CS0}$	00 to $0F_{HEX}$
0	0	0	1	0	0	0	0	10	$\overline{CS1}$	10 to 1F "
0	0	1	0	0	0	0	0	20	$\overline{CS2}$	20 to 2F "
0	0	1	1	0	0	0	0	30	$\overline{CS3}$	30 to 3F "
0	1	0	0	0	0	0	0	40	$\overline{CS4}$	40 to 4F "
0	1	0	1	0	0	0	0	50	$\overline{CS5}$	50 to 5F "
0	1	1	0	0	0	0	0	60	$\overline{CS6}$	60 to 6F "
0	1	1	1	0	0	0	0	70	$\overline{CS7}$	70 to 7F "

FIGURE 1

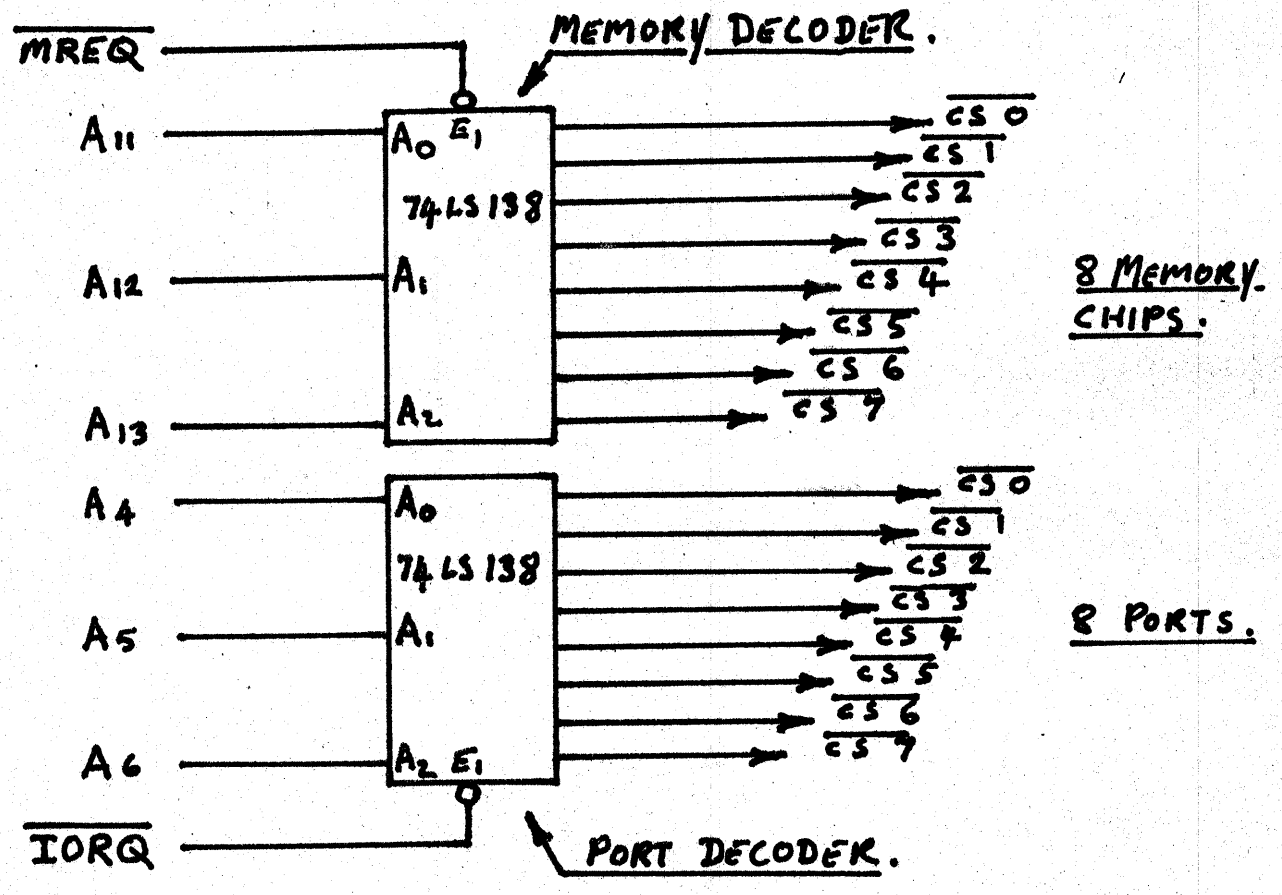
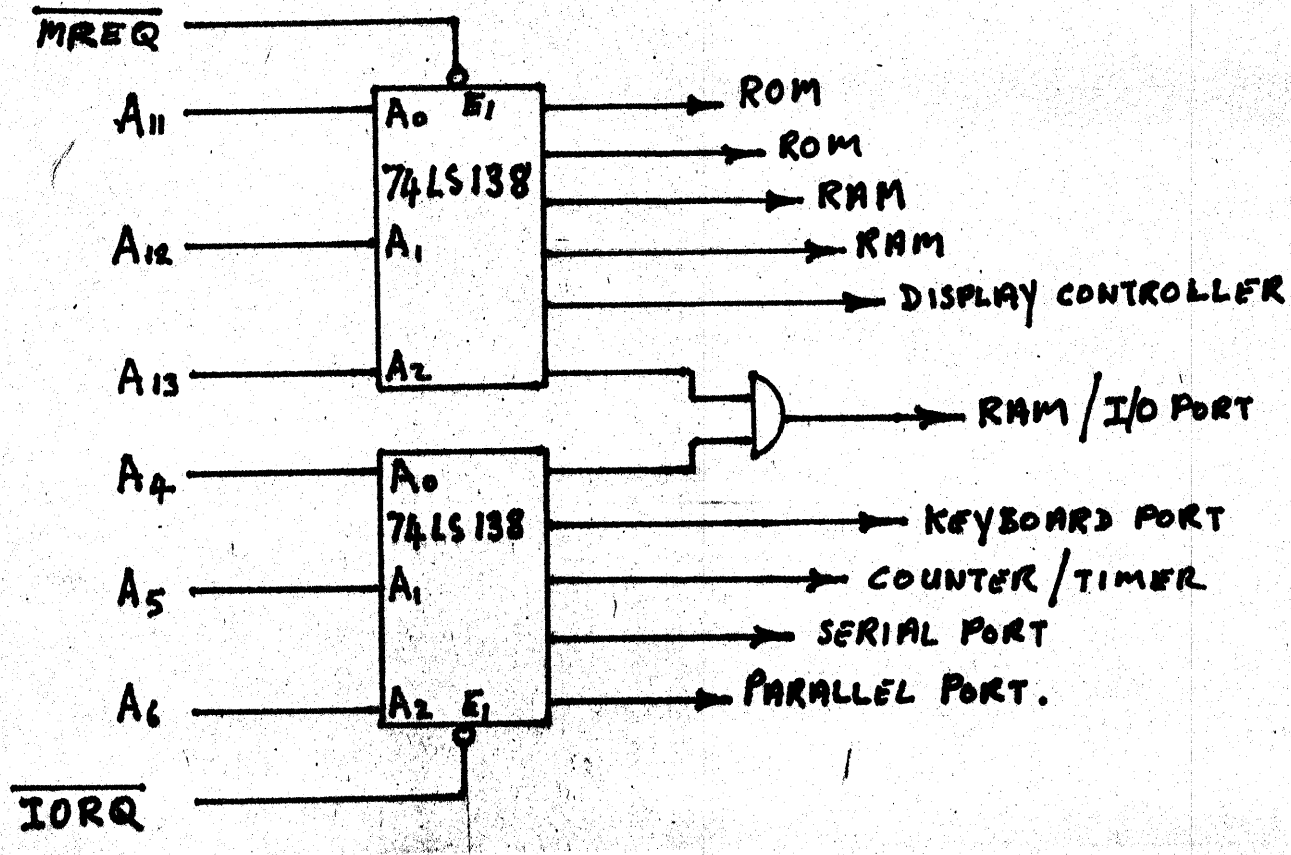


