80-BUS NEWS

JANUARY – MARCH 1982 VOL. 1 ISSUE 1

£1.50

The Magazine for NASCOM & GEMINI USERS

JANUARY—MARCH 1982

80-BUS NEWS

VOL.1 ISSUE 1

CONTENTS
========

 Published by Interface Data Ltd and printed by Excelsior Photoprinting Ltd., High Wycombe.

SUBSCRIPTIONS

Annual Rates (6 issues)	UK	£9	Rest of World Surface	£12
	Europe	£12	Rest of World Air Mail	£20

Subscriptions to 'Subscriptions' at the address below.

EDITORIAL

Editor : Paul Greenhalgh Associate Editor : David Hunt

Material for consideration to 'The Editor' at the address below.

ADVERTISING

Rates on application to 'The Advertising Manager' at the address below.

ADDRESS: 80-BUS News,
Interface Data Ltd.,
Oakfield Corner, Sycamore Road,
Amersham, Bucks, HP6 5EQ.

EDITORIAL

Welcome to 80-BUS News.

This is the first issue of a magazine which is to be published six times a year. The aim of 80-BUS News is to provide owners of Nascom and Gemini systems with a comprehensive guide to the equipment that they own, and to provide information on the extremely wide range of further hardware and software products which are available for their machines. There are already some 10 manufacturers producing over 20 Bus-compatible cards, which must make the 80-BUS the most widely supported Bus structure by British manufacturers.

Each issue of 80-BUS News will provide a blend of hardware and software reviews, articles on hardware construction and on programming, a number of tips and hints, letters received from readers, and hopefully some occasional light relief - all with the aim of assisting both Nascom and Gemini owners to obtain the maximum use and/or pleasure from their chosen equipment.

80-BUS News also incorporates INMC80 News. This was a somewhat irregular magazine originally set up to promote discussion relating to Nascom equipment. With the availability of a very wide range of compatible product from sources other than Nascom, and with the arrival of the Gemini processor cards that can use many of the same expansion products as the Nascom, it has become much more appropriate to produce a magazine that is allied to the 80-BUS/Nasbus bus structure as opposed to the one specific manufacturer.

A major objective of 80-BUS News is to act as a central discussion point for Nascom and Gemini users. The contents of the magazine will reflect the range of articles submitted or requested by the readers. If you believe that there is not enough coverage of any specific topic, then YOU can correct this by sending in YOUR articles and letters.

In this issue you will notice that a lot of space is devoted to disk systems. The proportion of space dedicated to disks reflects the balance of articles received. I summise that it is mainly the very dedicated computer users who write articles and send them in, and that these same people are the ones busy expanding their systems to include disks - thus there are many articles about disks. I also recognise that a very large number of readers, although possibly interested to read about how their computers can be upgraded in this way, may not want to spend such a large amount of money on what may well be a hobby. So....it's bribery time! In contrast to INMC80, 80-BUS News will make nominal payments for every single word published (apart from Adverts - I don't think that's quite on!!) The writer of any letter published in future issues will receive £3, and the author of any article or program £10, per published page. I hope that this gesture will encourage many people who have been sitting quietly on the sidelines to come forward and start making noises about why their system is so great, even with (or because of!) no disk drives.

Advertisers in 80-BUS News are also very welcome, on two counts. Firstly the revenue from adverts helps to keep our cost down and thus will hold back magazine price rises, and secondly the reader gets the opportunity of seeing the availability of some of the vast array of products that are compatible with their chosen Nascom or Gemini system. Potential advertisers please write for details of our ridiculously low rates.

Well that's all from me for this issue. If you like or dislike what you see, please let me know. I look forward to receiving your comments and articles.

Dear Ed.

RE. HISOFT PASCAL

I read with interest the review of the Hisoft Pascasl compiler in the May/September issue of INMC80.

It is indeed a very nice piece of software but unfortunately the writer of the compiler has done certain things which are not mentioned in the documentation but which are nevertheless extremely important when using the compiler for anything more than playing about. For example, the compiler workspace is quite different from that stated in the documentation and this has lead us in wasting approximately two weeks of precious time, and this is only part of the story. However Hisoft have been very careful not to publish their telephone number and, not surprisingly, they are not listed in the telephone directory and their telephone number is not available from directly enquiries. This and the fact that they tend not to answer letters makes such a complex piece of software as a compiler rather useless for applications where questions have to be asked which are not answered by the documentation.

I would be very grateful if you can prompt them to telephone me during office hours so that I can ask Hisoft a few questions.

Finally, I would like to say that your magazine is very good indeed and hope that you keep up the good work.

P.Holy, Ringdale Engineering Co. Ltd. (09063-64423)

BITS & P.C.'S TOOLKIT

If you want to change the speed of the keyboard repeat on the Bits and P.C's toolkit this can be done by poking the location 3120 (0C30) with the value for the speed of repeat, and locations 3118-3119 (0C2E-0C2F) for the delay before repeating starts. The original values are 50H in 3120 and 0280H in 0C2E-0C2F. When the toolkit is reset the original values are placed in these locations. It seems that this keyboard repeat is similar to the one Richard Beal published in INMC issue 6.

A. Sircombe, Tewkesbury, Gloucester.

ERROR IN LIFE

I am delighted to see myself in print (conversion of Life from Nasbug to Nas-Sys) but I have noticed a small error; the patch at 0E64 should not be there at all. Oops sorry.

May I comment on Stephen Prince's letter on Transparent Video. I have just added the Snowdinger from the first issue of Micropower. This has not only given me (and my Nascom 1) a snowfree screen without black flashes but also where my screen previously was obviously made up of a number of distinct squares, now it is a uniform colour. I hope this is of some help and I am sure that Program Power will be only too pleased to provide back issues.

P.S. I am making progress with Robots, although this has been interrupted by my own "Tiny Adventure" and housing my Nascom and a Supermum in a case instead of spreading it across the floor.

W.H.T., New Malden, Surrey

MISSING STAR

My system is a Nascom 1 with 64K RAM & Microsoft (ROM) basic interpreter. I have recently found something funny and I don't have any explanation about a M.S.S. (Missing Star Screen) phenomenum. This little program will show you the problem:

```
10 CLS
20 FOR  Y=1 to 15
30 FOR  X=1 to 48
40 SCREEN  X, Y
50 PRINT "*";
60 NEXT X, Y
70 GOTO 10
```

This program is self explanatory but it does not work !

Philippe Wetterwald, France

(Ed. - It is all to do with the SCREEN command not resetting BASICs counter for the number of characters in a line, so every so often BASIC says `Ah, time for a line feed.' Try setting WIDTH 255 to disable line feeds.)

ROM/RAM ZEAP, TAPES & DISKS

On page 28 of INMC80 issue 3 Mr. C. Bowden is having some problems with running ROM ZEAP in RAM.

This is cured by replacing the contents of location D5F0 (which normally contains the code INC (HL) with a hex 00 (code for NOP). This disables the memory `test' routine which arranges for ZEAP to crash when run in RAM.

Of course, you may not like printing the above but I think that anyone with the infinite intelligence required to get CP/M running should be able to work it out in no more than 15 minutes.

On page 8 of INMC80 issue 4 Mr. R.S. Marshall writes about poor quality recording tapes. I feel that the problem is basically two-fold: the so-called `computer' tapes (i.e. C12 long etc) are of lower quality than practically any common audio tapes. Certainly their mechanics are very poor and the only good reason for using them (rather than audio tapes) is their lower price - if they work.

