copy 2

Record [illegible]

C3PO

VOL 4 NO 9 OCTOBER 1981

It Takes All Sorts

Hardware Class

Educational Forum

Load Accumulator

Floppy Diskette Users

Cape Computer Club Printout

P O BOX 6251, ROGGEBAAI, 8012

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

Contents

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

VOL 4 NO 9 OCTOBER 1981

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

EDITORIAL : Donald Cook 1

C3PO SCRATCHPAD : Geoff Sturges 2

IT TAKES ALL SORTS : Geoff Sturges 3

HARDWARE CLASS IN RETROSPECT : Neil Walsh 6

THIS MONTH'S MEETING 7

EDUCATIONAL FORUM : Donald Cook 8

FLOPPY DISKETTE USERS : Kevin Johansson 10

C3PO FLEA MARKET 11

LOAD ACCUMULATOR : Peter Llewellyn 14

6800 MICROPHILE : Neil Walsh 17

APPLE TURNOVER : Anthony Rose 22

Z80 GROUP REPORT : John Whitchurch 24

ATOMIC MATTER : Peter Reber 25

TRS-80 GROUP REPORT : Ian McQueen 28

Editorial

DONALD COOK

First, the bad news - if you missed last month's meeting you missed the first three parts of "The Mighty Micro". The good news? This month you get to see the last three parts, again thanks to Ian and Gunther for the use of their V.C.R. and TV.

"The Mighty Micro" - presented by Dr Christopher Evans - reveals the importance of this, the so-called era of the silicon chip. The 1st Part, "The Coming of the Processor" traced the beginning of calculating machines, from Pascal to Babbage and then to the early war time machines that cracked the virtually uncrackable German Enigma codes, and finally to the micro or silicon chip. The 2nd Part, "Of Machines and Money", discussed the role of the micro in our daily lives suggesting a totally cashless society, micros in schools to aid personal tuition, instant diagnosis in hospitals, in short a social revolution, one which man must be trained to cope with. Part 3, "The Political Revolution", looked at the concepts of Prestel and similar ideas applied to voting - just imagine, instant results!

To come ...

Part 4: "The Introverted Society" will look at the idea that children will be taught by computers and that books, as we know them, may well be replacd by silicon chips that would be cheaper and take up far less storage space. Would computers teaching in homes be the start of an introverted society?

Part 5: "The Intelligent Machine". Is the micro a threat to mankind? Could man be replaced as the dominant being on this planet? Will man and micro work as partners to combat problems that afflict our world ... famine, disease and the dangers of war.

Part 6: "All Our Tomorrows". Four leading figures in the field of micros give their opinions of the changes in the wake of the computer revolution.

See you there ...

Accolades to Neil (the noted hardware boffin). Thanks, Neil, I know that those who attended the first class are pleased that you agreed to run the course.

Which reminds me, a popular column has reappeared this month. Yes, Microphile has made a comeback - the 680X's are getting things together in a big way.

In compiling this issue of C3PO I was struck by the quantity and quality of original articles. It is really gratifying, especially as they were unsolicited. Many thanks to these contributers who make this a worthwhile publication.

"I think what we need now is someone called a computer programmer."

C3PO Scratchpad

GEOFF STURGES

This month's meeting is, of course, at the Athenaeum and we shall be continuing the screening of that great classic "The Mighty Micro" by Dr. Evans.

It struck me at the last meeting that at the tea break there was a single lone solitary 20 cent piece all on its own (Tautology Rules, O.K.!). That means that there was one honest visitor who dipped into his pocket that night.

Last month, in Apple Turnover, mention was made of the purchase by the Club of floppy diskettes for sale at the fantastically low price of R 3,50 each (incl. G.S.T.). It should be pointed out that this special offer is for Members only, and also for private use only. We don't want to go into competition with the normal retailers, but if we can obtain a special benefit for our Members we shall of course do so.

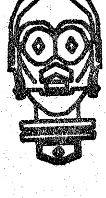
We have been informed that there is a supplier who is willing to make bulk purchases of certain magazines for the Club (at bulk prices, naturally) should there be sufficient interest. The relevant magazines are "Kilobaud", "80 Microcomputing" and "73". Anyone interested should contact us at the next meeting if not sooner.

The other day I saw a newspaper report (with photo) that the R.S.A. has produced its own integrated circuits which consisted, if my memory serves me correctly, of 6 transistors, a diode and a resistor. That's progress, hey. At this rate we'll be churning out J-K flip-flops before Buck Rogers marries Wilma!

It has been suggested that the Club sponsor a competition, presumably for both hardware and software. Home-built machines abound in great numbers now as was demonstrated at the "Microbition 81" so there is no excuse for not entering, we have seen the evidence, Holmes. More details will be given later.

Another suggestion is that the Club has another Project (remember the Glass Teletype - mine still works 100%!). The suggested project is for a Modem so that your computers may all talk to each other while you watch "Westgate". The design will be of the G.P.O./H.P.K.-approved variety and very cheap to costruct (mention was made of about R40,00). There will also be standard formatting, etc., so that your machine may talk to any other machine (in theory at least).

I have to be vague in certain matters, as you may have noticed, as the left hand rarely knows what the right foot is doing. Anyway, there may be a column elsewhere in this magazine called "Query Corner", of so iets. The idea is that if you have a problem then let us know and we shall not only let you have the benefit of our wisdom but everyone else will know how daft you are for asking the question in the first place! By the way we do not want queries from "Worried, Kenilworth".



It Takes All Sorts

GEOFF STURGES

This article does not set out to discuss the different sorting techniques and to compare their advantages and disadvantages. That topic could well form the basis of another series of boring articles. Rather, the objective of this article is merely to discuss modifications to any sort program in order to adapt it to your requirements.

There are many algorithms for sorting, but it has been stated that there is no "best" method. All have some disadvantages, such as being very slow if the array is already partially sorted. I read somewhere that an average of 60% of all computer time was spent on sorting, so that this aspect of programming is bound to affect you sooner or later.

The algorithm used in this article was devised by D.L.Shell and, although not reputed to be the fastest method under ideal conditions, is very much faster than the "bubble" sort and is also easily coded into BASIC.

The BASIC Listing No. 1B was begged, borrowed, stolen, pillaged, raped and adapted from an undisclosed source, namely "Kilobaud" via "CPUCN", and Listing No. 1A is my adaptation of it in order to speed up the processing time by approximately 25%. You can try your hand at making it even faster.

In order to sort a set of variables they must be stored in a subscripted array. Now we come to the question of how many elements may we have in an array to be sorted. Well, there are four approaches:-

Firstly, you can ignore the question completely and continue this article further on, or start another article further on.

Secondly, you can look in the manual and find out that your BASIC only allows up to 255 elements per array.

Thirdly, you can choose a largeish number and keep running the program several times, each time decrementing the number you first thought of, until you don't get an "out of memory" error message.

Fourthly, you can be brave and attempt to calculate the magic number. Naturally, each BASIC differs in specific details, but I shall explain how one BASIC interpreter works and you can experiment with your own BASIC if you wish to pursue this method further.

Numeric and string variables are treated differently (but equal - we musn't be unNATriotic) as can be expected. Arrays for numeric variables consist of the following:-

2 bytes for the variable's name
3 bytes for don't know yet
    (at least I'm honest)
2 bytes for no. of elements
5 bytes for <u>each</u> element

My trusty calculator tells me that that is an overhead of 7 bytes per array. The overhead will be 12 bytes if your BASIC starts from element "Ø" but you don't make use of it.

Arrays for string variables consist of the following:-

2 bytes for the variable's name
3 bytes for still don't know yet
    (I'm still honest)
2 bytes for no. of elements
3 bytes for <u>each</u> element
    (1 byte for length of string
     and 2 bytes for address of
     string)

This is still an overhead of 7 bytes per array, but only 10 bytes if you don't use element "0".

On top of that of course are the lengths of the actual strings. The lengths of the strings should be taken as the maximum length that can occur if you don't know what the average length is PLUS 2 bytes which appear to be pointers back to the next element.

To summarise then, if we are to sort "x" elements then our calculations would look something like this for a numeric array:-

Top of available memory	(m)	32767
" " prgm & ord vars.	(p)	2096
Free memory		30671

Array overheads: 7 bytes per array	7
Bytes per numeric array	5

Therefore the number of elements we should be able to sort at any one time is:-

$$x = \frac{30671 - 7}{5}$$

Which rounded down gives 6132 elements per array. Hurray!

If we are sorting a string array, with the maximum length of a string being say 12 bytes, then our calculations will look something like this:-

Free memory	30671
Array overheads: 7 bytes per array	7
Bytes per string array	3
Max. length of strings	12
Pointer	2
	14

Therefore the number of elements we should be able to sort at any one time is:-

$$x = \frac{30671 - 7}{3 + 14}$$

Which rounded down gives 1803 elements per array.

Now we can proceed towards the programs proper. The table below gives the combinations (or is it permutations?) of the parameters to be considered when analysing your requirements.

No. of disk drives	Ratio of length of seq. file to disk capacity	Single variable records		Multiple field records sorted on:		
		Numeric	String	1 Field	2 Fields	>2 Fields
0	RAM only	A	B	C	D	E
1	< 1/3	F				
2	< 1/2	G				
2	> 1/2	H				
>2	< 1	J				

This article will only give guidelines to the routines corresponding to the squares marked with a letter. The routines for the unmarked squares may be easily obtained from combining the two "primary" routines.

AT LAST!

Routine "A":

This is the basic routine to sort an array "A(N)" with "N+1" elements (element "Ø" being used). "T" is a temporary variable for use while exchanging elements:-

```
10 REM SHELL SORT OF SINGLE NUMER
   IC VARIABLE
20 N=   :DIM A(N)
.
.
.
100 GOSUB 1000
.
.
.
990 END
1000 REM SORT SUBROUTINE
1010 M=N+1:FOR H=1 TO 16:M=INT(M/
     2):IF M=Ø THEN RETURN
1020 FOR J=Ø TO N-M:FOR I=J TO Ø
     STEP -M
1030 IF A(I)>A(I+M) THEN T=A(I):A
     (I)=A(I+M):A(I+M)=T:NEXT
1040 NEXT J:NEXT
```

Listing No. 1A

If your program drops out of the last "NEXT" then you have a 16-bit processor and an array size greater than 64K elements!

For those whose BASICs do not use element "Ø" in an array, do not accept multiple statements on one line, and probably don't allow more than 255 elements in an array, then the following "traditional" subroutine is for you:-

```
1000 REM SORT SUBROUTINE
1010 M=N
1020 M=INT(M/2)
1030 IF M=Ø THEN RETURN
1040 J=1
1050 K=N-M
1060 I=J
1070 L=I+M
1080 IF A(I)<=A(L) GOTO 1140
1090 T=A(I)
1100 A(I)=A(L)
1110 A(L)=T
1120 I=I-M
1130 IF I>Ø GOTO 1070
1140 J=J+1
1150 IF J>K GOTO 1020
1160 GOTO 1060
```

Listing No. 1B

Routine "B":

This is identical to Listing No. 1A except, of course, that we are now using string variables. We now have an array "A$(N)" and a temporary variable "T$".

```
10 REM SHELL SORT OF SINGLE STRIN
   G VARIABLE
20 N=   :DIM A(N)
.
.
.
100 GOSUB 1000
.
.
.
990 END
1000 REM SORT SUBROUTINE
1010 M=N+1:FOR H=1 TO 16:M=INT(M/
     2):IF M=Ø THEN RETURN
1020 FOR J=Ø TO N-M:FOR I=J TO Ø
     STEP -M
1030 IF A$(I)>A$(I+M) THEN T$=A$(
     I):A$(I)=A$(I+M):A$(I+M)=T$:
     NEXT
1040 NEXT J:NEXT
```

Listing No. 2

Next month: Part 2 - multiple field sorts, etc.

Hardware Class in Retrospect

Neil Walsh

The initial discussion was aimed at getting across only the most basic ideas behind the working of a microcomputer. The approach was to introduce the newcommer to the way a microprocessor is put to work in a system. One need not be an electronic wizard to get a system working sucessfully but is a great help to have some idea what goes on below the top cover of your microcomputer. With the new micros we see today, it is impossible for the sucessful programmer to remain ignorant of the hardware.

The microprocessor chip is really only the following things:

a) a complex electronic component
b) a complex switch
c) a sequencer
d) a decoder
e) an encoder

The first point cannot really be covered in any detail. One can show someone the slab of plastic with 40 little legs coming out of it, however, few people are really impressed by its appearence. One could try to explain how it is made but that is of little importance to the user. It is one of those things we accept as a "black-box".