FIGURE 2



JUMP ON RESET [280 & 8080/85]

Ivan Kohler

Firstly, I would like to credit Mr K.J. Sterling for the original design and hope that he won't mind sharing its usefulness to some of us.

This circuit is a very useful one, allowing the programmer to jump anywhere within 256 bytes on $\overline{\text{RESET}}$.

It is essential to systems that have a monitor higher up on the address map.

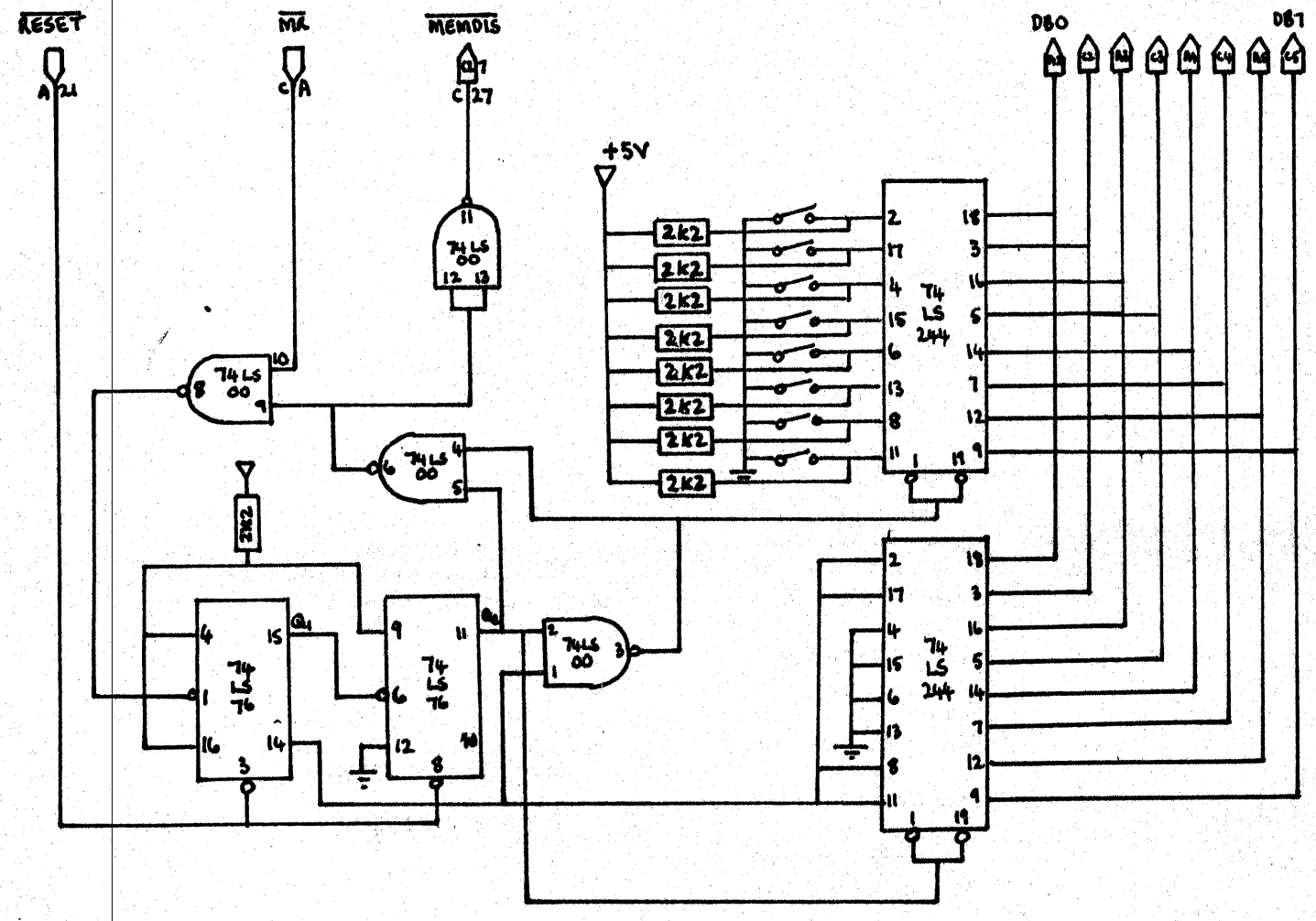
A prerequisite is that the card living at zero address must have its $\overline{\text{MEMDIS}}$ line implemented.