The other reason is that the data density at (say) 1200 baud is not far from 1 bit/0.001" and most commonly used (i.e. cheap) cassette recorders do not have their head azimuth adjusted to this degree of accuracy. If (as is often the case) the tape is recorded on one machine and replayed on another, then unless the heads of the two machines are identically set-up to within 0.0005" or better (over the tape width) the data will be unreadable. This particular problem can be eliminated by replaying the tape (which was probably recorded in stereo with the same data on both channels) on a stereo deck and only connecting one channel.

I myself have been using TDK D-C60 & D-C90 tapes for about two years and on average get about one error per several hours (yes) of playback time, at 1200 baud. The deck is a Pioneer CT-200, costing about 85.00.

Regarding David Parkinson's "1.5 Mbaud" article on page 41 (INMC80-4) I would be very interested to hear more about his 8" Nascom disk system, particularly how he wrote his BIOS. Also, does he still have working versions of MOVCPM.COM and SYSGEN.COM? It seems to me that as soon as the BIOS is modified (for 8" drives) then the standard MOVCPM cannot any longer relocate the CP/M package for different system sizes. Also, which format did he chose?

To the editor of this excellent mag: can you accept contributions on Gemini 5.25" disks with ED files, and return the disks afterwards? (Yes thanks, but Pen disks would be nicer! - Ed.)

Finally, with so many talking about Nascom 1s running at 4 MHz ("they work a lot more often than you would think"), has anyone tried running a Nascom 2 at 6 MHz, perhaps with 'B' chips?!!!

P. Holy, Worthing

TRY THIS INSTEAD

I have only just seen a copy of INMC 80 No.5. and I read with interest the article by P. Forrester, but why make the routine decoder so complicated? Why not just use the address of the target routine as the USR argument? Then if the decoder is at 0C80:-

```
0C80   CD 8B E9 ;  Get argument in DE
0C83   EB       ;  put into HL
0C84   E9       ;  jump to this address
```

The decimal value must be used - the only problem would be if the routine was above 8000H when the appropriate negative value would have to be used. If your routine is at 0C90H (or 3216 decimal) then it can be called by A=USR(3216), if at 8000H by A=USR(-32768). Both Mr Forrester's and the above mean variables must be passed by POKE or DOKE - or does it? It is quite easy to follow the call with arguments and then pick these up in your machine code. Passing fixed numbers is easy but Basic variables slightly more complex. If anyone is interested send an SAE to the address below and I'll send details.

Bill Lounds, Cotcliffe Villa, Sunnyside Rd, Ketley Bank, Telford, TF2 0DT.

BLS PASCAL = NASCOM PASCAL

I read with great interest the review of our Blue Label Software Pascal in INMC80-5, and I must say that I find it to be accurate in every detail. However, in the past eight months some changes have occurred to the package, both in terms of modifications and additions, as well as to distribution and pricing. Therefore this letter is to keep readers up to date.

As from December '81 distribution was taken over by Lucas Logic, which means that the package should now be available through most (if not all) Nascom dealers. At the same time the name was changed to Nascom Pascal, and the prices dropped a bit to 45.00 for the tape version, and 70.00 for the EPROM version.

Nascom Pascal is almost identical to the version of BLS Pascal described in the review. The differences are the following additions: Port I/O is supported through a predefined procedure called OUT, which sends a byte to a port, and a predefined function called INP, which reads a byte from a port. CODE procedures and functions give the ability to define machine code subroutines within a program through a list of constants (in hex or decimal as appropriate) representing Z80 instructions, and hex integers can be input in READ and READLN statements.

In the review Mr O'Farrell expresses his wish that the package, in a future version, should include sets, user defined types and records. I do, of course, agree that the inclusion of these features would result in an even better subset of Pascal, but it would also increase the memory overhead by say 4-8K bytes, and I am not sure whether the equivalent loss of workspace would be worth it or not. The main attraction of Nascom Pascal is that due to its compactness, it can reside in memory in full, still leaving considerable room

for source text and object code, and thus to a high degree ease development and debugging of programs, and, as Mr O'Farrell points out, the omissions can be overcomed by a bit of programming, if the use of these features is essential.

And now for something completely different. On page 7 in INMC80-5, Mr. B.M. Farrelly of Belfast provides a program for retrieving lost Basic programs. An excellent idea, and very useful indeed, but the little piece of code has a couple of errors in it, I'm afraid, since it doesn't restore all of Basic's internal pointers. Only one of them is restored, and, in fact, with a wrong value. The correct code is shown below:

```
21 FE 10 7E 23 B7 20 FB 22 FA 10 5E 23 56 EB 7E
23 B6 2B 20 F6 23 23 22 D6 10 22 D8 10 22 DA 10
DF 5A
```

Like the first version, this code can go anywhere in memory as long as it doesn't overwrite the Basic.

Finally I would like to congratulate everyone on the excellent quality of this magazine. It just grows better and better, and every issue provides me with many hours of interesting reading. Keep up the good work!

Anders Hejlsberg, Poly-Data, Copenhagen.

FREE PROGRAMS, SOUND & SPEECH

You used to give us occasionally a free program or two to try out on our domestic Nascoms, but lately you have been less generous with these gifts. At one time also you were collecting programs of sufficient merit, and selling copies - or have I got it wrong? Anyway, the rival magazine prints free programs! In the latest issue it has a version of the Towers of Hanoi. Now it so happens that I have written a program of the same nature, but of course much better! I am enclosing a listing of the program in case you are interested. (See elsewhere in this issue - Ed.) It makes good use of Nascom graphics, though I have limited the number of disks to five, since we all have finite lifetimes.

I was very interested in the article on the PHG Electronics Programmable Sound Generator. I bought a similar item from Easicomp Ltd. The generator plugs straight into the Nascom motherboard, and has its own built-in speaker. Like the one reviewed, it uses the AY-3-8910 chip. Control is effected through ports 2 and 3, with the Nascom 2 LSW2-8 I/O Select switch in the 'Ext.' position. Documentation is only fair, but it has not proved too difficult to program tunes and sundry noises. I have written a program which converts the Nascom into a keyboard instrument, with a range of three and a half octaves. It is only monophonic, of course, but the tonal quality can be varied by adding various harmonics or altering the sound envelope. I could write up a review of this product if you think it worthwhile. (Please do - Ed.)

Easicomp also advertise a Phonetic Voice Synthesizer at about 150.00. I would be very interested in a review of this - before I splash out that amount of money. The Arfon product sounds alright so long as one sticks to the given limited vocabulary, but to program anything else is, I am told, a fearsome task, and doesn't sound too marvellous when accomplished. Now the Easicomp synthesizer uses 64 phonemes to build an infinite vocabulary, but what does it sound like? That's what I want to know.

I look forward eagerly to each issue of the newsletter and hope it won't be nearly so long before the next issue appears.

John Waddell, Harrow, Middlesex.

N2 VIDEO

REVIEW OF SCREEN WEAVE & SCREEN FLASH ELIMINATORS FOR NASCOM 2 by R.Dowling

SCREEN WEAVE ELIMINATOR

WHAT YOU GET : A small fibre glass board approx. 1.5 x1 inch comes ready built comprising of one i.c.(74LS90), one socket and one 16 pin plug to attach to the Nascom as a piggy-back board.

FITTING : Full instructions are supplied and the unit was found to be simplicity itself in fitting. One i.c. is removed from the main board and housed in the empty socket provided on the small board which is then plugged into the consequent empty socket on the main board. One other connection has to be made to the main board from a terminal pin on the piggy-back board.