Under the second point it was explained that the switching was done by the device (MPU) by making its control lines have a ON or OFF voltage from its various connector pins. The signals on the lines were represented by the presence of +5V on the "ON" or HIGH state and by 0 volts when in the "OFF" or LOW state.

Because there is so much that has to be switched on and off in a MPU system it is possible to have a separate line for each item. The problem is over come by internal encoding and then sending the signal along several lines and the external device decoding the 'encoded' signal and then activating the right element after the process has been reversed.

The major part of the system that is switched is memory. Most microprocessor (MPU) chips have the ability to switch on or select any one of 65000 possible memory cells. This needs only 16 control lines to do this. This is because, by switching some of the line on and some off in different ways it is possible to get over 65000 combinations. The receiving device,in this case memory,would have to look at the signal pattern on the 16 lines, decode the pattern and switch the corresponding memory cell on.

Because the hardware sesssions are short there is no time to go into the 'mathematics' behind why it is possible to get so many possible switching combinations on only 16 control lines. However,it was suggested that the interested person could start with, say, only 4 control lines and see how many combinations can be made. Remember,some or all can be low while the rest, or all, high. The remote or external device would have to look at the pattern on the parallel lines and, by a process of 'decoding', select the relevent device. Notice that only one device can be selected at a time.

Encoders or Decoders today are usually single components. Quite often they are actually included in the RAM chips themselves. Again, it is not really necessary to investigate their internal workings except under cases of extreme interest.

When the selected memory location is switched into the system, the stored signal again is represented by a high or low voltage level. These level patterns are held in a particular location in parallel, eight 'bits' at a time.The memory location can receive into itself or output this pattern over another eight lines which run in parallel and enter the MPU chip through eight connector pins. These signal lines are called "the data bus" while the other 16 control lines mentioned are called "the address bus". The data bus is said to be 'bidirectional' because data can be travelling to and from the MPU and memory in either direction depending on whether a read or write operation is taking place.

The MPU is the master device and therefore the controller which activates the address bus. Therefore the address bus is undirectional. The memory or RAM, as its called, has to be told to send back or receive data patterns.Therefore,it is necessary for the MPU to produce another signal which tells the bus lines to switch over direction causing the RAM board to respond in the correct manner.

Towards the end of the discussion it was mentioned that the memory in a system has two major uses. Firstly, to hold sequencer patterns,known as instruction codes, and the latter was to hold data values. The MPU being a squencer device is continually trying to cycle. A cycle consists of setting all the address lines after reset startup to the low state corresponding to address zero. The MPU also via the data direction control line will tell the memory addressed (addr zero) to send its contents to the MPU. This pattern will, if all's well, correspond to a relevent sequencer code. The MPU will decode this pattern and cause some internal activity to take place. After completion of this activity the next address will be put on the bus causing the next RAM location to be selected. The sequencing will go on for ever or until something

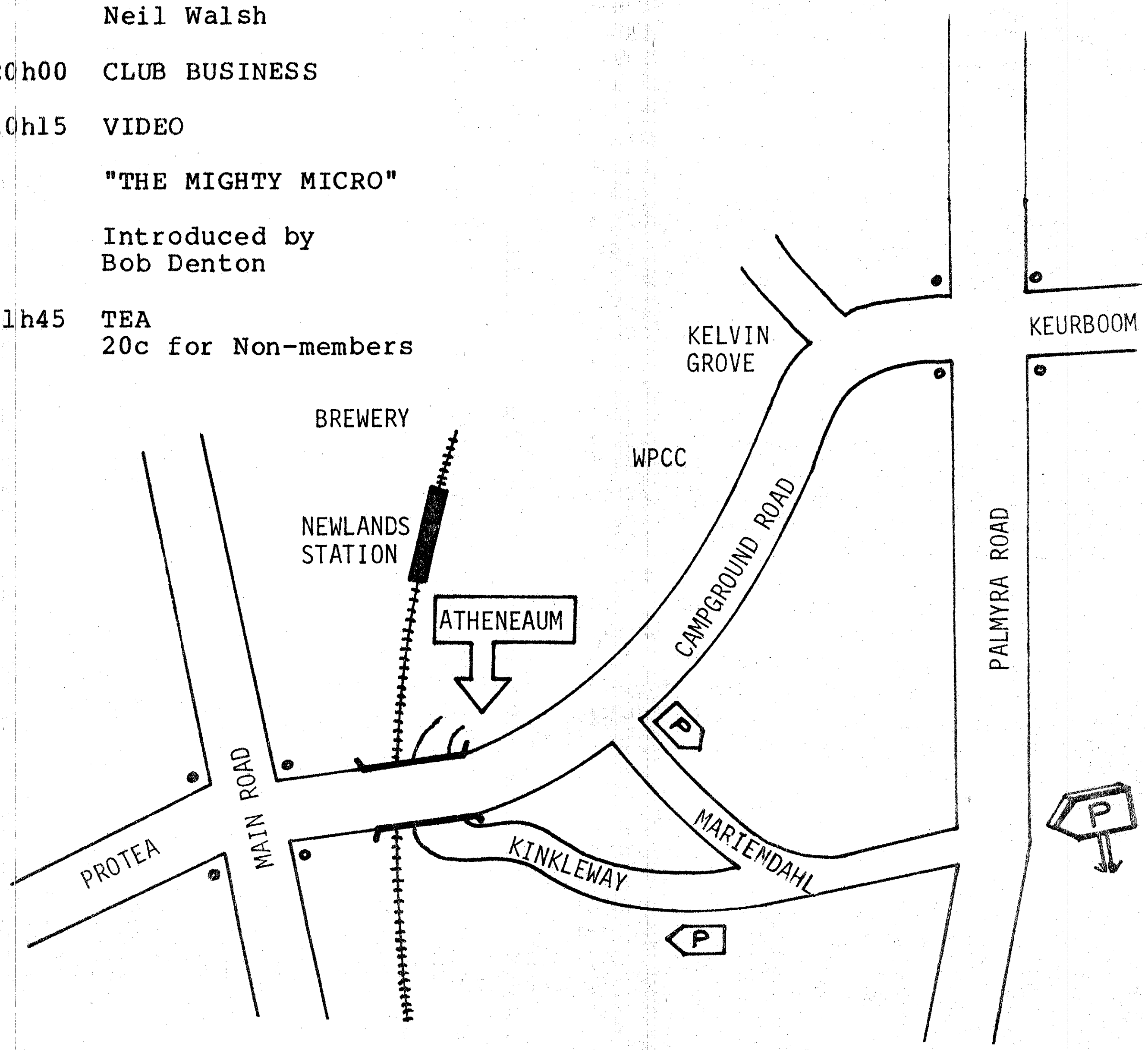
goes wrong or the machine is switched off.
The actual sequencing process will be investigated more deeply at the next session.
The general MPU chip we have looked so far has 16 address lines, 8 data lines; one to switch the data direction. We mentioned also that there is need for a RESET line which can be activated remotely by the user or startup logic. The electronics people will also remind us that there must be one to supply the power to the chip. This is +5 Volts and, of course, an additional line for ground or zero volts. There are other lines coming into and out of the MPU component but these will be gone into at a later stage.
For all of you who are wondering what a "chip" is an explanation of the term follows. The components mentioned so far, like MPU's, RAM's, decoders and encoders are complex or integrated circuits which are fabricated on a single fine slice of silicon known as a 'chip'. This chip is usually moulded in a block of plastic with little connector legs coming out of it. The word CHIP is electronics jargon, which generally refers to any complex component of this nature.

This Month's Meeting

THURSDAY, 01 OCTOBER 1981

THE ATHENEAUM, NEWLANDS

19h30	HARDWARE CLASS Neil Walsh
20h00	CLUB BUSINESS
20h15	VIDEO "THE MIGHTY MICRO" Introduced by Bob Denton
21h45	TEA 20c for Non-members



Educational Forum

DONALD COOK

I have often made the suggestion that one interesting and beneficial application for the home hobbyist to consider is that of computer aided learning. The most common reply is: "That's all very well, but what could I possibly do for my children/niggies/small fiends." The time has come the walrus said ... (not again!!) ...

Firstly, what is so special about using a micro for assisting the learning process? In the good old days ... about 10 to 15 years ago! ... a certain type of book dubbed "the programmed learning text" was extremely popular. Many of you have no doubt seen examples of these, if not worked from them. Truly a revolution ... well, almost. Like Babbage, Skinner was before his time. He was unable to bring his ideal learning situation to fruition because the technology of his age was not able to meet his needs. Skinner's concept was an interactive medium which prompted and rewarded the learner, allowing him to progress at his own rate and giving instant feedback of success or failure. I am sure few of you have not suffered the frustration of waiting one or more weeks to receive the answers to a question paper. The immediacy of the challenge has been lost and no or very little learning takes place when the correct answer is finally revealed. Skinner's aim was to overcome this as well as the problem of the person who needs to regress to consolidate a fact or idea. (All pigs are equal!)

Unfortunately for Skinner the medium he required was in part unavailable and in part just too young to be of any value ... the war time computers were far from capable of storing the data required and lacked much of what Skinner needed for interaction with the user in the way of peripherals. Just come and try my teletype if you don't believe me! Hard copy output is simply no better than Skinner's learning machine with a paper roll that displays questions and answers in a simple linear program.

By now those of you with micros ... and memory mapped VDS's to boot ... will know that you can supply most, if not all the needs for Skinner's ideal. Of course, one dreams of a complete learning system consisting of a comprehensive library of video illustrations, both still and moving, linked with a voice recognition and generation system all controlled by a micro and capable of being reprogrammed at the drop of a hat (or should it be chip nowadays?). Enough said ...

To a large degree what you do, educationally speaking, with a micro is dependant on your ability to express ideas generally. It can be a dry and factual question and answer technique or a more humorous and fun approach using graphics to illustrate your point and, for example on the Apple, using the paddles as a means of input. In particular here, I am thinking of a pointer that can be moved around on the screen to answer such questions as: "Where on the graph would you find...?" or perhaps in a simulated experiment: "To what level would you fill the pipette ...?". the same idea could also be applied to a multiple choice response situation.

Before one embarks upon an educational program some thought must go into its design. Here, more than anywhere else, design is important and I refer not only to the structure of the program but also to the manner in which information is presented and responses solicited.

To get a better idea of what I mean let's look at Skinner's first attempts and the aims he had in mind. Firstly, let's consider the idea of the interactive medium. The response of the machine must be polite and friendly, not deprocating or patronising. I mean why: "SORRY, that's wrong."? Who's sorry? It is far better to use: "Wrong, David, try again.". Which brings me to another point, that of using the student's name ... not every time, but frequently. Vary the comments, e.g. "How are you doing, David?", "That's good.", and so on.

Secondly, let's look at the idea of reward. Well, depending on the level and age groups this will take on different forms. For the oldies it's the warm comments already indicated, or perhaps simply "Correct" with a "Well done", "Nice going", "You're catching on, David" tagged on in a random (RND) fashion. For the younger people a little "Fred" or "Maxwell" waving or clapping is more appropriate. Possibly 5 shots or so in a popular game after a successful test would fit the bill. It is also possible that the whole program could revolve around a game. (Many American CAL programs do this.)

Lastly, we must consider the idea of the student progressing at his own rate. This implies two things:
(1) that speed of operation can be controlled, and
(2) that the level of difficulty can be controlled.

I find that the first need can be supplied adequately by two simple devices, a wait loop and a simple GET; after the statement "PRESS <SPACE> TO CONTINUE OR <ESC> TO END". The second need is not so easy to fulfill but there are many approaches, typically, lots of small modules to which the program can branch according to the success rate of the student, or a simple selection in terms of size of numbers used, as for example in a multiplication exercise. If you've got disk storage, of course you can make use of files that contain the data to suit a choice of level of difficulty.

Now a few pointers to help you get the message across. As you would normally do with TOP DOWN DESIGN:

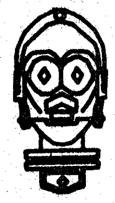
1. Begin by writing down the aim of your program.

2. Plan the screen display carefully - don't overcrowd the screen. Keep it simple and if you need to display lots of text, don't fill screen after screen of never ending words, break the message up into readable chunks and vary the presentation. Even go to the extent of flashing a few words singly and then repeating them as a phrase to give emphasis.

3. Keep the program modular, one in, oneout. That way you will use the same modules over and over again, e.g. wait loops, etc. I try to keep my modules parameter driven which increases portability.

4. As you progress from level to level in your design make sure you don't lose sight of your original aim. Always make sure you don't proceed onto a new idea in the lesson before you've consolidated and reviewed the student's progress.