The circuit as shown in the diagram consists of two flip flops which give a possible 4 unique states. On $\overline{\text{RESET}}$ three of the states are sequentially executed, each state being synchronised by the $\overline{\text{MR}}$ from the CPU.

During this time $\overline{\text{MEMDIS}}$ is held low forcing all cards with $\overline{\text{MEMDIS}}$ implemented into the high impedance state. The first state causes a C3 to be placed on the data bus in answer to a $\overline{\text{MR}}$. The second state causes a 00 on the bus and the third state places the contents of the SWITCH REGISTER on the data bus.

Thus performing the instruction:
JP SS00 H (Z80)
JMP SS00 H (8085)
where SS can be anything from 00 to FF.

Having RAM from zero upwards allows one access to all the software (RST) restart instructions and with the 8085 all the hardware interrupts RST 5.5, 6.5 and 7.5 jump addresses.



JUMP ON RESET TO A SWITCH SELECTABLE ADDRESS (8080/85 & Z80)

SOFTWARE CORNER

The program listed below (which speaks for itself) was written by Davin Milun, a Standard 6 student at Herzlia High School, in Tiny Basic (2K). Davin and his friends have amused themselves writing short routines. Lack of suitable recording facilities have dictated that the routines are short. We are, however, running out of ideas and suggestions would be welcome. Please submit your ideas to C3PO. Perhaps you have your own program which you wouldn't mind sharing with members.

```
OK
>L.
   9 P. "I HAVE CHOSEN A RANDOM NUMBER . THIS PROGRAM ALLOWS YOU "
   10 P."TO GUESS THAT NUMBER WHICH IS BETWEEN THE TWO NUMBERS "
   20 P." SHOWN ON THE PAPER . YOU HAVE SIX TURNS TO GUESS"
   30 P."IT IN."
  40 A=RND (99); Z=0
  50 B=100 ;LET C=0
  55 FOR Q=1 TO 6
  60 P. B,"( )",C
  70 IN."YOUR GUES",S;Z=Z+1
  80 IF S=A P. "WELL DONE YOU DID IT IN ",Z,"GOES";G. 140
   90 IF S>=BP."REREAD INSTRUCTIONS";G.70
 100 IFS<=CP." GUESS ONLY A NUMBER BETWEEN THE OTHER TWO";G.70
 110 IF S>A LET B=S
 120 IF S<A LET C=S
 130 NEXT Q
 135 P. "BAD LUCK THE NUMBER WAS ",A
 140 IN."DO YOU WANT ANOTHER CHANC",E
 150 IF E=N S.
 160 IF E=Y G.40
 170 S.

OK
>

 R.
I HAVE CHOSEN A RANDOM NUMBER . THIS PROGRAM ALLOWS YOU
 TO GUESS THAT NUMBER WHICH IS BETWEEN THE TWO NUMBERS
  SHOWN ON THE PAPER . YOU HAVE SIX TURNS TO GUESS
 IT IN.
   100( )      0
YOUR GUESS:23
   100( )     23
YOUR GUESS:42
   100( )     42
YOUR GUESS:75
    75( )     42
YOUR GUESS:56
WELL DONE YOU DID IT IN       4GOES
D YOU WANT ANOTHER CHANCE:Y
   100( )      0
YOUR GUESS:48
   100( )     48
YOUR GUESS:78
    78( )     48
YOUR GUESS:59
    78( )     59
YOUR GUESS:69
    69( )     59
YOUR GUESS:60
    69( )     60
YOUR GUESS:65
BAD LUCK THE NUMBER WAS      67
```

SPOT THE BUG

Nikki Parkyn

Since BASIC has no standards most implementations, though they possess the same instructions, do not necessarily produce the same results.

Execution of the code below on different interpreters may produce totally different answers, though on the author's one it produces the correct answer. With modification to one statement it would always produce the same answer. See if you can find out why - the modification will be printed in next month's C3PO.

```
10 DIM A(3)
20 LET A(1) = 123
30 LET A(2) = 456
40 LET A(3) = 567
50 LET H = 3,99
60 LET I = H - 1
70 PRINT A(I)
80 END
```

PUZZLE

<u>DID THE BUTLER DO IT?</u>

"Where are those valuable Indian-head pennies I left on the table this morning, James? I put them in a square array and now there are only two left? You didn't take them, did you?"
"Well, sir," replied the butler, "shortly after you left, three burglars came in. They divided the pennies equally among themselves, but left these two because they could not divide them equally."
Is James telling the truth?