UNIT IN USE : On power up it was found that screen weave had disappeared totally and the resultant monitor display was very pleasing. On disconnecting the wire link to the main board, the display was exactly how it was before fitting the unit. However it was discovered at a later date that the unit is not compatible with the well published modification to recover the "lost" bottom two lines of graphics characters which is necessary if fitting a programable graphics generator. The result was screen weave at twice the intensity originally experienced. As the graphics mod. also has the benefit of stabilising the screen as a whole, removal of the screen weave eliminator produced a stable display with only very slight weave noticible if looking closely at the screen.

CONCLUSIONS : A well documented and finished product but almost the same effect can be realised by the above mentioned graphics mod., assuming you don't mind several wire links running along the top of your main board.

SCREEN FLASH ELIMINATOR

WHAT YOU GET : A fully built fibre glass PCB approx 2.5x2 inches consisting of two IC's (socketed), an empty socket, a 14 pin plug and a few discrete components.

FITTING : Instructions supplied with the unit are very easy to follow and consist of removing one IC and fitting this onto the empty socket of the PCB and plugging the PCB into the subsequent empty socket on the motherboard. Two other connections are made from two terminal pins on the PCB to plated through holes near IC7 and IC67. A third terminal pin on the small board enables the user to switch the unit on and off via a switch or port. Most users will not require this facility and therefore will leave the unit permanently working. It is then only necessary to remove the Snowplough (IC58) and put it in the spares draw. The Nascom is then ready to be powered up.

UNIT IN USE : The unit produces wait states but the extra execution time, even for programs that have extensive screen access, is claimed to be less than 1%. It was found that on continuous tabulation to the screen T 0000 FFF8 FFF8 (enter) took 202 seconds on my standard Nascom running at 4MHz with a wait state and with the unit fitted took 203.5 seconds. Tabulating data was found to give a much improved look and on a space invaders program the usual black streaks across large graphic areas had completely disappeared. Running the Nascom at 2Mhz showed that the unit did not work at this speed and, indeed, because the Snowplough IC had been removed, produced large areas of white streaks. The manufacturers claim that replacing one of the IC's on their board from a 74LS74 to a 74S74 would cure all. This has not been tried out.

CONCLUSIONS : The screen flash eliminator does exactly what its name suggests and very neatly too! For what you get it does seem overpriced but the results are certainly worth the expense.

Both units are available from Edac Engineering, 257 Orphanage Road, Erdington, Birmingham or dealers at 8.75 & 14.75 exclusive of VAT respectively.

DOS

Polydos. A review by Stuart Wood

Polydos is designed as a disk operating system for a Nascom with either Nas-Sys 1 or 3 monitors, Nascom ROM BASIC and ideally 48k of RAM. If the full 48k is not fitted then 4k must be fitted at address C000 hex as this is Polydos's workspace area. Two types of disk hardware are supported, namely the Gemini G805 (single density) and Gemini G809/G815 (single or double density) disk drive units. The type of disk unit must be specified when Polydos is ordered since the actual disk driver routines are different. The package as supplied contains two 2708 type EPROMs which contain the bootstrap routines, the package also contains a serialised distribution copy of the operating system itself and one very thick manual! The EPROMs are located at address D000 hex and fitting instructions are given in detail. Don't worry if you have ZEAP in EPROM at this address because you can throw it (or give it) away! Incidently, if you have Naspen or Programmer's aid, you will probably find these also residing in the depths of your dustbin before long. What is the reason for this madness? Very simple really, along with the disk operating system itself is an excellent disk based assembler, an enhanced BASIC and a disk based text editor along with a few other goodies! The package is divided into several major sections which are as follows-

Exec. This is an overlay file which is loaded into the overlay area at C800H to CFFFH. This program handles the command lines which prompt the user with a `$' in the first character position in the line. If the `$' is backspaced over, normal Nas-Sys commands are available with the exception of the `S' and `B' commands. An `E0' command will restore normal operation on these commands. The Exec file contains the code for most of the commands available including: DIR - directory, upto 50 entries with options for statistical data and directory to printer; DEL and UNDEL - allow files to be deleted, and recovered if accidentally deleted; SAVE and LOAD - normally used to manipulate machine code files without execution, an optional load address can be specified; LIST and SKIP - printer control commands for hardcopy of files or list to display; READ and WRITE - direct sector versions of LOAD and SAVE; NEW - tells Polydos you have replaced a disk since directory is held in RAM and must be updated; BOOT - allows complete reboot of system.

Dfun This is another overlay file which handles the following commands: COPY - disk to disk copy and allows single disk copying; REN - rename files; LOCK and UNLOCK - prohibits file deletion or listing in the normal form of the DIR command; ATTRIB - redefine load and execution addresses of files; PACK - removes deleted files and re-packs disk; NAME - give a disk a new name; BUFFER - redefine start and length of RAM buffer used in PACK, COPY and LIST commands.

Emsg This overlay is invoked everytime a system error occurs to print the error message.

Info This file is loaded on boot up (if present) otherwise default parameters are substituted into the information file area at C200H to C2FFH. The file contains parameters relating to the cursor including blink rate and character type and also to the printer including initialisation string and low level driver.

Edit and Ecmd These overlay files control the EDIT command, enabling seperate source and destination files to be given as parameters. The editor supports a wide range of functions and acts as a window on the text. Commands are available for moving/inserting/deleting/finding-changing portions of text in a most powerful way. The files created are suitable for submitting to the assembler or BASIC or as just text files. The Edit overlay may also be called by a user's program for specialised applications. A seperate utility program is available called Polytext for formatting and print time adjustment of letters and documents.

FORMAT The format utility programmme is used for the first time initialisation of blank diskettes. It allows disks to be formatted in single density or double density in the case of a G815/G809 system with automatic verify to check the data can be read back. Diskettes may also just be verified if required.

BACKUP This utility program is used to make direct image copies of diskettes as the name implies, it can also be used on a single drive system allowing the user to place the source disk in the drive which is then read, the user is then prompted to place the destination disk in the drive and the data is then transferred.

SZAP This very powerful program allows you to dive straight into the disk and display the data held there. This is not for the faint hearted since the data can be directly modified on the disk, thus a few careless key presses could result in loss of directory information and render the data useless.

When "BASIC" is typed in the command line mode, a file called "BSdr.BR" is loaded which holds the routines for the Disk BASIC, control is then passed to the ROM BASIC. The BASIC under Polydos is an enhanced version of the Nascom BASIC allowing such features as:

Direct mode commands.

LOAD	Load a BASIC program file from disk
SAVE	Save a BASIC file to the disk in memory image format
SAVET	Save a BASIC file in ASCII text format
EXEC	Load and RUN a BASIC program file
AUTO	Automatic line numbering
REN	Line renumbering
FIND	Find a string

Program/direct mode commands.

SETNEW	Open disk data file, random or sequential access are available
SETINP	Input data from a disk file
SETOUT	Output data to a disk file
SETPOS	Set file position pointer in random access mode
SETCLS	Close a disk file
SETLOAD	Load a file into memory (typically machine code subroutines)
SETCHAIN	Load and RUN a BASIC program and leave variables untouched
SETPRON	Echo output to the printer
SETPROFF	Turn off the above option
SETERR	Trap program errors
SETREAD	Input string variable with editing facilites
SETCLEAR	Define string space and memory size

Polyzap. This is the disk based assembler provided with the Polydos package. It has all the usual assembler features such as pseudo operations which allow conditional assembly, object file definition and the ability to reference previously created symbol files. The ability to use previous symbol files is very powerful and is illustrated by the fact that a file called "SYSEQU.SY" is provided on the master disk which holds all the equated values relating to Polydos workspace, NAS-SYS workspace, subroutine numbers and ASCII character codes. Instead of typing a list of equates in at the start of each program, a statement referencing the file can be used and then a line such as:

```
DEMO    SCAL ZB2HEX ;this routine converts binary to hex
```

may be used which saves lots of tedious typing. There are several other features which are available at assembly time such as force second pass, create symbol file, list symbol table, assembly listing to printer with heading plus a few others.