Next month I will publish a program or parts thereof to illustrate these ideas. I am also eager to publish any ideas you have found or think would be useful.



Floppy Diskette Users

KEVIN JOHANSSON

As suppliers of floppy disks/diskettes/flippies - call them what you will - we feel a bit needs to be said about this form of magnetic media storage, as very little ever seems to be written on this subject. A person buys a diskette drive, but nobody ever warns him that the diskette needs care in handling.

I flip whenever I walk into a diskette user's office. What do I find? Diskettes on the table, out of their envelopes, near ash trays or sandwiches, or even in direct sunlight, and then I hear that the lifespan is only 6 months to 1 year!! Compare these diskettes to the identical ones sold to major computer users where the environment is controlled, and the lifespan is up to 6 years! Wouldn't you like 6 years life out of a diskette? (Actually, I don't mind if you buy a replacement every 6 months.)

Right, so let's try and increase the life of a diskette. At the end of this article, I've included a "Care and Handling", which is somewhat general, but effective.

One other area is cropping up which is very important: make sure you are using the right diskette. There are 30 different types available, in sectoring, track densities, etc. and what I find for, example, is people using 40 track tested at 77 tracks or 80 tracks. That's fine, sometimes you get away with it, but sometimes you don't.

I think it is fair to say there are no bad diskettes on the S.A. market; all the brands available meet ECMA specifications - but unfortunately not all the drives meet ECMA standards. Drives are sometimes set up with a certain diskette, and will therefore only operate successfully with that particular brand, because the oxide coatings differ from manufacturer to manufacturer. Actually, I find it is quite difficult to get specs on drive read/write currents (bias) in Cape Town - am I perhaps asking the wrong people? If anybody can help me with these specs I'd appreciate it. I have the specifications of ECMA and ANSI standards for drives and the specifications for the diskettes - if anybody's interested, I'll be glad to post you copies. (Phone me at 220650.)

I must say all you hobbyists are quite clued up, and I'm sure a lot of this is old hat to you, but I still ask one question - how carefully do you treat your diskettes? Anyway, I'd appreciate it, on behalf of all diskette manufacturers, if you'd pass on the care and handling details to your computer operators, staff who handle diskettes - or even computer salesmen. (I found a box of warped diskettes in the back of a salesman's car once - he was I!) And again on behalf of all manufacturers, don't blame the diskette because it doesn't work. It could be the diskette at fault - it is after all a manufactured item and millions are manufactured a year - but it could be the wrong diskette for the drive, it could be damaged by the environment, it could be static, it could be wrong bias, ... or a hundred and one other things. But on the whole, touch my head, they stand up to misuse. So happy flips and hope nothing flops. Here's the Care and Handling.

CARE AND HANDLING OF DISKETTES

(1) Disks to be stored should not be removed from their shipping container, and stored in a cool environment of 10° - 52° C and R.H. of 10% to 90% with noncondensation. Avoid rapid changes in temperature and humidity.

(2) Do not fold or flex - even though the name "flexible" is used, this is descriptive in nature, not suggestive.

(3) Do not touch exposed areas of the disk surface. Skin oils and other residues from creams and after shave lotions can damage magnetic coating.

(4) Do not force the diskette into the drive opening. FRONT LOAD: Make sure the disk is fully inserted before closing the door; this prevents damage to the centre hole of the diskette. TOP LOADING: Allow the disk to free fall into the drive; forcing can cause damage to the disk jacket. If diskettes prove difficult to load, call your authorised service engineer.

(5) Remove the disk easily from the drive.

(6) Keep operational areas clean. Avoid smoking, eating or drinking around diskettes as they can become charged with static and attract dust particles.

(7) Do not write on the disk jacket. First write on a self-adhesive (non transferable adhesive) label, then apply to diskette.

(8) When not in use, keep the diskette in the envelope and in the box. Never attach paper clips or staples, and don't throw diskettes into baskets to be transferred to other departments.

(9) Album record racks are not a safe means of storing diskettes as they tend to dent the disk.

Teletype Terminal. Olivetti Te300 with paper tape punch and reader on trolley with castors. Excellent condition. Price R150. Phone Dr Wilter 555151(W) or 554176(H).

C3PO Flea Market

FOR SALE

THE BARGAIN OF THE YEAR

ADVANCE PROMOTIONS have very kindly allowed the Club to make a bulk purchase of their Verbtim Minidisks MD 525-01, single sided, double density, soft sectored, certified 40 track. These are available to members in sealed boxes of 10 disks at the special price of R35 including GST per box. They will be available against cash or cheque at the next meeting, so come prepared to buy your requirements for the next 6 months or so.
If you wish me to reserve some for you please phone me - Peter Llewellyn at 492089, Cape Town.
P.S. If you haven't paid your subs yet, what you save on the purchase of just one box of disks more than pays it!

SW Tech CT64 with matching green phosphor monitor in excellent condition. R400.
Teletype KSR33 complete with manuals and step down transformer. R75.

Miscellaneous junk to clear e.g. caps, transformers, etc.

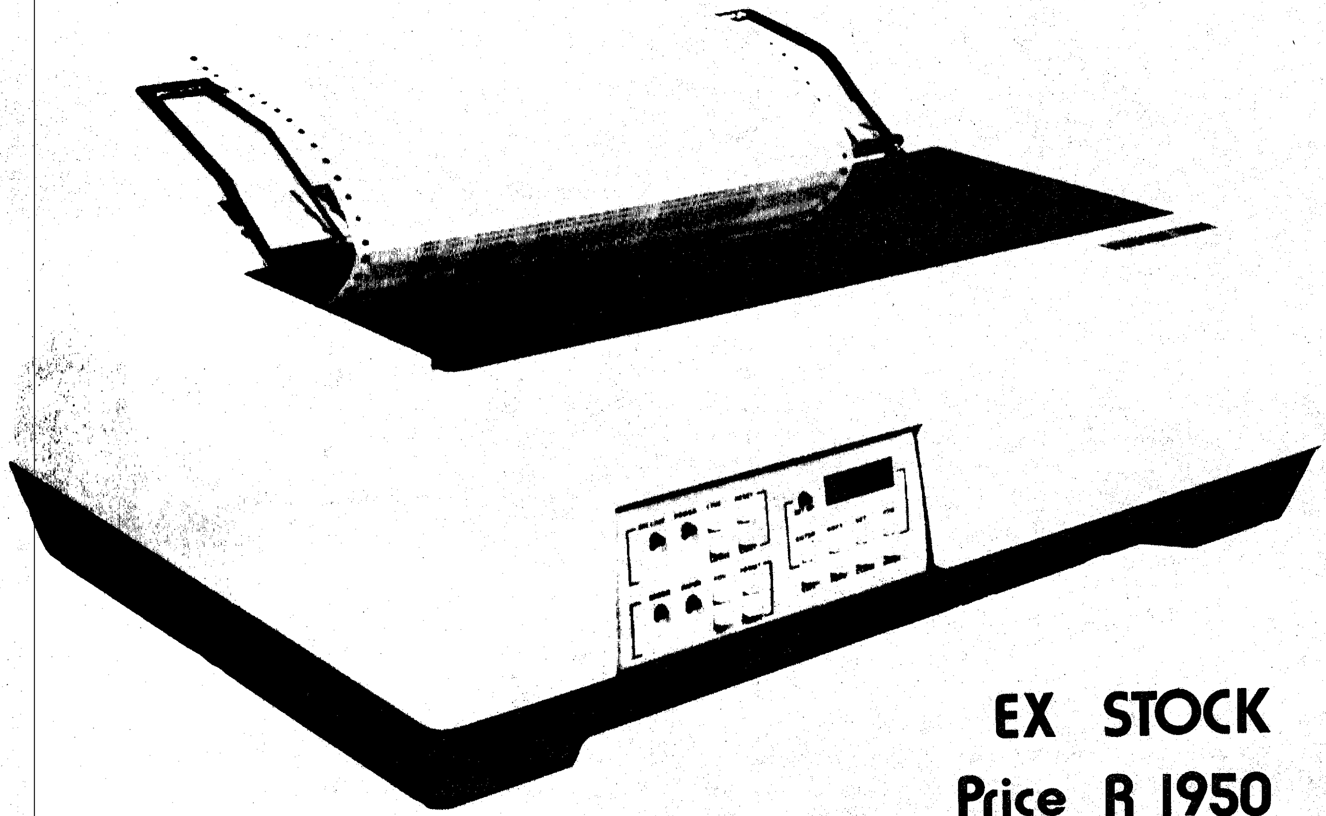
Phone Tony Kenwood : 771309

CPU CARD	R65
4K ROM/1K RAM CARD	R74
I/O CARD	R64
2 MEMORY CARDS	each R80
KEYBOARD	R50

Contact Donald Cook 463615

DATASOUTH DS180

HIGH SPEED MATRIX PRINTER



EX STOCK

Price R 1950

The Datasouth DS180 is a dot-matrix serial impact printer designed for high performance at an economical price. Application flexibility and a long list of standard features make the DS180 an ideal device for small business systems, distributed communications networks and intelligent terminals.

HIGH SPEED PRINTING

Utilizing 180 cps optimized bidirectional printing, the DS180 offers higher throughput than any printer in its class. Its 9-wire printhead produces highly legible 9x7 characters with decenders for lower case letters and true underlining. All 96 ASCII characters may be printed across a 132 column line at 10 characters per inch. Expanded characters (5 cpi) may be selected for highlighting portions of the text.

USER PROGRAMMABLE

The DS180 offers a large number of user programmable features, yet is easy to operate. A unique programming keypad with a non-volatile memory makes printer set-up quick and simple. Top of form, horizontal and vertical tabs, perforation skip-over and auto line feed are just a few of the features the user may select. Communications status may also be programmed and monitored using the indicator panel lights and LED display.

ATTRACTIVE DESIGN

Compact, desk-top packaging allows the DS180 to fit into almost any installation. Its noise dampening cover makes it suitable for use in a quiet office environment. The cartridge ribbon makes routine changes clean, fast and convenient

MICROPROCESSOR ELECTRONICS

Through the use of state-of-the-art microprocessor electronics, reliability and maintainability have been greatly improved The simple modular design of the DS180 provides easy access to all major components. A single printed circuit board contains both the power supply electronics and digital controller for the printer. A self-test feature and diagnostic display panel help the user verify proper operation of the unit and isolate problems should they occur.

COMMUNICATIONS

Interfaces on the DS180 include RS232 and 20mA current loop serial interfaces, and a Centronics compatible parallel interface. Baud rates from 110-9600 and parity selection may be keyed in by the user for his specific application.

FORMS HANDLING

Adjustable tractors accomodate forms from 3-15 inches wide. A head-to-platen gap adjustment ensures optimum print quality on up to 6-part forms. Fanfold paper may be fed from the front or bottom of the DS180. A paper out sensor may be programmed to send a stop transmission character and sound an audible alarm.

QUALITY MANUFACTURING

Reliable performance is ensured by a stringent quality control program. Datasouth uses pretested, high reliability parts from leading manufacturers. Multiple tests are performed on subassemblies during each stage of production, with each completed unit undergoing a final 24 hour print test and burn-in. The DS180 carries a 90 day warranty on materials and workmanship.