Write a program in BASIC 6800 or Z80 code to solve this puzzle. Your program should be well documented and be accompanied by a flow chart.

A R5,00 cash prize is offered to the best solution. The winner will be announced in the August issue of C3PO. The Editor's decision is final.

UHF / UHF TV Modulator on Veroboard

Geoff Sturges

The layout shown opposite is based on the circuit as described in Elecktor in October, 1978.

Your rationale for putting this circuit onto Veroboard could be for one or more of four reasons:-

1. A PCB may not be available off the shelf.

2. An off the shelf PCB may be too expensive.

3. You may not be able to make your own PCB.

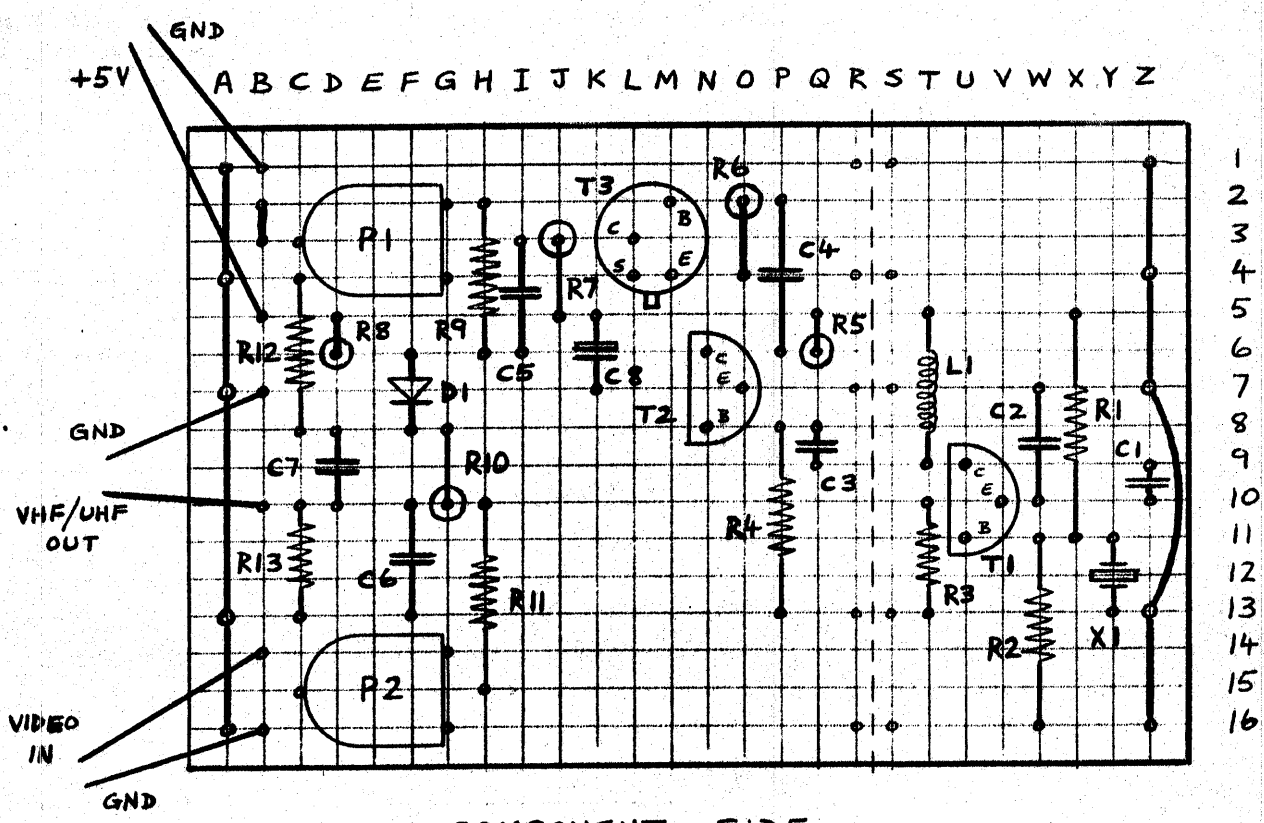
4. Your first few attempts at using a bought PCB ended in failure, as did mine.

The onboard regulator has been omitted as you should have got a +5V supply by now!