Summary. The package is well presented and very well documented although there are various spelling mistakes in the manual which one does not expect when so much trouble has gone into the rest of the system. The availability of a decent Nascom oriented disk operating system is a relief since the only other real option is CP/M - a silly choice as you have to throw away your existing software, get 64k of RAM and a Gemini 80x25 IVC card to get the best from it and after that you do not have a Nascom anymore, just another machine which runs CP/M! (Note also that CP/M does not include any form of BASIC or even a Z80 assembler which would together cost in the region of 200.00!). Polydos was purchased from EV Computing in Manchester but is available from your friendly local Microvalue dealer for a fee of 90.00 plus Maggie's bit, which considering the cost of ZEAP, Programmer's aid and Naspen (which are not required) is excellent value for money.

GROUPS

Scottish Nascom User Group

It is proposed to form a local Nascom User Group, probably as a sub-group of the Grampian Amateur Computer Society, depending on prospective members wishes. If you are interested then contact either:

Steve Stubbs, 15 The Glebe, Kemnay. Tel: Kemnay 3070, or
Mike King on Aberdeen 861553

SHROPSHIRE NASCOM GROUP

If anyone is interested in forming a Nascom group in Shropshire please contact me at the address below, giving brief details of equipment and interests. I am interested in the use of my Nascom 2 in Amateur Radio on 70cm.

Has anyone used the CTC chip on the Nascom I/O board successfully? I would welcome some basic help in getting some results in this area.

Don Sunderland, 57 Sutton Lane, Shrewsbury, Salop, SY3 7QQ. 0743 63450

64K EXPAND YOUR RAM A TO 64K

by Alberto di Bene, Italy

I would like to take this opportunity of writing to you, to describe a modification I made on the memory board (the old type, RAM A) of my Nascom 2, to push it to the heights of 64K.

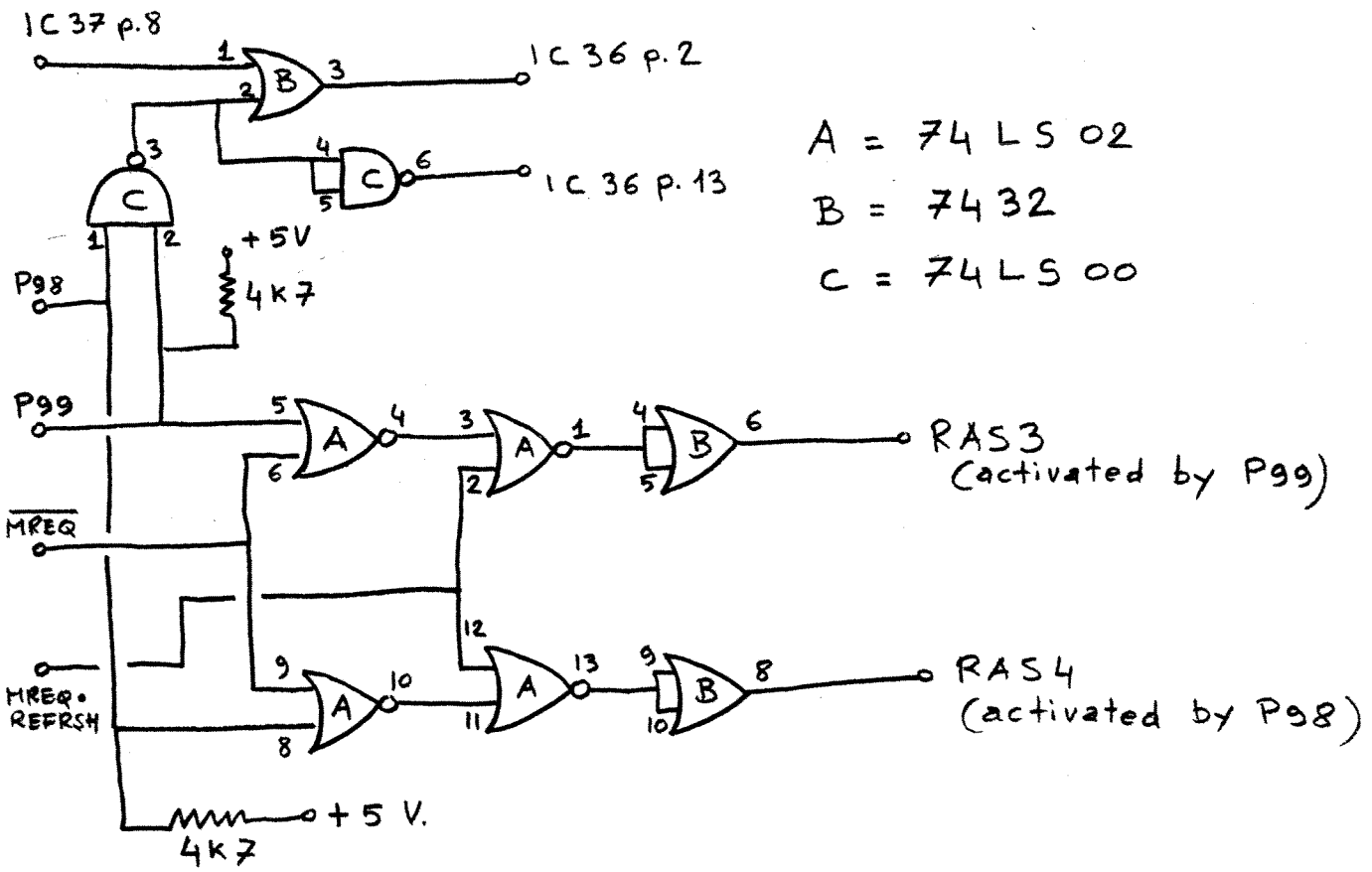
An article about this subject has already seen the light in INMC80-3, but the expansion was limited to 48K, and as I am planning to connect an IBM floppy disk drive 53FD (8" double density, dual side, 1.2 Mbytes!) to the Nascom, I needed a full address space, to be able to run a fully fledged CP/M. The mods I have made to Mr. Fisher's circuit are mainly concerned with the generation of the second additional RAS signal, and with ORing it with the previous one, in order to achieve correct enabling of the output buffer. The IC count has thus increased to three, which is still an acceptable figure.

The step-by-step instructions given by Mr Fisher are still valid, you just have an additional address select pad, let's call it P98, and an additional RAS (RAS4) which must be connected to the fourth row of 4116's piggy-backed on top of ICs 4-11. As already stated in the aforementioned article, you must not connect pin 4 of the top 4116s in parallel with pin 4 of the ICs below. Instead all pin 4s of the fourth row of 4116s must be wired in parallel and then to RAS4.

The circuit has been in operation on my Nascom 2 for a couple of months and has passed all sort of memory tests, op-code fetch included, with flying colours, running at 4 MHz without wait states.

Should someone be concerned with the problem of where to put in the address space of the NAS-SYS monitor, the 8k BASIC and the video-refresh RAM , having already 64K on the expansion board, well, this is a false problem...