DS180 PRINTER STANDARD FEATURES

* Microprocessor Control
* 180 CPS Print Speed
* Bidirectional/Logic Seeking
* 1000 Character Buffer (Expandable)
* 9x7 Dot Matrix
* Expanded Characters
* Adjustable Printhead/1-6 Copies
* 96 ASCII Character Set
* Cartridge Ribbon
* 132 Column Print Width
* Tractor Feed (Front or Bottom)
* Non-Volatile Format Retention
* Top of Form
* Horizontal Tabs (Addressable & Absolute)
* Vertical Tabs (Addressable & Absolute)
* Perforation Skip-Over
* Auto Line Feed
* 6/8 LPI
* Auto End of Line Carriage Return
* 6.2 IPS Paper Slew
* Parallel and Serial Interfaces
* 110-9600 Baud Communications
* Terminal Status Indicators
* Audio Alarm
* Self-Test

PHYSICAL & ELECTRICAL

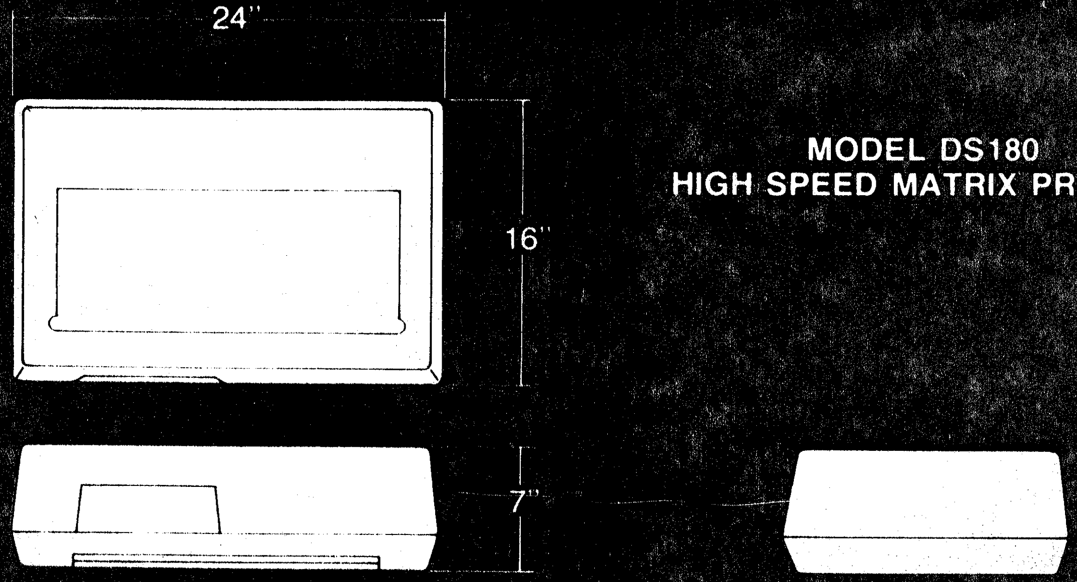
Power:	110 Vac, 60 Hz or 220 Vac, 50 Hz
Temperature:	40° - 100F
Humidity:	20 to 90% (No Condensation)
Size:	7" H x 24" W x 16" D.
Weight:	35 lbs.

PARALLEL INTERFACE

PIN	SIGNAL	PIN	SIGNAL
1	DATA STROBE	19	---
2	DATA BIT 1	20	---
3	DATA BIT 2	21	---
4	DATA BIT 3	22	---
5	DATA BIT 4	23	---
6	DATA BIT 5	24	---
7	DATA BIT 6	25	--- TWISTED PAIR GROUND
8	DATA BIT 7	26	---
9	------	27	---
10	ACKNOWLEDGE	28	---
11	BUSY	29	---
12	PAPER OUT or PRINT OFF	30	---
13	------	31	------
14	GROUND	32	------
15	------	33	GROUND
16	GROUND	34	------
17	CHASSIS GROUND	35	------
18	+5V	36	------

RS 232-C/CURRENT LOOP INTERFACES

PIN	SIGNAL
1	CHASSIS GROUND
2	SERIAL OUT
3	SERIAL IN
7	SIGNAL GROUND
8	CARRIER DETECT
17	CURRENT LOOP TX +
20	DTR
23	CURRENT LOOP RCV +
24	CURRENT LOOP TX
25	CURRENT LOOP RCV



MODEL DS180
HIGH SPEED MATRIX PRINTER

ELECTRONICS **eagle electric co (pty) ltd**

p o box 3106 cape town ☎ **(021) 451421 telex 57-20713**

Load Accumulator

..TER LLEWELLYN

By now I hope you are all getting to know something about the other fellows' micros. As I said in my first article under this heading, in spite of their many differences and individual advantages the 6502, 6800 and Z80 can all perform the same tasks although at times different approaches are needed. In the next few articles I hope to show how certain elementary yet essential tasks are performed by our 3 micros.

8-BIT ADDITION

This is a very basic task. The contents of memory location 0040 is to be added to the contents of memory location 0041 and the total stored at location 0042. All addresses will be hexadecimal form as will data unless otherwise stated. These memory locations are in page zero and as explained last month the 6502 and 6800 have a particular way of doing this:

```
6502:   A5 40    LDA    $ 40
        18       CLC
        65 41    ADC    $ 41
        85 42    STA    $ 42

6800:   96 40    LDAA   $ 40
        9B 41    ADDA   $ 41
        97 42    STAA   $ 42
```

Points to note here are that only the low order byte of the address is required, the high order being 00. The difference in the two programs is because the 6502 has an instruction add to accumulator with carry, ADC, but no straight ADD instruction. The 6800 and Z80 have both ADC and ADD. However when an ADC instruction is to take place for the first time the carry must be cleared to zero, thus the clear carry flag instruction CLC prior to ADC for the 6502. In this case of an 8-bit addition where there is no carry in, the 6800 therefore uses the ADDA instruction, that is add to accumulator A. This micro also has a B accumulator and could therefore have solved this problem using op-codes for LDAB, ADDB and STAB. The Z80 solves this problem using the full two byte address as below.

Where the memory locations are 2040H, 2041H and 2402H, the solutions are as follows:

```
6502:   AD 40 20     LDA       $ 2040
        18           CLC
        6D 41 20     ADC       $ 2041
        8D 42 20     STA       $ 2042

6800:   B6 20 40     LDAA      $ 2040
        BB 20 41     ADDA      $ 2041
        B7 20 42     STAA      $ 2042

Z80 :   3A 40 20     LD A (2040)
        47           LD B,A
        3A 41 20     LD A,(2041)
        80           ADD A,B
        32 42 20     LD (2042),A
```

Reiterating a previous statement, the 6502 and 6800 are more memory orientated while the Z80 is register orientated. The Z80 solution above could have been similarly performed using registers C, D, E, H and L as the intermediate register to which the data is transferred from the accumulator as the contents of a memory location cannot be loaded directly into either B, C, D, E, H or L. The Z80 solution above is noticeably longer than any of the other solutions but is shown as it is the nearest approach in form to the other examples. However a shorter method is to use the register pair HL as a pointer to the memory locations used:

```
Z80 :   21 40 20     LDHL, 2040
        7E           LD A, (HL)
        23           INC HL
        86           ADD A (HL)
        23           INC HL
        77           LD (HL),A.
```

Here the HL register pair is first loaded with the starting 16-bit address. LDA, (HL) means that the accumulator is loaded with the contents of the memory location pointed to by the contents of the HL register. The register pair is then incremented by 1 to point to the next memory address the contents of which in turn are added to A, ADD A (HL). The pointer HL is again incremented and the contents of A are then loaded back to the address now pointed to by the HL register. A point of terminology here is that while the Z80 always uses the expression load, LD, for the increment of data to and from registers and memory, the 6502 and 6800 load their registers, accumulators, index and stack pointer, LDA, LDX, LDS as applicable, but store data from register to memory, STA.

Also note that the 6800 expresses the address 2040 H in the same sequence with the high order byte first, followed by the low order byte, whereas both the 6502 and Z80 reverse this giving the low order byte first followed by the high order byte. 6800 can also solve this basic problem amongst other ways by using its program counter P.C as an index register.

MASK OFF FOUR MOST SIGNIFICANT BITS (MSB)

The logical AND instruction is frequently used in computer programming techniques, for example prior to testing one or more bits for a particular condition. In this example, bearing in mind the op-codes already used above, it is not difficult to construct the necessary programs. We wish here to force the 4 MSB's of the contents of memory location XX40 to zeros without changing the 4 least significant bits, LSB, and place the result in location XX41. In other words, logically AND the accumulator with 0F (0000 1111 in binary-B).

For example:

```
A contains  57 H (0101  0111  B)
  AND        0F   (0000  1111  B)
  Result     07   (0000  0111  B)
```

Any number of bits in any position may be masked off in this way.

In page zero, memory location 0040:

```
6502:  A5 40     LDA $40
       29 0F     AND 0F
       85 41     STA $41

6800:  96 40     LDAA $40
       84 0F     AND 0F
       97 41     STAA $41
```

Elsewhere in memory, e.g. 2040

```
6502:  AD 40 20    LDAA $2040
       29 0F       AND 0F
       8D 41 20    STA $2041

6800:  B6 20 40    LDAA $2040
       84 0F       AND 0F
       B7 20 41    STAA $2041

Z80 :  3A 40 20    LD A,(2040)
       E6 0F       AND 0F
       32 41 20    LD (2041) A
```

Again the 6800 can perform this logical instruction by using its program counter as an index register. Also all three of our micros can perform the other logical instructions, both the OR and Exclusive OR, examples of which we should come to in course of time.

Finally, for this issue of C3PO, I am going to summarise numerically the equivalent op-codes we have come across so far in this series of articles, for ease of future reference, together with some close relatives.

1. LOAD ACCUMULATOR WITH 8 BIT DATA (nn)

```
6502:  LDA,nn       A9 nn

6800:  LDAA,nn      86 nn
       LDAB,nn      C6 nn

Z80 :  LD A,nn      3E nn
```

2. LOAD ACCUMULATOR WITH CONTENTS MEMORY

a) ZERO PAGE - MEMORY LOCATION 00YY

```
6502:  LDA $YY      A5 YY

6800:  LDA $YY      96 YY
       LDAB $YY     D6 YY

Z80 :  See 2. b)
```

b) ANY MEMORY LOCATION XXYY

```
6502:  LDA $XXYY      AD YY XX

6800:  LDAA $XXYY     B6 XX YY
       LDAB $XXYY     F6 XX YY

Z80 :  LD A,(XXYY)    3A YY XX
```

Note inversion of low and high order bytes for 6502 and 6800.

3. STORE/LOAD ACCUMULATOR TO MEMORY

a) ZERO PAGE - MEMORY LOCATION 00YY

6502:	STA $YY	85 YY
6800:	STAA $YY	B7 YY
	STAB $YY	F7 YY
Z80 :	See 3. b)	

b) ANY MEMORY LOCATION XX YY

6502:	STA $XXYY	8D YY XX
6800:	STAA $XXYY	A7 XX YY
	STAB $XXYY	E7 XX YY
Z80 :	LD (XXYY),A	32 YY XX

4. LOGICAL AND WITH ACCUMULATOR

6502:	AND	29
6800:	ANDA	84
	ANDB	C4
Z80 :	AND	E6

5. ADD TO ACCUMULATOR - 8 BIT DATA

6502:	Only has ADC	
6800:	ADDA nn	8B nn
	ADDB nn	CB nn
Z80 :	ADD A nn	C6 nn

6. ADD CONTENTS MEMORY TO ACCUMULATOR

a) ZERO PAGE - MEMORY LOCATION 00YY

6502:	ADC only	
6800:	ADDA $YY	9B YY
	ADDB $YY	DB YY
Z80 :	See 6. b)	

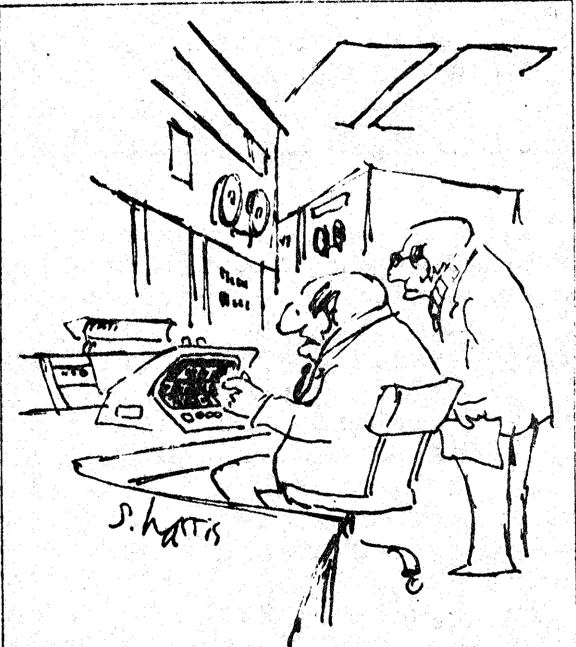
b) ANY MEMORY LOCATION XXYY

6502:	ADC only	
6800:	ADDA $XXYY	BB XX YY
	ADDB $XXYY	FB XX YY
Z80 :	ADD A,(HL)	86

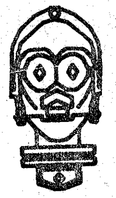
(with the implied address pointed to by the contents of the HL register)

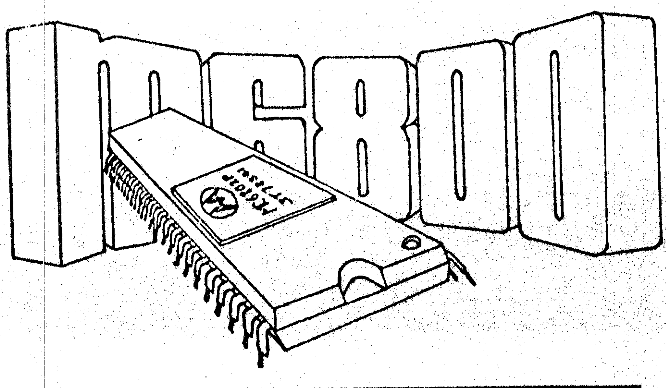
7. DECIMAL ADJUST

6502:	SED	F8 (set mode)
	CLD	D8 (clear mode)
6800:	DAA	19
Z80 :	DAA	27



". . . and in 1/10,000 of a second, it can compound the programmer's error 87,500 times!"