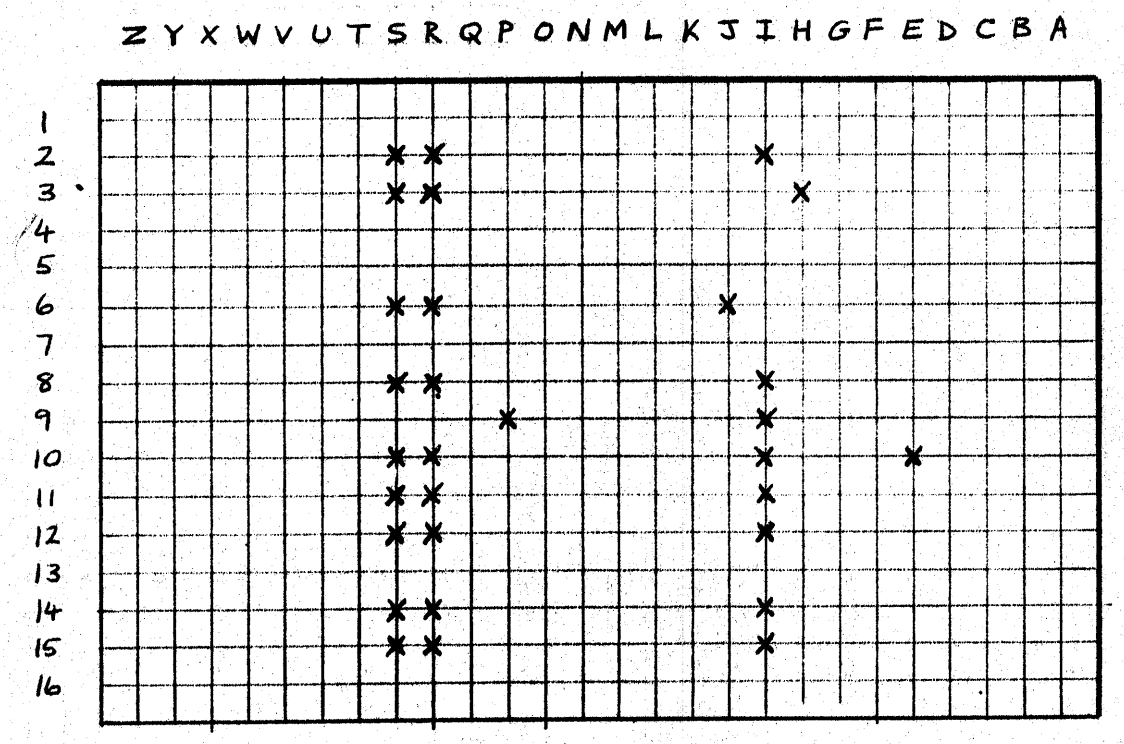
A piece of sheet metal approx. 15 x 50 mm is placed vertically between columns R and S and is held in place by four copper wire staples soldered to the Ground strips as shown.

The completed board must be kept in a metal box such as is sold by the Electronics Supermarket for R4,00, otherwise you may get a visit from Fred!

I used a 27,005 MHz crystal. I haven't tried using just a capacitor, and don't intend doing so, now that it works! Don't be put off by the guy at Hamrad explaining about the crystal only being a third harmonic of the frequency, or something. I thought harmonics were like big mouth organs!



COMPONENT SIDE

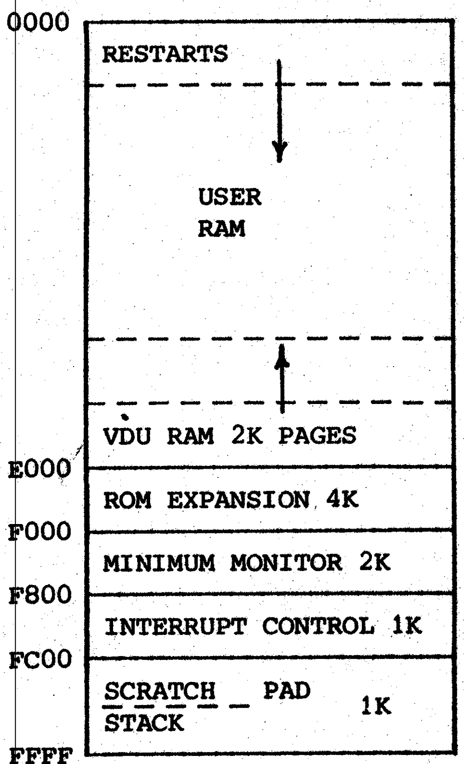


SOLDER SIDE — TRACK BREAKS

Group Report

Donald Cook

Well, after finding Ivan's pad (not easy) we got down to a successful idea swopping session. It had become obvious that a number of members of the group are considering memory mapped VDU's and as such, provision should be made for this in the memory map and the monitor. It was agreed that since the restarts, all 8 of them, reside in the first 100_H locations of memory and also, since they provide useful programming aids, that a "Jump on Reset" should be implemented (See the article elsewhere in this issue.) together with the following memory map:



Ivan Kohler explained the tape I/O system that he has b-en using with great success at 9600 baud; the system is saturated recording off the TTL signal - this makes use of an increased tape speed 4 times and a very simple op amp interface with the micro. Most people were very keen on this system; the reservation being that of compatibility - Nikki and Ivan are going to experiment further.

The alternative suggestion by Alan Day is Manchester code. This of course could be a completely software orientated system. Being self clocking would be much more versatile. It was decided to liase with the 6800 Group on this matter since it seemed important to have inter-group compatibility.

QUICKIE

Answer to last month's quickie:
18 — 81
Sneaky, eh?

How about this one:
A tramp (name omitted to protect innocent members of the public) rolls cigarettes from butts he picks up. He finds that four butts make one new cigarette. How many cigarettes can he smoke from a haul of sixteen butts?

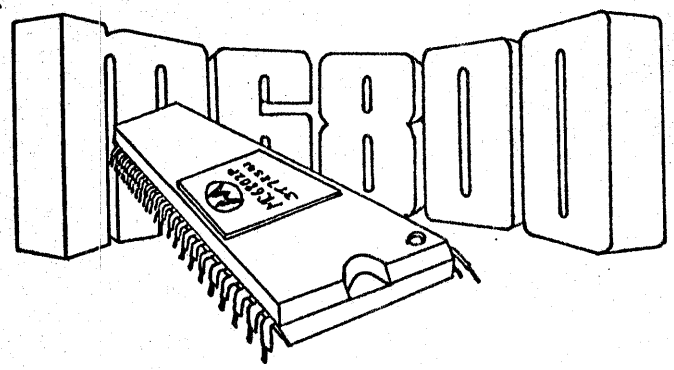
Dumpout

INTERRUPT: A hardware based facility which allows the suspension of a current program while and alternative "Interrupt Handler" program is executed. At the end of the interrupt sequence the original processor status is restored and the previously executing program is allowed to continue from the point at which it was interrupted. Interrupts are a powerful and widely used tool for the handling of peripheral input/output transfers.