In fact, the RAM board covers the full 64K address space, and the various ROMs and the video RAM are simply overlaid on it, thanks to the existence of the RAM-DIS signal on the NASBUS, which give precedence to memory on the CPU card. So when I'm ready for CP/M I'll just have to substitute the N2MD PROM on the CPU card with a new one, suitably programmed, and the now masked RAM will suddenly appear starting from address 0. Simple, isn't it?



The Dr.

Doctor Dark's Diary - Episode the Tenth!

A little thing for Nas-Sys 3 users.

Scenario: a Nascom user (not you, though, because you only use your Nascom for serious purposes, of course) has just typed in the whole 4K of Space Invaders, only to find that it won't run. It must be a typing error. Now, if he had had T2, T4, or even Nas-Sys 1, the whole thing could have been typed in using the L command, and lines with errors in them would have been rejected by the monitor. So, for Nas-Sys 3, here is a fairly minimal program to generate checksums from a program in memory. Comparing these with the listing will allow the easy discovery of typing errors.

```
2A 0E 0C AF 7C 85 06 08 86 23 10 FC F5 E5 EF 20
20 20 20 00 E1 F1 DF 67 EF 0D 00 CF FE 20 28 E3
11 F8 FF 19 DF 66 DF 5B
```

To use it, you would just type the Hex code into an unused bit of RAM, and execute it with the following instruction: Exxxx yyyyNL, where xxxx is the address of its first byte, and yyyy is the address of the first byte of the program you wish to check. The checksum for the first eight bytes will be printed. Press the space bar to produce each subsequent one. Press any other key to exit from the program. When you do this, the address of the start of the line just checked is printed.

I was originally going to put a listing of the program in, but it would have taken up a lot of space, besides which, programs to do this are in all the glossies [Computing Today, anyway] so the Nas-Sys 3 users will all have written their own by now.

Another Nas-Sys 3 fix.

Bits and P.C.'s have kindly told me how to change their BASIC programmers' toolkit so that it will work with Nas-Sys 3. All you have to do is change the byte at B1B5 from 08 to 0C, and they say all will once again be sweetness and light. I have not tried it, but I bet they have! Just goes to show, if you write and ask your friendly dealer a question, they will let you have all sorts of valuable information. I suppose we had best not overdo it, or they will not have enough time left to sell Nascoms in...

I have edited out an incredible [but completely true] story, which was here, about a different Nascom dealer who lied to me on the phone, since it could have been a mistake on their part.

*** INTERLUDE *** [Cue potter's wheel, etc.]

While you watched the soothing picture of a man making a jug on a 78 rpm record player, I got me a Nascom 2 with a disk drive and CP/M. The latter is a lovely thing to have, with some reservations about the standard CP/M issue assembler and editor provided: the assembler is only capable of 8080 code, using weird mnemonics [Intel], and the editor, whilst being powerful in many ways, seems to think a teletype is the only form of input and output in use. By this I mean that you can't move the cursor up the screen to some other line and work there, or step back to the middle of a line and amend it. But...

MONITOR.COM

This is a program that runs on a Nascom 2 under CP/M, and pretends to be an improved version of Nas-Sys. The improvements consist of additional routines to move sections of the memory to disk, and read them back. The details of how to do this have appeared in the second issue of Micropower, a magazine that says quite kind things about this magazine. As quite a lot of typing is needed, I will repeat here the offer I made in the Micropower article. Anyone who wants a copy of MONITOR.COM should send a disk, initialised for their system, plus a pound for postage, packing and not a lot of profit! Frankly, if you have no desire to use all the programs you wrote before you had your disk drive/s, or have a hardware way of switching to Nas-Sys, you don't need this, so there! As an aside, I should mention that the rather primitive disk read and write routines I have written for the program don't work for files longer than 16K. There, saved you the effort of writing to tell me!

For those who have lost INMC80-3, my name and address is: Chris Blackmore, 31 Herne Rise, Ilminster, Somerset, TA19 0HH

DISKPEN.COM

I just converted my EPROM Naspen to use the somewhat altered screen addressing of my system, in a RAM version loaded from disk by MONITOR.COM, and am rather pleased with myself! I wanted Diskpen, but there was no more cash to hand, so that was that, for a while. See later in the episode for details of another editor.

Of course, the files produced by my rather clumsy modifications are not quite CP/M compatible, having only Carriage Returns where there should also be Line Feeds. I get round this slight problem by using another program, whose function is to grab the Naspen file left in memory by the converted Naspen, add line feeds to it and store it as a CP/M file on disk. They can then be edited again with CP/M compatible editors, if necessary, though I keep my working copy on tape, ready to send in for publication.

Meanwhile, here for your interest, if any, are the changes to Naspen needed if the VDU RAM in your system is at F800H:

Address	Change this	To this
B869	11 D0 0B	11 D0 FB
B8A2	11 09 08	11 09 F8
B91C	11 EA 0B	11 EA FB
B94D	21 8A 0B	21 8A FB
B953	11 4A 0B	11 4A FB
BB98	21 80 0A	21 80 FA
BBD3	21 8A 0B	21 8A FB
BBE1	11 8A 0B	11 8A FB
BBF5	21 8B 0B	21 8B FB
BC02	21 90 0B	21 90 FB
BEB6	11 E3 0B	11 E3 FB

One slightly unexpected problem I had during the conversion of Naspen was that the board I was reading it from, a Merseyside Nascom Users Group EPROM board, doesn't seem to send out a RAMDIS signal. Filling the RAM addresses concerned with FF's enabled me to copy the program to an area that wasn't duplicated by more than one board. You may well ask how much RAM there is on the new system... I hardly like to go on about this, but it is a certain manufacturer's 64K page mode job, and what was memory plague, anyway? I should know better than to write that, as I am proposing to reassemble the venerable Marvin soon, so that I can play at Nascom 1/2 system games.

TAB.COM ?

I think that the CP/M routine DUMP.COM has very unsightly output, when used on a Nascom, because it is expecting a screen that is wider than 48 characters. So, if you want a program that tabulates 8 bytes to a line, this is what you do:

1/ Insert a disk with DDT.COM and DUMP.COM on it.
2/ Type DDT DUMP.COM and press Enter.
3/ Type S12E and press Enter.
4/ Amend the byte displayed, which should be 0F, to 07. [i.e. type 07 and press Enter, then type "." and press Enter.]
5/ Type G0 and press Enter.
6/ Type SAVE 3 TAB.COM and press Enter for the last time.

You now have a new command, which will impress all sorts of people, or [more likely] make them wonder what on earth you are going on about, which is what happens when I try to explain about computers to the people I work with. Console yourself with the thought that they are not on the leading edge of the highest technology around, and probably can't even remember where they left their towels.....

NIFTY WAY TO PROGRAMMABLE GRAPHICS

Of all the equally effective ways of producing the character set of your dreams on your Nascom, I chose to use a Nasbus compatible board made by a firm called Aids To Industry. This board can be fitted with from 1K to 4K of RAM, which the system accesses through the bus, or through a ribbon cable with a 24 pin header on the end. This latter plugs in where the graphics EPROM used to be. Normally, the board is supplied in a form that requires you to set a switch to decide which side the board is read from, but if you ask them nicely, ATI will modify the board slightly, so that it gives priority to the Nasbus, but is connected to the character generator socket at all other times. Handier than pushing a switch? Lots.

SAD EVENT (NO. 12345?)