MICROPHILE

Neil Walsh.

At long' last Microphile is on the air again. The 6800 Group members have unfortunately been engaged in over-work, European or South African travel, mad scientific projects, or chasing women and have not been free to write their Microphile articles. However, I hope that things will run more smoothly now.

GROUP REPORT.

As mentioned before, all the existing members have machines running and are dying to find time to play around on them. At the last meeting all the guys decided that it would be a good plan to construct a 6802 to 6809 converter board so that they would have all the fancy new software available to them. This is presently in progress and should be reported more fully in the next issue. Also under investigation is the producing of a firmware BASIC language card. These things will be discussed at the next group meeting and thrown around.

DYNAMIC RAM CARD

The 64K dynamic RAM card for the C3 bus has been proto-typed and sucessfully tested. The PCB artwork is near the completion stages. Again, this will be reported in the next C3PO. It looks good so far. Of course, like any of the modules used by group,it will go through the normal testing stages before we all go mad and build many copies of it.

SOFTWARE REVIEW

These days all serious microcomputer users store their programs and files on floppy disk. For the user who has purchased one of the popular home computers, upgrading to minifloppy disk is simple and painless. However, to the home constructer the task is little short of a nightmare. The required software and controller are not easily interfaced to an existing system. A hobbyest who has constructed his machine from the ground up, will most likely have made no provision for installing a disk system to his computer. For this reason I looked upon the idea with caution because it seems a little ambitious.
When you purchase the FLEX 6809 Operating System the Image diskette plus one, just in case, comes sealed in a plastic wrapper saying "read instructions carefully before breaking this seal". Up to the time the user breaks the seal the software house, Technical Systems Consultants, promise to refund the purchaser if he gets cold feet. They go to great lengths to warn the prospective user of the assumed knowledge and skills required to get the package working. They strongly recommend that the buyer reads the manual first. The software under review is the "general" version of FLEX which has not been adapted for any particular machine. Versions already adapted can be purchased from the machine supplier and do not need formal adaption.
The very well documented file holder comes complete with a 100 page adaption guide. This document gives a detailed step-by-step list of procedures required to test the hardware. They give sample programs that will sucessfully prove a working controller card and drives. After following the extensive check list the user is ready to load the system from disk. At this stage the user is unable to boot the disk and has to key-in or casette load a special initial loader program provided. The user then goes to the cold start point using his monitor program and, if all's well, presto FLEX comes up and prompts the user for the date. The user is now advised to test his disk driver read program,which he previously wrote and tested under the guidance of the literature. He will be able to look at the disk directory and list text files on the provided disk. Now comes the chance to test the sector write drivers. If the user somehow manages in error to erase both diskettes TSC will provide a new system diskette at the price of a new floppy on proof of purchase of FLEX. However, this should not happen to the careful person. Now it is time for the user to prepare a bootable version of FLEX by using the FLEX Append command and appending his drivers to the body of the core of FLEX. Once working the disk-operating-system disk provides TSC's excellent macro 6809 assembler and editor which aids the completion of the adaption process. The user has then to adapt the floppy formatting source program on the supplied diskette with his own write track routine and to assemble

it with the above assembler. Also provided is the printer driver source program.
After finding one's way through the well written and informative guide, the new commer to TSC software need not feel lost but there is still another two to three hundred pages to wade through before being able to fully master FLEX. The Users Manual gives an ready and unambigious reference to the command format and parameters of the utilities which all work well and are reasonably fool-proof.
The Text Editor provided is TSC's well proven editor. It is as good as any of the popular line editors, providing powerful search and change commands which make it a pleasure to work with once the user has acquainted himself with it. The file management side is handled automatically by the editor,requiring the user only to specify the name of the file to be edited.
The macro assembler gives one all the facilities one expects from a good assembler. These include library functions, macros and conditional assembly and good error reporting. The assembler also has the abilitiy to assemble 6800 code with a few convenience nemonics included. However, most of the pleasure in using it comes with the ease of programming the 6809 itself, which is a powerful 8 bit processor.
The DOS section of FLEX can be called as a subroutine if required. This means that one has great ease in handling machine files even from BASIC or PASCAL This also means that utility programs can be called straight from disk, overlayed in utility RAM and executed, and then control returned to the main program. The File Management section can be called from assembled programs by passing a function code before calling FMS. This means the user can open or close, read or write, to random or sequential files with ease even from machine level programs. The System does not include a linking loader. This is because good 6809 code is position independent. The user can, though, specify a 16 bit offset that will be added to the load address and call the binary loader routine. The instructions for calling DOS FMS or any of the utility subroutines comes in an extensive Advanced Programmer's Guide which is included in the main folder.
The I/O section and command handling in FLEX handle like a dream. One can easily send program listings to the printer, to the console back into another disk text file or to any other peripheral driver routine supplied. The printer driver routine is automatically overlayed into a special RAM area the first time the user needs to print after start up. Custom-made printer routines can be automatically overlayed in the same manner. The documentation provided gives instruction on how to create printer driver system file which FLEX will call. Custom I/O routines for other peripherals can be handled with the same ease. TSC has gone out of their way to make the system usable over a wide range of applications. The general version of FLEX also gives the user advice on adapting the system to use different types and sizes of floppies with mixtures of sizes, densities, number of sides,and the inclusion of Winchester hard disk drives to the system.Again, all these things require the skill of the user but at least TSC detail all the special things one must look out for or take into account.
TSC claim that FLEX is the industry standard for 6800 users similar to CPM for non 6800 users. It is certainly widely used and all software offered by other software houses is available on FLEX compatable media. The software is friendly to the first time user and I am sure it will spoil him for any other system! The writer has been extensively reading microcomputer literature and documentation for over six years. I was immediately impressed by the standard of the documentation from TSC and continued to be ever since. Good documentation tends to give the user confidence in the product apart from the fact that the software actually works and appears to be free from the bugs that users of other systems complain of.

CRT CONTROLLER

In the previous Microphile, if you can remember that far back we promised to print the circuit diagram of the Group CRT board. We are still praying for a volunteer to lay out the artwork for this board. Also included with the diagram is the following functional description.

HOW DOES IT WORK?

The 6845 controller does most of the counting required by the circuit. You will notice that the video Ram has its own internal data and address busses. Most of the time the CRT controller chip has access to these. When the MPU requests access, the arbitration logic will grant access when the video refresh logic does not require the busses. The 74LS244 buffers on the address bus are paired so that either the MPU address bus, or the CRT controller side is switched on but never both. The MPU data bus is switched in after MPU access grant and the video RAM chips are switched over to be written into. The MPUGRANT signal is used to turn around the address bus switch the data bus in and to bring the RAM into the receive state.
To make it possible for the main computer to get in without showing any visible signs of its access there is a latch on the internal data bus. This will ignore the write data on the bus during a MPU write but will grab the data from the RAM when it is required for display purposes during video refresh. The latched data then drives a character to dot pattern generator ROM (2716). The eight bit pattern has to be split into individual bits which will then appear as dots on the screen. The para-

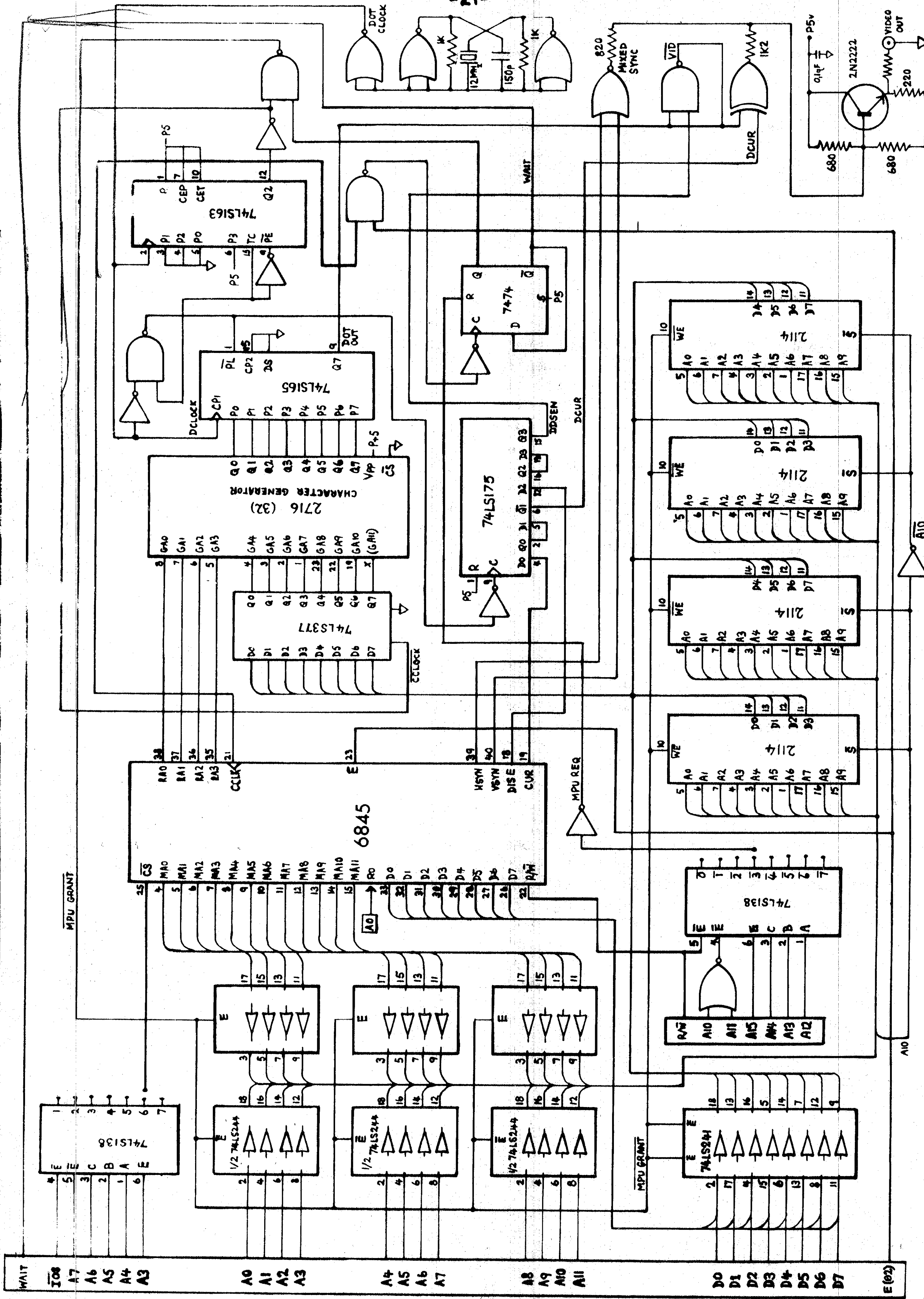
llel to serial converter is the 74LS165 chip. This is clocked by the dot frequency oscillator.
Because there are eight dots across per letter the the dot clock has to be divided by eight by the 74LS163 to derive the character clock frequency. This basic timing period is then used by the 6845 controller to generate all the other reference signals like the line and framing sync pulses which always occur after an integral number of characters displayed on the screen. The decoded terminal count of the 74LS163 is trimmed to a half a dot pulse and used to load the 74LS165 shift register for each character. This short pulse of 40 nanoseconds occurring before each character is used to clock the 74LS175 pipeline which tidies up the video display.
The 6845 is not able to take into account the delay taken by the RAM chip access.For this reason the actual video signal must be kept black for one character time to give the RAM time to catch up. Also the display must not be swiched off at the end of the line until the character has been displayed. Because of the 74LS377 data latch between the RAM and the 2716 character generator there is still a further delay of one character time. The 74LS175 quad flip-flops cause the display and the cursor to switch on and off exactly two characters late. The front and back porch of the video signals are adjusted by programming the 6845 to centre the picture.
Because the character frequency is higher then the MPU clock by about 2:3 one is always guaranteed a MPU access for each processor cycle. Because both clocks are independent of each other timing has to be asynchronous. The MPU is only allowed access during the low half cycle of the character clock. Another clock is derived by ANDing the MPU and CRT character clocks. This overlap clock is used to clock the wait flip-flop 74LS74 which goes high during the overlap period which will extend the MPU clock by 250 nanoseconds if needed, thus achieving synchronisation. The grant signal is derived by decoding the inverted character clock AND the wait signal. Normally the wait flip-flop is held in the reset state until a MPU request is made preventing unwanted wait signals.
You will notice that the MPU request is generated by the second 74LS138 chip which decodes the upper address bus and MPU write signal. Because the VRAM is write only, no read access can be granted. This may seem a disadvantage in certain cases, however, because it overlays the read-only monitor program of the 680X it means that memory space is not used up. In an improved version of the circuit the second 74LS138 is replaced by a Dynamic Address Trans fer register,which means that the absolute address of the Video RAM can be moved to overlay normal memory and give full read/write facilities under program control.
The horizontal and vertical sync pulses are mixed by an OR gate and mixed again by a resistor network with the dot signal. The dot or video signal is inverted by an exclusive OR gate during cursor display time. The cursor size and display format are automatically generated by the 6845 and can be changed under program control.
The primary signal required to make the board work at all is the 12 Megahertz dot clock. This frequency was selected to enable 64 characters of eight dots across, plus margins, to fit onto the screen. The eighth dot of each character is the blank space between letters. The 2716 pattern PROM can provide up to 16 rows per character downwards. Normal characters are only 9 dots high while the remaining rows correspond to the blank space between lines. Decending lower case or graphics characters will use up some of the lower bit positions. Using this format, one can only fit 16 lines of text on the screen. One can fit more lines in by programming the 6845 to provide less than 16 rows per character and thus compress the space between lines. You will notice that the row addresses to the dot pattern generator come directly from the 6845 CRT controller.
By dividing the 12 Megahertz clock by eight, one can calculate that the character time slot is 660 nanoseconds. Because this may be divided by 2 during a MPU access, the RAM must have an access time of 330 nanoseconds. Because the specifications given by RAM chip manufacturers represents the worst case one can usually get away with a '350' nanosecond chip.
As indicated, the prototype circuit is set up to generate a 64 by 16 line screen format using a 7 X 9 dot matrix. By calculating a different dot frequency the screen format can be easily changed by a simple hardware adaption. The dot format can also altered by changing the hard wired programming inputs of the 74LS163 character clock. It may be necessary to select faster RAM chips as well.
The 2716 PROM can provide up to 128 different charactor or graphic patterns using the present format.Its most significant address lines are used to select the respective ASCII code while the four least significant lines will count up the the dot row number.
The above circuit has been tested over many months now on two different prototype boards and works well. It gives the programmer the ability to give the user an instantaneous screen display. However, the software must leave the data in the video RAM at least long enough for the user to see it! The board could be still improved with an attribute register for poviding the video with blinking or intensified or underlined characters.