INTERRUPT VECTOR: This is the address at which the start of an "interrupt Handler" program will be found. Some micro-processors used fixed interrupt vectors set by the chip manufacturers, while more sophisticated systems with many possible interrupt sources allow the interrupting device to provide its own vector via an input port.

LABEL: A string of characters used instead of absolute addresses in programs. These are usually alphanumeric.

Hear the latest in computerese?
byte : word
nyble: ½ byte
chomp: nyble + byte



MICROPHILE

This being the issue of the AGM, I would also like to add a few words to the fray. The past year has seen an interesting development - whilst the Club meetings have received dwindling support and enthusiasm, the special interest groups (or SIGs as the perpetrators of computerese would have it) have almost suffered from boundless enthusiasm. I think that this is a clear indication of a trend developing, and although being a fervent advocator of participation in SIGs, I still see the role of the Club at large being of paramount importance: not only to co-ordinate the activities of the SIGs, but indeed to bring them together, and to impart to members of SIGs an eclecticism of approach which is so necessary in a field where over-specialisation can so easily occur, with nothing but deterimental effects (strains of the medical profession ?!).

So endeth this epistle with an impassioned plea -

VIVA LA Cape Computer Club!
VIVA LA membership drive!
VIVA LA SIGs!
The struggle continues!

SOFTWARE SECTION

Unfortunately, space does not permit me to publish the 6800 Group's Memory Test. I am publishing, however, a BASIC program found by Anthony Rose:

```
6800 MICROSYSTEM
>LIST

010 PRINT "THIS PROGRAM COMPUT
    ES WHAT DAY OF THE WEEK TH
    E DATE YOU ENTER IS"
030 J$(1)="SUNDAY"
040 J$(2)="MONDAY"
050 J$(3)="TUESDAY"
060 J$(4)="WEDNESDAY"
070 J$(5)="THURSDAY"
080 J$(6)="FRIDAY"
090 J$(7)="SATURDAY"
095 PRINT
100 PRINT "ENTER D,M,Y"
110 INPUT D, M, Y
120 IF Y>1752 THEN 150
125 PRINT
130 PRINT "NOT BEFORE 1752 PLE
    ASE"
140 GOTO 100
150 K=INT((1/M)+0.6)
160 L= Y - K
170 O = M + 12 * K
180 P = L/100
190 Z1 - INT(P/4)
200 Z2 = INT(P)
210 Z3 = INT((5*L)/4)
220 Z4 = INT(13*(0+1)/5)
230 Z = Z4 + Z3 - Z2 + Z1 + D-
    1
240 Z = Z - (7*INT(Z/7)) + 1
245 PRINT
250 PRINT "THE DAY OF THE WEEK
    IS ";J$(Z)
260 PRINT
270 PRINT "AGAIN ? (Y/N) >"
280 INPUT L$
290 IF L$="Y" THEN 050
300 PRINT "THE DAY OF THE WEEK
     PROGRAM NOW SAYS BYE"
999 END

>BYE
```

BUYLINES

Motorola USA have introduced an evaluation module for the 68000 (along the same lines as the D2 et al) and it includes 32K PROM/RAM, two 16-bit PIAs, three 16-bit PTMs, and two serial RS232 ports. It apparently can be used either stand-alone, or in its own card-cage with 6800 memory, or in an Exorciser system. Price ? Forget it!

Percom now sell a 6809 adapter: this neat little device is merely a small PCB with a 6809 on it, in addition to a few other chips and a crystal, with a 40pin plug projecting from its rear. All you do is unplug you 6802 chip, plug this kludge-board in, swop over your monitor to a 6809-version, and, presto, for only $70 you have a working 6809 system! Following shortly in this column will be a review of this product.

GROUP NEWS

Following the abysmal failure of the Club's magazine subscription idea, the Group has formed two journal clubs of four members each. The idea is that each member of a group subscribes to a magazine (and thus, with four to a group that means four magazines) and after he has read the magazine, he circulates it to the other members. With a week per member, that means that the subscriber gets his magazine back three weeks later. It is then his to keep. In return, of course, he receives three other magazines for a week each. Amongst the magazines subscribed for by the 6800 Group are Byte,Interface Age, Kilobaud, Dr Dobbs, 68 Micro Journal and Personal Computer World.

The journal clubs are:
Pierre, Neil, Geoff and Anthony
GÜnter, Ansgar, Paul and Jonathan

The Group has acquired a copy of "Best of Interface Age" Volume 1 - it consists of source-code listings a four tiny BASIC interpreters: a 6800 one, two 8080/5 ones, and one for SC/MP.

Our MPU board (containing a 6802, 1K RAM, 1K EPROM, a PIA, cassette interface, and single-step circuitry) is now finally ready for the camera - I will publish in this column a series of specials on this board. In addition, the Group has encountered quite a demand for our 8K static RAM board (using 2114s) and we are now going to make a second run of these. The board has worked perfectly for months now, and will cost R18 to R20 (it is a double-sided thru-hole plated PCB, with gold-plated edgeconnectors. It was reviewed in C3PO (2(8): 14) and should cost no more than R100 to build (including the RAM! - but excluding the cost of the PCB). If you want one, please phone me.

JUNE DIARY ENTRIES:

Wed 4: 20h00 Club meeting
Sat 21: 14h00 Group meeting
(Muller Residence)
Sun 29: 14h00 Workshop 2 Session 5

JOURNAL CLUB

Again, due to space problems, the book review on "Programming for Microprocessors" will have to be carried forward to next month's edition. However, here are three magazine reviews:

MACHINE CODE TIGHTENING

Two useful little articles on how to prune 6800 machine code to fit into ROMs of limited size - attention all those attending the Group Workshop session on the Group Monitor!