Microdigital, who used to be a Nascom dealer once upon a 1 MHz clock, have sold their magazine, the Liverpool Software Gazette. It has been taken over by the Apple users group, and renamed "Windfall". I only mention this in passing because under the new management, and printed on very posh type glossy paper, it looks exactly the way this magazine would, if it had more advertisers. They have even thrown out the machine that they were using to put all those mispnirts in it....

If anybody has one they don't want, I would like to haggle over a Microdigital Relay board, on account of it seems like the easiest way of automating things like my tape machine.

WHATEVER HAPPENED TO.....? SECTION

Whatever happened to the following more or less interesting (according to your point of view) things?

1/ The letters from all the Citizens Band experts, telling us how we can set up a Nascom network, or that it is impossible. Can it be that they are too busy discussing the "Smokey Bacons" to spare the time? Networking is too important and interesting to remain the preserve of TRS80 owners and the millionaires who have British Telecom modems..... [In the event that they just don't answer, how about an article from one of the legitimate radio hams who use radio teletypes?] (See Dave Hunt's bit elsewhere - Ed.)
2/ The answers to the last "whatever happened to...."?
3/ The list of things I was going to ask about?

PILOT FEEDBACK

Quite a while ago, I had a letter from R.B.Poate, who said, "I would suggest that you use Naspen or something similar for your line numberless editor." This is in fact, very near to what I actually intended to do. My plan was to say something along the lines of "Right, here is the specification for each of the parts of the Pilot interpreter, now let's each write a bit..." Nasty, isn't it? The specification for the editor only has to state the way the program is stored, and where in memory it actually is, and the task of adding your (or somebody else's) editor is left to you. Do I hear howls of "Cop out!"? Probably, but you see I don't want to write two or more versions of the same thing, and my one is to run under CP/M, not Nas-Sys. That is why all the monitor calls in the code so far printed go to vectors.

WITHOUT FURTHER ADO...

The editor will be any code you care to write that enables you to manipulate the stored program to your own satisfaction. It fits into the free code from the earlier episode where lines 1490 to 1500 currently are. Yes, Heloise, I do know I forgot to resequence that bit. The program storage area starts at location RAM, and the first two bytes hold the address of the last byte of the program [low byte first, of course.] Having tried more than once to write a decent editor program, I am copping out yet again; I shall be using an editor written by the genius at Hi-Soft. See next section... In order that the programs we all produce will be compatible, we will all separate lines with the combination of characters, CR LF [in other words, 0D 0A]. This will cause howls of apathy, or possibly anger I shouldn't wonder, from those of you without CP/M and disks, who would have preferred to use 0D on its own. Why aren't you writing this?

The Load module, which replaces lines 1450 to 1460 of the listing, reads from tape, disk, Tibetan prayer wheel, or other mass storage unit into the memory, at location RAM. Users of any Nascom monitor, it is your turn to be smug, because the monitor will do this for you with very little extra code of your own.

The Save module, lines 1470 to 1480, it will hardly surprise you to learn, transfers the program to the storage medium from the memory. Now do you see why RAM and RAM+1 hold the address of the end of the program? [If you don't, why are you reading hard core computer magazines?]

A SOFTWARE REVIEW.

This review concerns the "Z80 development package" currently being sold by Hisoft. The package consists of an editor, an assembler, a debugger [*** semantics error ***] and another program whose use I have not yet discovered. The package sells for 50.00, a price which compares extremely favourably with the amount you would need to lash out to buy Diskpen and Macro 80, [about 150.00, I think.] The assembler works very well, also fast. It can handle source files far larger than your computer's memory, as can the editor. The editor is a joy to use (I know!! - Ed.), especially after you have become used to the horrors of ED.COM. An especially nice feature is the way it copes with very long lines, by scrolling the whole screen across, instead of wrapping round to the next line, or crashing, or any of the weird things some editors do when you ask more of them than their authors intended. The debugger takes a while to get used to, but is very useful, I have found it very good indeed at finding the silly errors in my early attempts at converting the skeletal Pilot to a CP/M form. If this package does not seriously dent sales of both Diskpen and the grossly overpriced Macro 80, I shall be very surprised indeed. So that you will not think I am being paid by Hi-Soft to say these nice things about their product, I will just mention in passing that I was very annoyed to have to wait over a fortnight for delivery. It was worth waiting for, but why was it necessary to wait?

end end end end end end end end end end end

RTC Board

NASCOM REAL TIME CLOCK AND CALENDER by J. R. Williams

National Semiconductor MM58174

The National Semiconductor MM58174 is a CMOS real-time clock and calender chip in a 16-pin DIL package which is easily interfaced to the Nasbus. This device incorporates 4-bit registers from which the CPU can read, as BCD digits, the time from 1/10ths secs. to tens of hours, the day of the week, and the date from days to tens of months. The on-chip oscillator is controlled by a 32768 Hz crystal and timekeeping can be maintained by a backup battery when the 5V supply is switched off. In addition, the MM58174 provides a facility for generating CPU interrupts at intervals of 0.5 sec, 5 sec or 60 sec.

Interfacing to Nascom

The circuit used by the writer to interface the MM58174 to his NASCOM 1 is shown below. The circuit was assembled on a Vero DIL prototyping board which plugged into a spare 80-way connector on the NASBUS. Bus timing was not a problem with the NASCOM 1 which runs with a 2 Mc/s clock and should still be acceptable with the 4 Mc/s clock of the NASCOM 2 provided that the 500ns MM58174N is used.

Addressing

The clock chip is interfaced onto the bus occupying 16 I/O port addresses (20H to 2FH in the writer's system). The bus address lines A0 to A3 connect to the clock register address input pins AD0 to AD3. The upper I/O address lines A4 to A7 are decoded by a 74LS145 to select addresses in the range 20H to 2FH. The clock chip is selected when an address in this range coincides with $\overline{\text{IORQ}}$.

Data input/output

The clock's data I/O pins DB0 to DB3 connect to the NASBUS data I/O lines D0 to D3. The NASBUS $\overline{\text{RD}}$ and $\overline{\text{WR}}$ lines connect directly to the MM58174's NRDS and NWDS pins. The NASBUS data bus is switched to the read direction by pulling $\overline{\text{DBDR}}$ down when the clock chip is read by the CPU.

Interrupt Control

Additional logic is included to enable the MM5174's interrupt facility to be used. When the clock's interrupt output goes low, an interrupt is initiated provided IEI is high - i.e. no higher priority interrupt is active. While either the clock interrupt or a higher priority interrupt is active, IEO is held low to inhibit any lower priority interrupts.

Operation in the Z80 interrupt mode 2 is provided for by pulling all eight data lines down to return a zero interrupt vector (i.e. 00H) in response to the CPU's interrupt acknowledge ($\overline{\text{IORQ}}$ active with $\overline{\text{M1}}$). Simultaneously, $\overline{\text{DBDR}}$ is pulled down to switch the data bus to the read direction.

All this logic can, of course, be omitted if the interrupt facility is not needed.

Power supply

When the NASCOM 5V supply is available, the clock chip runs with Vdd=5V via TR1. It is important that this arrangement, rather than a simple diode, be used to ensure that Vdd is close enough to the 5V supply to ensure that an input which is pulled up to 5V does not exceed the MM58174's maximum input voltage rating of Vdd+0.3V.

Standby power (10uA approx.) is provided by a pair of AA-size NiCad cells. These are trickle charged via R1 when the 5V is available.