BOOK REVIEW.

The book under review here is the CRT Controller Handbook by Gerry Kane which is part of the Osborne / Mc Graw-Hill series on microprocessors. This is an essential book for anyone who wants to get to know more about one of the more important aspects of microcomputers, for it is through the video screen that the full power and versatity of of a particular system is mirrored. Graphics and instant screen updating are all part of a whole new World. The programmer has the screen under his full control. Needless to say the hardware designer must also keep up to date.
I remember when I designed my first CRT generator board, having only the vaguest idea of the timing signals required. I even wondered how the monitor could tell the differnce between a verticle or horizantal sync pulse once mixed! Unfortunately this book was not around at the time.
Before giving detailed descriptions of the different controller chips available these days,the book guides the reader through the basic principles of the CRT itself and then its timing requirements.
As with any technical book, I am sure that the new comer will have to read the paragraphs very carefully many times, before gleaning an indepth understanding. All the information is there and I am certain that it will make a good reference book.
The ideas behind the different kinds of memory addressing are dealt with. Good diagrams show how the data in video RAM will actually look on the screen. The author also shows how the dots are formed during screen scanning. Cursor forming and hardware scrolling are also important features covered. The author also goes into the different calculations required, before the design can be completed.
All the major CRT controllers are discussed in detail. Mr Kane give's his own comments and explanations of the special features of each chip. He also points out their various disadvantages which is something the component manufacturers would omit. In many of ways this book covers the subject on hand more clearly and concisely than the main manuals, because the subject is more straight forward and there is less dope to plod through.
Apart from the first few chapters of general information, the rest of the book is split into sections concentrating on each chip in turn. Also included in each section is the full specification from each manufacturer.
This book is a good reference for the designer but will be of little help to the home-constructer who is looking for circuit diagrams with no intention of understanding the design principles involved. It of course, covers the 6845 which will help those who want to understand the controller board covered in this article.
Other chips covered in the Handbook are the famous Intel 8275, the new DP8350 series, the widely used 5027 or the new Synertek 6545,which is an improved version of the 6845. These all represent the most widely used controllers in the industry.

"Not bad for a computer, but the chimpanzee's work had more feeling."

We extend our thanks to COMPUTERS ETC for their assistance in the production of this issue of C3PO.



680X CRT Controller Board

Apple Turnover

Anthony Rose

This month:
next meeting
modem revisited
lower case key entry
useless program of the month

Hello. Some confusion was evident over the arrangements for this month's meeting. It is to be held at the end of the month from now on, and this month it will take place on Wednesday 30 September, at 8:00 pm at the Athenaeum. Donald Kool will be speaking on Forth and a machine will be available (his - I hope). Those who want disks at R35 per box, bring dat boodle and we'll give you storage for the masses.

Modem, we visit again...

Last month a modem project was discussed, and I have done some investigation into the matter. Modems are beginning to be used in that smallish gold-town up north, and the standard they are using is this: 300 or 110 baud ASCII, all transmission being asynchronous, half-duplex, with 8 data bits and 2 stop bits. No parity is used. Peter Hers has constructed one which works quite well, and this costs around R100. He is overseas at present, so we cannot negotiate anything. In addition to this, it may be necessary to use a serial card, although it may be possible to drive the modem under software control, via the game paddle connector. I would like to know your ideas (on this matter).

Keyboard lower-case

If you have a lower-case adapter (eg/ Ram-It card) you will be able to print lower-case on screen. However, it is difficult to enter it into a BASIC program. I have built a lower-case adapter which fits onto the keyboard cable and lets you use the shift keys like those on a typewriter (A.R. Key-It is the name). If you do not wish to use something like this, you can use the routine presented here, so that typing shift-P will toggle between upper- and lower- case input. Commands cannot be in lower-case, but the contents of strings, print strings or REMs can. Lower-case file names can also be used on disk. Simply type in the program (or get it at the next meeting) and then CALL 768. This will only work if you have a lower-case adapter, of course. I have also written a patch to Pascal to give you lower-case throughout, but it is too lengthy to print here.

Useless program of the decade

You probably know how to produce tones on the Apple speaker, but what about random noise (white noise)? You cannot set up loops using the RND function, as BASIC is too slow. I think the shortest method is to load the accumulator from a non-existant memory location, and use this apparently random value in the loop. Addresses in the range $C080-$C0FF contain no memory and yield acceptable results. Would anyone care to write a fast random-number routine?? (FOR I=1 TO 20: PRINT "?";:NEXT). The routine listed here does an ample job on my Apple.
Random data: Rumour has it that a cheap disk drive will shortly be available for the Apple. It will sell for under R300. I wonder who could be working on it...?

```
:ASM
                   1000      .OR $300
0300- A9 03        1010      LDA /INPUT     SET INPUT
0302- 85 39        1020      STA $39        VECTORS
0304- A9 1B        1030      LDA #INPUT
0306- 85 38        1040      STA $38
0308- A9 03        1050      LDA /OUTPUT    SET OUTPUT
030A- 85 37        1060      STA $37        VECTORS
```

```
030C- A9 42     1070          LDA #OUTPUT
030E- 85 36     1080          STA $36
0310- A9 FF     1090          LDA #$FF      SET LATCH
0312- 85 FE     1100          STA $FE       FOR UPPER
0314- A9 00     1110          LDA #$00      CASE
0316- 85 FF     1120          STA $FF
0318- 4C EA 03  1130          JMP $3EA      TELL DOS
                1140
031B- 20 1B FD  1150 INPUT    JSR $FD1B    INPUT ROUTINE
031E- C9 C0     1160          CMP #$C0     IS IT '@'
0320- D0 17     1170          BNE RETURN
0322- B1 28     1180          LDA ($28),Y   OK FOLKS..
0324- 29 3F     1190          AND #$3F      HE HIT '@'
0326- 09 40     1200          ORA #$40      SO CHANGE
0328- 91 28     1210          STA ($28),Y   THE UC/LC
032A- A5 FF     1220          LDA $FF       FLAG
032C- 49 60     1230          EOR #$60
032E- 85 FF     1240          STA $FF
0330- A5 FE     1250          LDA $FE
0332- 49 80     1260          EOR #$80
0334- 85 FE     1270          STA $FE
0336- 4C 1B 03  1280          JMP INPUT
0339- C9 C0     1290 RETURN   CMP #$C0     IS IT AN
033B- 90 04     1300          BCC RETI1    ALPHANUMERIC?
033D- 25 FE     1310          AND $FE      IF SO, THEN
033F- 05 FF     1320          ORA $FF      CONVERT IT
0341- 60        1330 RETI1    RTS
                1340
0342- C9 60     1350 OUTPUT   CMP #$60     IS IT A
0344- 90 08     1360          BCC RETO1    FLASHING LC?
0346- C9 80     1370          CMP #$80
0348- B0 04     1380          BCS RETO1
034A- 09 80     1390          ORA #$80     THEN CONVERT
034C- 05 FF     1400          ORA $FF      TO NORMAL
034E- 4C F0 FD  1410 RETO1    JMP $FDF0


SYMBOL TABLE

INPUT  031B   RETURN 0339   RETI1  0341
OUTPUT 0342   RETO1  034E

:ASM

                1000          .OR $300
0300- AD 00 C8  1010 RANDOM   LDA $C800    AS RANDOM AS
0303- 2A        1020          ROL          ANYTHING ELSE
0304- 45 4F     1030          EOR $4F      I CAN GET.
0306- 85 4F     1040          STA $4F
0308- 4D 00 C9  1050          EOR $C900    MORE RANDOM
030B- AA        1060          TAX          GARBAGE
030C- 48        1070 WAIT     PHA          WAIT A WHILE,
030D- 68        1080          PLA          FOR MY EARS
030E- CA        1090          DEX          TO BE IN
030F- D0 FB     1100          BNE WAIT     RANGE
0311- 2C 30 C0  1110          BIT $C030    CLICK SPEAKER
0314- 2C 00 C0  1120          BIT $C000    KEY PRESSED?
0317- 10 E7     1130          BPL RANDOM
0319- 2C 10 C0  1140          BIT $C010    CLEAR STROBE
031C- 60        1150          RTS


SYMBOL TABLE

RANDOM 0300    WAIT    030C
```

Thought for the month (Z80 group style): How would you like to be an electron?

Z80 Group Report

JOHN WHITCHURCH

Last month's Group Meeting was held at the Computer Shop, thanks to Derek Sherlock, who also managed to conjour up a seemingly unending supply of Boschendal Le Bouquet. I am beginning to wonder if the Group Members are subtly trying to tell me something. Okay, okay, wait for the warmer weather and we will do a return to the wine route.

John Vallence had been doing a great deal of leg work around Cape Town since the previous meeting, trying to find a supplier of the K2FDOS and Practical Microcomputer Programming Manuals, unfortunately as yet, without success. The group is determined to initiate its library with the latter book and will order the same direct from the States as soon as we have the supplier's address. John has lately been burning the midnight oil trying to get his SABUS system running, also without success. However, we hope to get that problem sorted out in the very near future.

The Group briefly discussed the Telephone Modem Project and Derek kindly volunteered to look out a possible suitable design together with estimated costs and parts availability. The design he has in mind runs at 300 Baud and is compatible with S.A.P.O. standards. We were also treated to a demonstration of a micro/mainframe telephone line hookup, which was only partially successful, but certainly gave one an idea of what could be achieved.

Official business being over, the Group turned to matters dearest to the heart. (Fill up my glass please, Derek.) Peter Llewellyn is still busy with his DOS, and I, for one, am eagerly awaiting his forthcoming article in C3PO. Dave Gargan is nearly ready to start brandishing multimeter and logic probe. He only now requires an I/O card with serial output (see last month's C3PO). My own rig will shortly be able to program 2708 and 2716's, provided I was sober when drawing up the logic. Derek mentioned Tony Putman's discovery of a bus conflict on the SABUS RAM card. How about a short explanatory article on that for C3PO, Tony?

To conclude the evening we were entertained by Space Invaders, Blitzkrieg and Sargon II.

"It's analyzed our situation thoroughly, and has concluded that our business doesn't need a computer."