The articles are followed by a sensible word of warning - to only indulge in the practices outlined in a last-stand situation to shorten code ... it is obvious to see what some of their techniques could do to structured code!
(Byte, 5(2):146)

6800 PASCAL

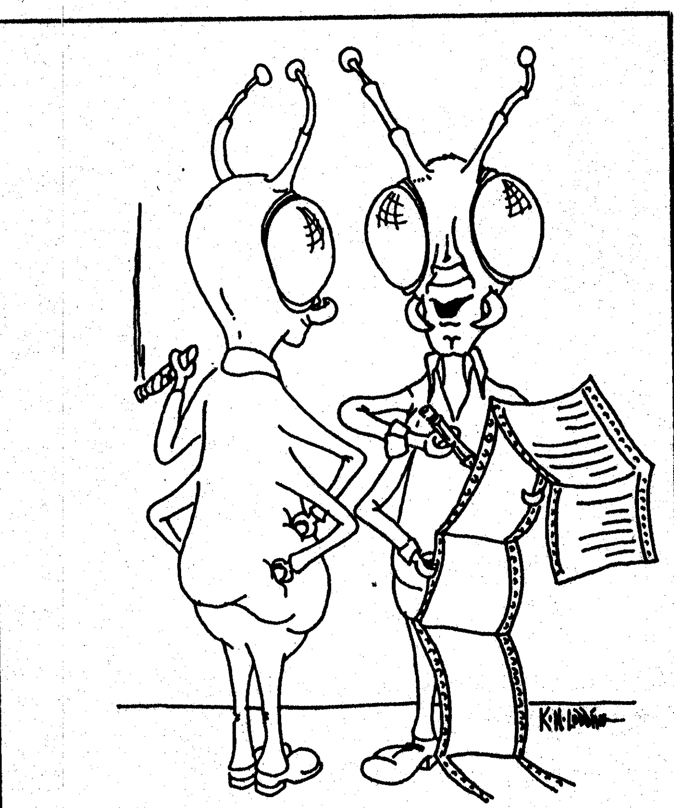
For those who are sufficiently advanced to be able to consider the purchase of Pascal, and who operate a SWTP system with Flex, this is a review of a European produced Pascal package selling for $150. For the rest, eat your hearts out!
(Byte, 5(3):184)

LANDING SIMULATOR

From the sophisticated to the simple: you too can play your own simple, but, I daresay, satisfying game of lunar landing - and all you need is a D2 Kit, two DtoA converters, and a scope. 6800 assembler listing provided.
(Byte, 5(3):132)

POWER DOWN

Something I saw in Byte about a year ago:



"I think I found the human in your code."

Pierre H Wilter

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

C3PO Flea Market

FOR SALE: 1 GLASS TELETYPE VIDEOBOARD, COMPLETE AND IN WORKING ORDER, ALL IC'S SOCKETED. WHAT OFFERS? PHONE DAVID FRANCIS: 52-2884.

WANTED: PRINTER OR MONITOR OR BOTH. CAN ANYONE HELP DAVID FRANCIS? PHONE 52-2884.

WANTED: STEEL OFFICE CABINET WITH DOORS AND SHELVES. ANY CONDITION. CHEAP. PHONE MARTIN: 51-2420.

FOR SALE: MEMORY, LOW POWER EQUIVALENT TO TMS4044 4Kx1 STATIC RAM. PHONE 77-1309.

FOR SALE: SWTP 6800. PHONE 77-1309 FOR DETAILS.

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

START SAVING NOW - YOUR ANNUAL SUBSCRIPTION WILL SOON BE DUE.

This Month's Meeting

WEDNESDAY, 04 JUNE 1980

MAIN HALL

HERZLIA HIGH SCHOOL

20H00

ANNUAL GENERAL MEETING

(SEE FRONT COVER FOR AGENDA)

FOLLOWED BY

CHEESE AND WINE

THE MINUTES OF THE ANNUAL GENERAL MEETING HELD ON 04 JULY 1979

The meeting opened at 20h00 with a welcome by the chairman.

Chariman's Report: The Chairman delivered a short report on the first year of the Club which he said had seen substantial growth and two changes of venue for the monthly meeting. The first large project - a glass teletype - had been successfully initiated and he hoped that this was the forerunner of many more.

Treasurer's Report: The Treasurer reported that the Club had a surplus of R620,00. Membership dues would remain at R12,00 for the next membership year.

Election of Office Bearers: It was proposed that the committee be reelected in toto and this was accepted by a majority showing of hands. In addition three more members were proposed and elected to the committee, viz. Messers Parkyn, Rudolph and Walsh. A new chairman, Mr R. Bernstein was nominated and elected.

General: No points were raised under this heading.

The meeting closed at 20h50.

TREASURERS REPORT

The Income and Expenditure account that follows outlines the expenditure that was made by your committee for the year to date.There is a surplus of funds which will be carried forward to the new year. There was a considerable amount of joint component buying which just exceeded R3000.You will notice that the amount deposited by Members exceeds the amount spent on component purchases by R178.28 which will be refunded upon request or credited against next years membership subscription. A list of these amounts follows together with the members number. This is the number which appears on your address label. We look forward to a good year ahead but must stress that it is your club and therefore hope that you will come forward with suggestions on the best way to spend club funds so that they can be of benefit to all members both in the enjoyment of their hobby and the sharing of knowlege.Because of increased running costs like printing and postage I recommend that the new committee increase next years subs to R15.