Clock chip operation

For full information on how the MM58174 is controlled, see the NS data sheet (Feb. 80 or later issue). However, it may be useful to note the following points:

1. Data-change flip-flop operation.

Whenever the data registers are updated - i.e. at 0.1 sec intervals when the clock is running - a data change flip-flop is set. This causes the data value "1111" (i.e. "-F") to be output at the next attempt by the CPU to read a data register. This informs the CPU that a data change has occured and also resets the flip-flop so that the new data can be read by subsequent CPU input instructions.

2. Clock counter errors.

When testing the MM58174, the writer encountered a problem of counting errors in the clock's 0.1 sec. counter and, less frequently, in the seconds counter. These caused the clock to gain up to 1 min. per hour when under test although it kept good time when running on the NiCad battery with the computer switched off. Investigation eventually revealed that this problem was related to the frequency with which the CPU read the data registers.

The test program being used when the above problem was found read the 0.1 sec. register repeatedly while testing for an "-F" data value indicating a data change. This meant that the register was being read perhaps 30,000 times per sec. With a modified version of the test program which tested for the "-F" condition at intervals of 50ms, clock counter errors were no longer detected.

3. Interrupt operation.

Interrupt operations are controlled via the interrupt register. First, the interrupt system is must be initialised by outputting zero to this register, followed by three input instructions to condition the interrupt logic. The required interrupt timing period will then start when an appropriate non-zero value is output to this register. For example, the data value "1001" (i.e. 9) will initiate a repeated 0.5 sec interrupt.

In order to reset the clock's interrupt output, the interrupt servicing program must read the interrupt register. A second input from this register is required to condition the interrupt logic and a third to restart the interrupt timer if in the repeated interrupt mode.

An irritating feature of the interrupt system is that, even with immediate servicing of a repeating 0.5 sec interrupt, the interrupts actually occur at intervals of 516.6 ms.

A further point to note is that the clock's interrupt system is not reset on a computer reset. This will cause problems if an attempt is made to run another interrupt-using program which does not expect clock interrupts, unless the clock's interrupt system is first reset by software. This can be done by outputting zero to the interrupt register and then reading the register.

Demonstration program

The listing of a demonstration program TIMEDATE 1 is given below. This is written for use with NAS-SYS and is entered at 1000H. The program then requests that the starting time (hh mm) and date (dd mm yy) be keyed in - e.g.:

```
23 59 31 1 82
```

The clock will be started from this time when ENTER is hit. If the line entered starts with a space (e.g. if no time is provided), the existing clock time will be retained.

TIMEDATE 1 then returns to NAS-SYS, from which a user program can be entered if required. However, NAS-SYS or the user program will be interrupted at intervals of 516.6 ms while the interrupt routine updates a time and date display on the top line of the screen.

Note that a separate stack is used by the interrupt program, because the interrupted program (such as NAS-SYS) may not have enough spare stack space for the interrupt routine. The interrupt pointer address defined by the contents of the I register and the vector (00H) returned by the clock is loaded with the start address of the interrupt routine.

Components list for real-time clock and calender

Resistors:
R1=220R
R2=4.7K
R3,R4,R5=10K
R6=1K
R7,R8=10K

Capacitors:
C1,C2=10nF
C3=10uF 16V
C4=6-36pF

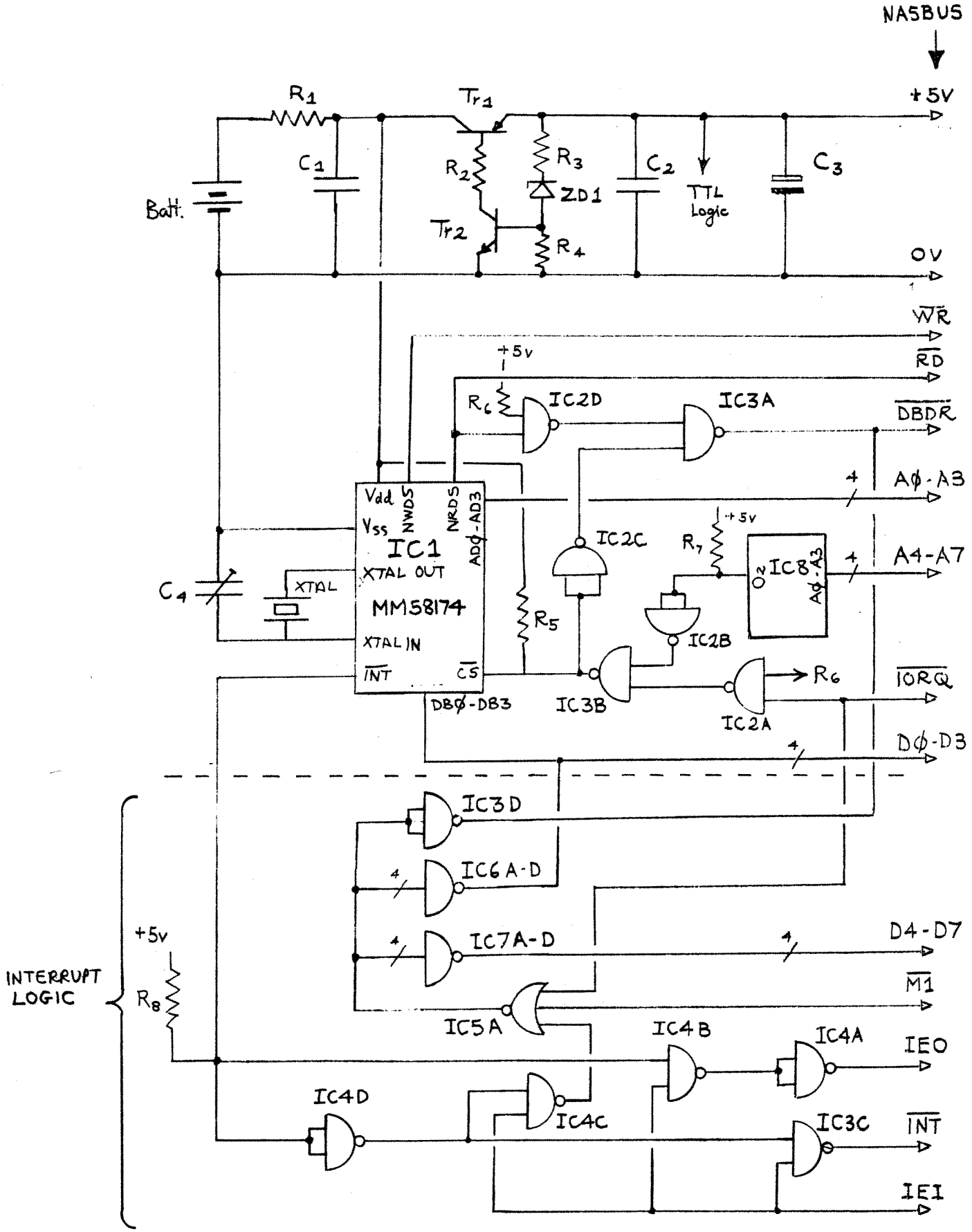
Integrated circuits:
IC1=MM58174N
IC2=74LS00
IC3=74LS01
IC4=74LS00
IC5=74LS27
IC6,IC7=74LS05
IC8=74LS145