Atomic Matter

Peter Reber

Before I go on talking about our BASIC I want to give the first 'status report' of our group. We have 3 members at the moment and all own a fully expanded ATOM. I modified my machine and I have an additional 3k internal RAM as well as an 8k RAM card connected via the expansion connector. Two members also have a printer and the third one found an used teletype recently. We hope to have an EPROM programmer soon, so that we can store our range of utility programs permanently. This will make the writing and debugging of programs much easier. The EPROM will reside at A000H and we will use the 4k EPROM's so that we still have some space left for improvements and new utilites. Now we return to BASIC.

ACORN BASIC program listings contain very often a few lower case letters. These are labels and are most often used to give the destination addresses of GOTO's and GOSUB's. Valid labels are a-z. They appear as inverted upper case letters on the screen and are easily identified. A short illustration of its use:

```
  10sPRINT"*"
  20 GOTO s
```

The label must follow the line number immediately and no spaces may be inserted. The advantage of using labels is an increase in speed. The label addresses are held in a table. Therefore the machine does not have to search for the line number but looks up the address in the table. This program from the manual produces a tone with a frequency of 187 Hz.

```
  10 P=#B002
  20a ?P=?P:4;GOTO a
```

If line 20 is written

```
  20 ?P=?P:4;GOTO 20
```

the frequency lowers to 144 Hz because the GOTO 20 statement takes longer to execute.

ACORN BASIC allows the abbreviation of many of the recognized words. Abbreviated words must end with a full stop and I will give the shortest possible form of each in brackets. You will notice that a few words have the same abbreviations. In these cases the context will always make it clear which word is meant. I will only elaborate on them if they differ from MICROSOFT BASIC.
The following words 'behave' normal:

LIST	(L.)
LOAD	(LO.)
NEW	(N.)
ABS	(A.)
LEN	(L.)
END	(E.)
FOR	(F.)
TO	
STEP	(S.)
NEXT	(N.)
GOTO	(G.)
GOSUB	(GOS.)
LET	(omit)
REM	
RETURN	(R.)
RUN	
SAVE	(SA.)
IF	
THEN	(omit or T.)

If the statement following the THEN is the operator '?','!' or begins with a T, a delimiter such as T. MUST be used.
The following 3 words do not exist in ACORN BASIC

DATA
READ
RESTORE

However, it is possible to produce equivalents with a short BASIC routine.

PEEK and POKE
Our BASIC provides a much more elegant mechanism to implement these two functions than other BASIC's. It uses the 'query' operator '?', which is interpreted as PEEK or POKE depending on the context.
    ?80=13   POKE's 13 into memory location 80 (decimal)
  PRINT ?80 PEEK's at memory location 80 and prints its contents
    ?80=?82 The '?' on the left of the equal sign is interpreted as POKE and the one on the right as PEEK.
We can also use variables to hold the PEEK and POKE addresses. In fact, the use of the '?' operator for these two

functions is just a part of a very useful concept, called 'byte vectors'. More about them later (just to confuse you a bit more, there are 'word vectors' too).

OLD
Every microcomputer should have such a statement. It recovers a program after it has been NEW'd out or after hitting the BREAK key. BREAK is connected to the CPU's RESET pin. This key is essential in machine code routines since it is often the only means to get out of an endless loop (another way is obviously to switch the power off). BREAK then (re)initializes the machine and the program is 'lost'. IT can then be recovered by OLD.

PRINT (P.)
Our BASIC does NOT give a newline after a PRINT statement unless specifically instructed to do so! The newline sybmbol is an apostrophe "'". Note that we have automatic wrap-around after 32 characters have been printed on a line. Therefore, PRINT "THIS ISAN EXAMPLE" may be correct if the S from IS is the 32nd character on the line and we will get a newline even without "'".
One of the features I like most in our BASIC is that we can input and print numbers in hexadecimal. The & (ampersand) operator forces numerical printout in hex until the next comma.

```
  PRINT &A B,C D
```

will print A,B in hex and C,D in decimal.
The # (hash) operator denotes the start of a hex number.

```
  PRINT #A0 will print out 160
  PRINT &10 20 30  prints out
  A  14  1E
```

And the following

```
  PRINT &?#80
```

will PEEK a memory location 80 hex and print its contents in hexadecimal.
Numerical values are printed out righthand justified in a field whose width is defined by the variable '@'. Default value of @ is 8, so that numbers would be printed in four columns.

```
  PRINT $A; PRINT $A+4
```

$ introduces a pointer to a string. The expression following the $ symbol is evaluated and the string pointed to by that value is printed out.
However, if the value of the expression is between 0 and 255, the ASCII character corresponding to that value will be printed.

```
  PRINT $65  will print an 'A'.
```

INPUT (IN.)
It can only use the 'simple' variables and inputing directly into arrays is not possible. If a numerical value is preceeded by # it is evaluated as a hex number.

DIM
All strings and arrays must be dimensioned. There is no default case.

CH
Change character to number.

COUNT (C.)
This function returns the number of characters printed since the last return. COUNT is useful for positioning table elements.

RND (R.)
This returns a random number between -2147483648 and 2147483647. Positiv random numbers can be obtained with the ABS function and numbers within a certain range can be produced with the remainder operator '%'.

```
  ABSRND%10
```

will return a positiv random number between 0 and 9.

DO...UNTIL (DO...U.)
Do something UNTIL a testable condition is true. This is another way of producing loops.

```
  DO...UNTIL 0
```

means DO forever.

OR and AND (A.)
This provides the logical OR/AND operation between two testable expressions. It is most often used after IF or UNTIL statements.

WAIT
This statement waits until the next vertical sync pulse appears from the VDG. It has two uses. First to provide a time delay of 1/60 of a second (the 6847 VDG works to the NTSC standard, although ATOM's sold in South Africa are converted to 50 Hz operation), and

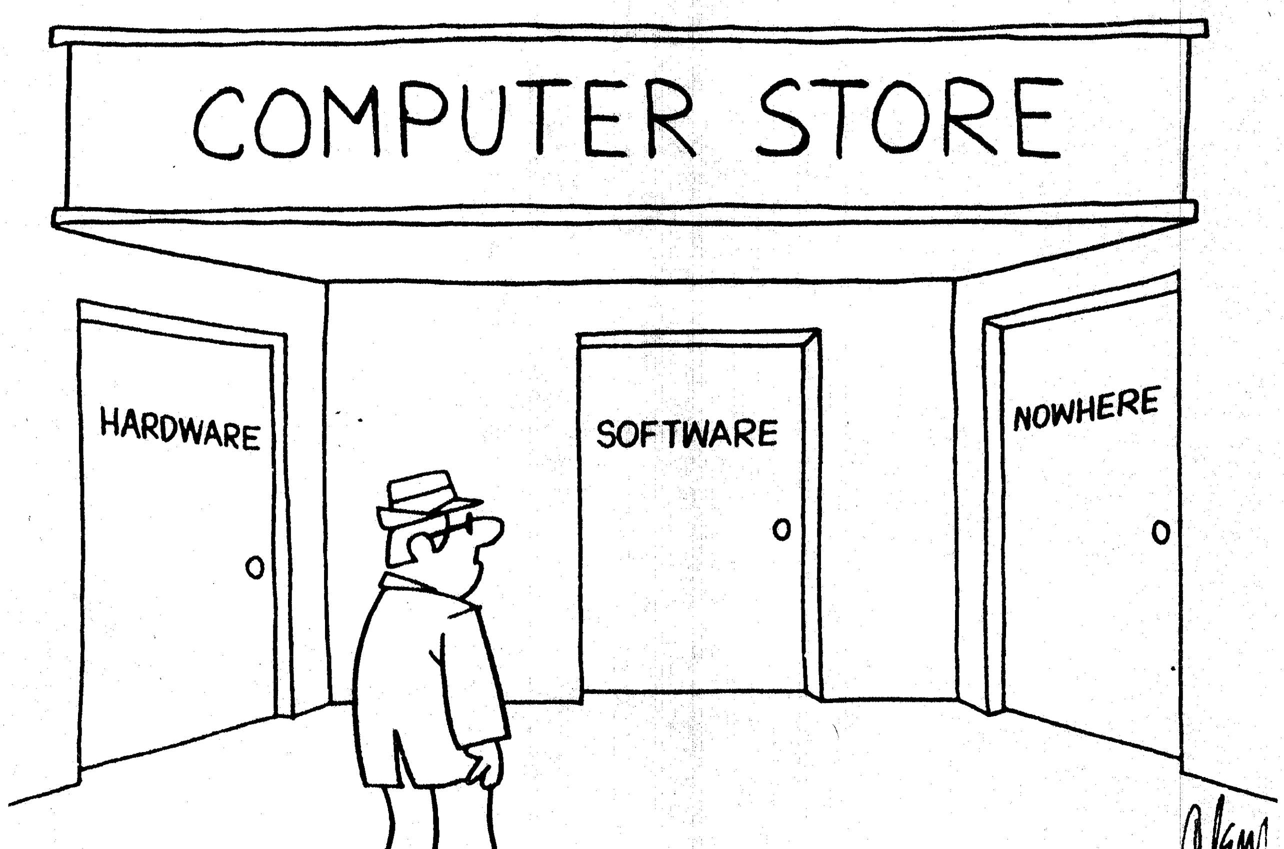
to wait until flyback, so that a subsequent graphics command does not cause noise on the screen.

LINK (LI.)
The least significant bytes of the variables A,X and Y are loaded into the accumulator,X- and Y-register respectively, and control is transfered to the machine code routine whose address is specified in the argument. Return to BASIC is via a RTS instruction.

TOP (T.)
Returns the address of the first free byte after the end of a BASIC program.

UNUSUAL OPERATORS

#	denotes a hexadecimal number
%	A%B returns the signed remainder of the division A/B
&	1) forces hex printout of numbers 2) produces a bitwise logical AND between two 32 bit words, each bit is a logical AND between corresponding bits of the two operands
:	produces a bitwise logical EXCLUSIVE-OR between two 32 bit values
⍂	(inverted backslash) produces a bitwise OR between two 32 bit words
?	byte indirection
!	word indirection



TRS-80 Group Report

IAN W. MCQUEEN

** GROUP MEETING **

ANOTHER MEETING OF THE GROUP WAS HELD AT THE PREMISES OF NASIONALE PERS ON AUGUST 27 AND WAS ATTENDED BY MORE THAN 25 OF THE GROUP MEMBERS.
THE USUAL ARRAY OF EQUIPMENT CONFRONTED THOSE ARRIVING WITH EXAMPLES OF THE MODEL III (THANK YOU DEREK) AND MODEL I WITH LARGE SCREEN VDU AND SOUND (THANK YOU BAREND) BOTH COMPUTERS OF COURSE CHATTERING AWAY WITH A SELECTION OF GAMES AND EVEN A LITTLE PRACTICAL SOFTWARE.
DEMONSTRATIONS WERE GIVEN OF TWO DISC INDEX PROGRAMS. ONE LOCALLY WRITTEN WHICH PROVED SIMPLICITY ITSELF TO USE, AND ONE VERY SOPHISTICATED AMERICAN SYSTEM WHICH HAD BEEN GREATLY ENHANCED BY THE LOCAL ADDITION OF MACHINE LANGUAGE SORT ROUTINES.
FRANCOIS LEBLOND, WHO CAME SPEEDING BACK FROM A BUSINESS APPOINTMENT IN STELLENBOSCH, GAVE SOME EXAMPLES OF THE APPLICATION AND OPERATION OF THE VISICALC PACKAGE. I AM SURE MANY GROUP MEMBERS, MYSELF INCLUDED, ARE NOW LOOKING FORWARD TO TRYING THIS PROGRAM IN SOME PRACTICAL APPLICATIONS.
FOLLOWING THESE DEMONSTRATIONS THE USUAL FREE DISCUSSION DEVELOPED AND DESPITE RUMOURS TO THE CONTRARY IT WAS STILL FELT BY THE GREAT MAJORITY OF MEMBERS PRESENT THAT FORMAL MEETINGS SHOULD CONTINUE ON THE CURRENT THREE MONTHLY BASIS AND IT BECAME OBVIOUS THAT THE FREQUENCY OF CASUAL MEETINGS OF GROUP MEMBERS INTERESTED IN PARTICULAR ASPECTS OF MACHINES OR SOFTWARE WAS INCREASING TO SATISFY INDIVIDUAL NEEDS BETWEEN FORMAL MEETINGS.
THE FORMAL MEETING ENDED AT 1930 AND AFTER MUCH CHIN-WAGGING AS USUAL BETWEEN 80 ADDICTS THE ROOM WAS PROBABLY CLEAR BY ABOUT 2100.
ONCE AGAIN THANKS TO NASIONALE PERS / BAREND VAN AS / FRANCOIS LEBLOND / AND OF COURSE DEREK AND M M ELECTRONICS FOR SPONSORING OUR NEWSLETTER AND PROVIDING THE BEVVIES.