Amounts owing to Members.

Member No.	Amount.
07	10.00
10	.44
23	16.68
37	11.09
21	10.00
15	30.15
39	10.00
19	5.00
11	10.00
77	31.66
56	5.00
61	38.26
Total	178.28

Amounts owing from C3PO Advertisers.

AUDIOLENS	25.00
NORTHERN OFFICE	50.00
CAPE TEKNIKON	25.00
SHARP ELECTRONICS	25.00
COMPUTER WORLD	25.00
JUTA'S BOOK SHOP	25.00
	175.00

CAPE COMPUTER CLUB
Provisional Income and Expenditure Account
for the membership year ended 31/5/80

INCOME:		
Subs	771.00	
Donations	21.16	
Component Sales	79.18	
Advertising Income	305.00	1176.34
EXPENDITURE:		
Bank Charges	12.44	
Catering	53.80	
Club Advertising	43.20	
Gratuities Paid	40.00	
Magazine Subs	53.01	
Postage	54.13	
Printing & Materials	183.19	
PCB Developement	27.67	
Provision AGM Catering	100.00	
Prov June C3PO Printing	25.00	
	592.44	
SURPLUS:		
Income over Expenditure	583.90	1176.34

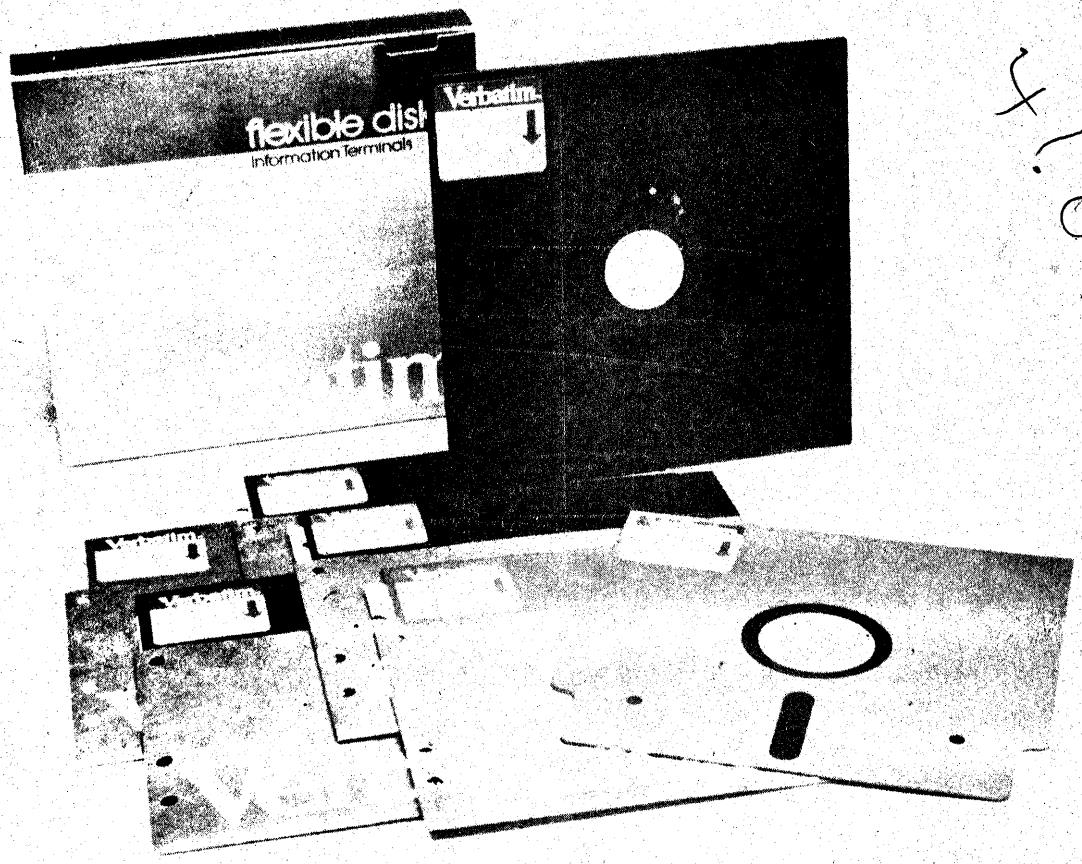
CAPE COMPUTER CLUB
Provisional Balance Sheet
as at 31/5/80.

Reserve	516.04	Scope	150.00
Surplus	583.90	Stationery	120.80
	1099.94	Debtors	175.00
Provision for AGM expenditure	125.00	Plus Plan	872.40
		Current a/c	85.02
Amounts Deposited by members	178.28		
	1403.22		1403.22

Verbatim®

Removable magnetic storage media

Flexible Disks



Our Verbatim diskette line includes the standard eight-inch floppy disk in all IBM-compatible series. We also offer all 32 hard sector configurations for drives manufactured by Shugart, Memorex, and other drive manufacturers.

Our Verbatim 5.25-inch minidisk product line has three industry standard configurations: soft sector, 10 sector and 16 sector minidisks.

- All diskettes are tested and certified to meet or exceed ANSI (American National Standards Institute), ECMA (European Computer Manufacturers Association), and ISO (International Standards Organization) specifications.
- When you buy Verbatim you're buying from the world's largest manufacturer of flexible disks. Verbatim media is the media drive manufacturers around the world prefer for their flexible disk needs.

Minidisks

Advance Promotions (Pty.) Ltd.,
P.O. Box 7250, Johannesburg, 2000
Telephone: 725-4200

CAPE TOWN
P.O. Box 614
7500 Parow
Tel.: 929-559

KEVIN JOHANNSEN