** CAPE COMPUTER CLUB **

DURING THE USER GROUP MEETING YOURS TRULY WAS NOMINATED TO REPRESENT THE USER GROUP ON THE COMMITTEE OF THE CAPE COMPUTER CLUB AND I WILL THEREFORE IN FUTURE BE REPORTING CLUB DEVELOPMENTS BACK TO THE GROUP AND VICE-VERSA WHICH I AM SURE WILL BE TO THE MUTUAL BENEFIT OF BOTH THE GROUP AND THE COMPUTER CLUB. ALREADY MORE GROUP MEMBERS ARE ALSO BECOMING MEMBERS OF THE CAPE COMPUTER CLUB AS OUR INTERESTS CONVERGE. FOR EXAMPLE IF YOU RECEIVE COPIES OF C3PO YOU WILL PROBABLY BE SEEING DETAILS OF A NEW DEVELOPMENT IN THE CLUB - A CLUB PROJECT TO BUILD INEXPENSIVE TELEPHONE "MODEMS" FOR AMATEUR USE. GROUP MEMBER RICHARD STRATFORD WILL BE LIASING WITH THE PROJECT ORGANISER ON OUR BEHALF TO ASSIST IN DEVELOPMENT OF A TRS-80/APPLE/HOMEBREW COMPATIBLE PIECE OF HARDWARE. MORE NEWS LATER !!

** DOSPLUS FOR MODEL III **

IT SEEMS THAT MORE AND MORE SOFTWARE IS BEING SEEN LOCALLY FOR THE MODEL III. LATEST TO ARRIVE IS THE DISC OPERATING SYSTEM "DOSPLUS" WHICH HAS THE REPUTATION OF BEING SUPERIOR TO "NEWDOS 80" AND DESIGNED TO OPERATE WITH MODEL I OR MODEL III SYSTEMS IN SINGLE OR DOUBLE DENSITY MODES. MAYBE ONCE IT HAS BEEN EXPLORED ONE OF OUR MODEL III USERS CAN GIVE US A REVIEW.

** FORTH **

GROUP MEMBER STEPHEN DAVIES HAS BEEN DOING QUITE A LOT OF WORK WITH FORTH AND SEEMS TO BE ENJOYING HIMSELF - IF I COULD UNDERSTAND WHAT HE IS TALKING ABOUT I'D TELL YOU BUT I AM STILL STRUGGLING - INTERESTED GROUP MEMBERS WILL FIND THE AUGUST 1980 COPY OF BYTE MAGAZINE WHICH WAS DEDICATED TO THE IMPLEMENTATION OF THE FORTH LANGUAGE MOST INFORMATIVE.

** SIEMENS DISC DRIVES **

IT WOULD APPEAR FROM SOME RECENT EXPERIENCES WITH THE ABOVE DRIVES THAT THEIR TERMINATION VALUES ARE (AND NEED TO BE) DIFFERENT FROM THOSE NORMALLY USED BY RADIO SHACK AND MPI DRIVES. WHILST TRYING TO GET A SYSTEM USING A COMBINATION OF BOTH TYPES TO OPERATE VERY STRANGE THINGS WERE HAPPENING !!!.
DISCS WERE BEING OVERWRITTEN AND DIRECTORIES WERE DISSAPEARING - MOST DISTURBING TO SAY THE LEAST. WE DON'T YET KNOW A SOLUTION BUT IF WE FIND ONE I WILL LET YOU KNOW. IN THE MEANTIME - BE WARNED.

** MORE MOD III'S **

IT WOULD APPEAR ANOTHER SHIPMENT OF MODEL III TRS-80'S HAS ARRIVED IN CAPE TOWN WITH QUITE A FEW NEW MEMBERS APPLYING TO JOIN THE GROUP. THE NEW MACHINES SEEM TO BE ALL FITTED WITH 48K OF MEMORY AS OPPOSED TO SOME OF THE EARLIER SHIPMENTS THAT WERE ORIGINALLY FITTED WITH ONLY 32K EX USA AND MODELS WITH 2 DISC DRIVES ARE PROBABLY GAINING THE MAJORITY. ITS LOOKS AS IF IT WILL NOT BE TOO LONG BEFORE THE MODEL I OWNERS ARE OUTNUMBERED.

** EXPANSION INTERFACES **

A FEW MORE EXPANSION INTERFACES FOR MODEL I MACHINES HAVE BEEN PURCHASED BY VARIOUS GROUP MEMBERS AND I SUSPECT THAT A FEW MORE DISC DRIVES WILL PROBABLY FOLLOW. AN UPDATED USER GROUP LIST IS BEING PREPARED WITH SYSTEM DETAILS - SO IF YOU HAVE ANY CHANGES LET ME KNOW.

** TELETYPE MACHINES **

ONE OR TWO TELETYPE MACHINES ARE CURRENTLY BEING OFFERED TO GROUP USERS. THESE MACHINES NORMALLY OPERATE WITH RS 232 CONNECTIONS BUT IT IS POSSIBLE TO INTERFACE THESE PRINTERS VIA THE CASSETTE PORT OR BY BUILDING A SERIAL INTERFACE. IF THE FIRST ALTERNATIVE APPEALS HAVE A LOOK AT SOME BACK ISSUES OF 80 MICROCOMPUTING FOR DETAILS. THE WRITER AT ONE STAGE ALSO USED A TELETYPE WITH A HARDWARE INTERFACE. IF YOU WOULD LIKE INFO ON THAT GIVE ME A RING.

** PERCOM DOUBLER **

THE PERCOM DOUBLER TOGETHER WITH SUITABLE OPERATING SYSTEM ALLOWS THE MODEL I TO STORE DATA ON DISC IN DOUBLE DENSITY FORMAT AND WITH SOME DOS'S PERMITS EXCHANGE OF MODEL I AND MODEL III DISCS. FRANCOIS LEBLOND HAS ONE ON ORDER AND I AM SURE WHEN IT ARRIVES WE WILL BE ABLE TO PERSUADE HIM TO GIVE US AN UNBIASED APPRAISAL. AT THE SAME TIME I HAVE HEARD A RUMOUR THAT ONE OF THE MEMBERS OF THE JOHANNESBURG TRS-80 USER GROUP IS BUILDING HIS OWN DOUBLER MODIFICATION. I HAVE ON BEHALF OF THE CAPE TOWN GROUP EXPRESSED INTEREST IN DEVELOPMENTS AND WILL KEEP YOU INFORMED.

** CALENDAR PROGRAM **

INCLUDED WITH THIS NEWSLETTER IS A PRINTOUT OF A VERY SUPER PROGRAM THAT WILL PRINT AN IMPRESSIVE CALENDAR FOR ANY YEAR OF YOUR CHOICE AND UNLIKE SOME COMMERCIAL PROGRAMS I HAVE SEEN ACHEIVES, I AM TOLD TOTAL ACCURACY. (TRY THE YEAR 1900 FOR EXAMPLE - THIS ONE COMMONLY FOOLS SOME SOLUTIONS). THIS PROGRAM WAS WRITTEN BY DAVE FRANCIS WHO BY THE WAY IS DETERMINED TO PI ON HIS TRS-80. WATCH THIS SPACE !!!.
THANKS DAVE.

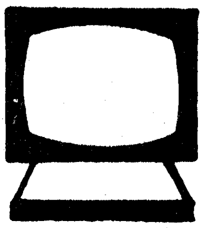
WELL THAT'S IT FOR NOW - MORE NEWS WHEN IT HAPPENS

```
10 DIM A$(11), S$(24)
20 PRINTTAB(21), "CALENDAR"
30 PRINTTAB(21), "========"
40 LET S$ = "************************"
50 PRINT "TYPE IN YEAR REQUIRED EG. 1976 THEN PRESS RETURN"
60 INPUT Y
70 IFY<>INT(Y)THEN50
80 IFY<1752THEN50
90 IFY>4902THEN50
100 '***FIND DAY OF WEEK OF JAN.1ST.
110 LETY1=INT((Y-1)/100)
120 LETY2=Y-1-100*Y1
130 '***THIS ZELLERS CONGRUENCE
140 LETD=799+Y2+INT(Y2/4)+INT(Y1/4)-2*Y1
150 LETD=-(D-(INT(D/7)*7))
190 LPRINTTAB(25); "CALENDAR FOR"; Y
200 LPRINTTAB(25); "======== === ===="
210 LPRINT
220 LETL=0
230 '***CHECK FOR LEAP YEAR
240 IF((INT(Y/4)*4)<>Y)THEN280
250 IF((INT(Y/400)*400)=Y)THEN270
260 IF((INT(Y/100)*100)=Y)THEN280
270 LETL=1
280 FORN=1TO12
290 LPRINT
300 LPRINT
310 READ A$, M
320 DATA "* JANUARY *", 31, "* FEBRUARY ", 28, "** MARCH **", 31
330 DATA "** APRIL **", 30, "*** MAY ***", 31, "*** JUNE **", 30
340 DATA "*** JULY **", 31, "** AUGUST *", 31, " SEPTEMBER ", 30
350 DATA "* OCTOBER *", 31, "* NOVEMBER ", 30, "* DECEMBER ", 31
360 LPRINTTAB(4); S$; A$; S$
370 LPRINTTAB(8); "SUN     MON     TUE     WED     THU     FRI     SAT"
380 LPRINTTAB(4); "************"; S$; S$
390 IFN<>2THEN410
400 LETM=M+L
410 FORI=1TO6
420 LPRINT
430 FORJ=1TO7
440 LETD=D+1
450 IFD>MTHEN520
460 IFD<=0THEN480
470 LPRINTTAB(J*8); D;
480 NEXTJ
490 LETJ=1
500 IFD=MTHEN520
510 NEXTI
520 LETD=1-J
530 LPRINT
540 NEXTN
550 FORI=1TO6
560 LPRINT
570 NEXTI
590 END
```

A great team. You and . . .

COMPUTERS PROSPUR

A BUSINESS SYSTEM
from **R120 p.m.**



AGENTS FOR:

- Hewlett-Packard
- Commodore Pet
- Apple

Ask us about

- Computer Consultants
- Systems Analysis
- Business Computers
- Programme Specialists
- Home Computers
- Computer Peripherals
- Servicing (Hardware and Software)
- Calculators
- Computer Games

Distributors of the world's most famous name machines!

31 UPPER PARKADE LEVEL, STRAND ST,
CAPE TOWN 8001. TELEX 57 6539.
PO BOX 117, CLAREMONT 7735, SOUTH AFRICA.
PHONES 41 2792, 41 2795.

COMPUTERS PROSPUR

WOULD YOU PURCHASE YOUR COMPONENTS FROM SOMEONE LESS TECHNICAL THAN HAMRAD?

STOCKISTS OF:

SINCLAIR ZX-80, ZX 81 MICROCOMPUTERS
ACORN ATOM MICROCOMPUTER
TONO COMMUNICATIONS COMPUTERS
YAESU 12" MONITORS
TONO HC 800, HC 900 PRINTERS
SEIKOSHA GP 80 PRINTERS

AGENTS FOR:

SOLDERING EQUIPMENT
electronica de precision, s.a.

INDICATOR LAMPS
L.E.D.'s
SWITCHES
CSC BREADBOARD

NANA COAXIAL CONNECTORS
LED's
AUDIO CONNECTORS
Hansen TEST EQUIPMENT

DISTRIBUTORS OF:

EDAC COMPONENTS
SIEMENS COMPONENTS
ROSE ENCLOSURES
molex® CONNECTORS

STOCKISTS OF A WIDE RANGE OF COMPONENTS, INCLUDING . . .

- **TRANSFORMERS**
- **PANEL METERS**
- **RESISTORS**
- **CAPACITORS**
- **PRINTED CIRCUIT BOARDS**
- **PUBLIC ADDRESS EQUIPMENT AND ACCESSORIES**
- **INTERCOMS**
- **SEMI-CONDUCTORS**
- **MULTI-TESTERS**
- **LOUDSPEAKERS**
- **HI-FI**
- **TV**

HAMRAD

CAPE TOWN (Head office): 114 Buitengragt ☏ 43·5264
DURBAN: 21 Aliwal St. ☏ 6·6578
JO'BURG: 123 President St. ☏ 37 3626
PRETORIA: 57 Esselen St., Sunnyside ☏ 42·4015

The Technical Dealer