Transistors and diodes:
Tr1=BC178
Tr2=BC108
ZD1=BZX83C 3V3

Miscellaneous:
XTAL= 32.768 KHz crystal
Batt.= 2 AA NiCad cells in series

REAL-TIME CLOCK AND CALENDER

by J. R. Williams



ZEAP Z80 Assembler - Source Listing

```
                    0010 ;TIMEDATE 1 21-2-82  by J. R. Williams

                    0030 ;Demonstration program for MM58174
                    0040 ;real-time clock - assumes NAS-SYS.

                    0060 ;Constants:
2F0A 0020           0070 CLOCK  EQU  20H
2F0A 0BE0           0080 CRT    EQU  0BE0H
2F0A 0011           0090 VECTH  EQU  11H
2F0A 000D           0100 CR     EQU  0DH
2F0A 0020           0110 SPACE  EQU  20H
2F0A 0028           0120 PRS    EQU  28H
2F0A 005B           0130 MRET   EQU  5BH
2F0A 0063           0140 INLIN  EQU  63H
2F0A 0064           0150 NUM    EQU  64H
2F0A 0C21           0160 NUMV   EQU  0C21H

1000                0180        ORG  1000H
1000                0190        ENT

                    0210 ;Set CPU interrupt mode
1000 3E11           0220        LD   A VECTH
1002 ED47           0230        LD   I A
1004 ED5E           0240        IM   2

                    0260 ;Demand time and date
1006 EF             0270        RST  PRS
1007 0D             0280        DEFB CR
1008 456E7465       0290        DEFM /Enter time (hh mm)/
     72207469
     6D652028
     6868206D
     6D29
101A 20616E64       0300        DEFM / and date (dd mm yy)/
     20646174
     65202864
     64206D6D
     20797929
102E 0D00           0310        DEFB CR,0
1030 DF63           0320        SCAL INLIN

                    0340 ;Test for valid input
1032 1A             0350        LD   A (DE)
1033 FE20           0360        CP   SPACE
1035 2839           0370        JR   Z CINT

                    0390 ;Initialise clock
1037 0E20           0400        LD   C CLOCK;Test register address
1039 AF             0410        XOR  A
103A ED79           0420        OUT  (C) A

                    0440 ;Stop clock
```

```
103C 0E2E      0450          LD    C CLOCK+0EH;Stop/start register
103E ED79      0460          OUT   (C) A

               0480 ;Enter time (hh mm)
1040 0E26      0490          LD    C CLOCK+6;Units of hours
1042 D73E      0500          RCAL  LOAD
1044 0E24      0510          LD    C CLOCK+4;Units of minutes
1046 D73A      0520          RCAL  LOAD

               0540 ;Enter date (dd mm)
1048 0E28      0550          LD    C CLOCK+8;Units of days
104A D736      0560          RCAL  LOAD
104C 0E2B      0570          LD    C CLOCK+0BH;Units of months
104E D732      0580          RCAL  LOAD

               0600 ;Store year (yy)
1050 DF64      0610          SCAL  NUM
1052 3A210C    0620          LD    A (NUMV)
1055 32E510    0630          LD    (YEAR) A

               0650 ;Calculate years since leap year
1058 D604      0660 DIV      SUB   4
105A 27        0670          DAA
105B 30FB      0680          JR    NC DIV
105D C608      0690          ADD   A 8
105F 27        0700          DAA
1060 47        0710          LD    B A ;answer (4 to 7) in B

               0730 ;Load leap year register-
               0740 ;Bit pattern for leap year (8) is loaded
               0750 ;to both halves of A .This is rotated B
               0760 ;times to give bit pattern for this year.
1061 3E88      0770          LD    A 88H
1063 0F        0780 MOVE     RRCA
1064 10FD      0790          DJNZ  MOVE
1066 0E2D      0800          LD    C CLOCK+0DH;Years status register
1068 ED79      0810          OUT   (C) A

               0830 ;Start clock
106A 0E2E      0840          LD    C CLOCK+0EH;Start/stop register
106C 3EFF      0850          LD    A 0FFH
106E ED79      0860          OUT   (C) A

               0880 ;Initialise clock interrupt
1070 0E2F      0890 CINT     LD    C CLOCK+0FH; Interrupt register
1072 AF        0900          XOR   A
1073 ED79      0910          OUT   (C) A
1075 ED78      0920          IN    A (C)
1077 ED78      0930          IN    A (C)
1079 ED78      0940          IN    A (C)
107B 3E09      0950          LD    A 9;Repeated 0.5 s interrupts
107D ED79      0960          OUT   (C) A
107F FB        0970          EI
```

```
                    0990 ;Return to NAS-SYS 3
1080 DF5B           1000         SCAL MRET

                    1020 ;Subroutine LOAD
1082 DF64           1030 LOAD    SCAL NUM
1084 3A210C         1040         LD   A (NUMV)
1087 ED79           1050         OUT  (C) A
1089 0C             1060         INC  C
108A 0F             1070         RRCA
108B 0F             1080         RRCA
108C 0F             1090         RRCA
108D 0F             1100         RRCA
108E ED79           1110         OUT  (C) A
1090 C9             1120         RET


                    1150 ;INTERRUPT ROUTINE

                    1170 ;Save stack pointer and registers
1091 ED73E610       1180 INTR    LD   (SPSTR) SP
1095 31FC10         1190         LD   SP STACK+20
1098 F5             1200         PUSH AF
1099 C5             1210         PUSH BC
109A E5             1220         PUSH HL
                    1230 ;Reset interrupt and restart timer
109B 0E2F           1240         LD   C CLOCK+0FH;Interrupt register
109D ED78           1250         IN   A (C)
109F ED78           1260         IN   A (C)
10A1 ED78           1270         IN   A (C)

                    1290 ;Display hours, minutes and seconds
10A3 21E00B         1300         LD   HL CRT
10A6 0E27           1310         LD   C CLOCK+7;Tens of hours register
10A8 D727           1320         RCAL DISP
10AA D725           1330         RCAL DISP
10AC D723           1340         RCAL DISP
10AE 3620           1350         LD   (HL) SPACE
10B0 23             1360         INC  HL

                    1380 ;Display dd and mm
10B1 0E29           1390         LD   C CLOCK+9;Tens of days register
10B3 D71C           1400         RCAL DISP
10B5 0E2C           1410         LD   C CLOCK+0CH; Tens of months
10B7 D718           1420         RCAL DISP

                    1440 ;Display yy
10B9 3AE510         1450         LD   A (YEAR)
10BC 0F             1460         RRCA
10BD 0F             1470         RRCA
10BE 0F             1480         RRCA
10BF 0F             1490         RRCA
10C0 D71C           1500         RCAL ASCI
10C2 3AE510         1510         LD   A (YEAR)
10C5 D717           1520         RCAL ASCI
```

```
                1540 ;Restore registers and stack pointer
10C7 E1         1550        POP  HL
10C8 C1         1560        POP  BC
10C9 F1         1570        POP  AF
10CA ED7BE610   1580        LD   SP (SPSTR)

                1600 ;Enable interrupts and return
10CE FB         1610        EI
10CF ED4D       1620        RETI

                1640 ;Subroutine DISP
10D1 0602       1650 DISP   LD   B 2
10D3 ED78       1660 DISP2  IN   A (C)
10D5 D707       1670        RCAL ASCI
10D7 0D         1680        DEC  C
10D8 10F9       1690        DJNZ DISP2
10DA 3620       1700        LD   (HL) SPACE
10DC 23         1710        INC  HL
10DD C9         1720        RET

                1740 ;Subroutine ASCI
10DE E60F       1750 ASCI   AND  0FH
10E0 C630       1760        ADD  A 30H
10E2 77         1770        LD   (HL) A
10E3 23         1780        INC  HL
10E4 C9         1790        RET

                1810 ;Data storage
10E5 82         1820 YEAR   DEFB 82H ;Default year=82
10E6 0002       1830 SPSTR  DEFS 2   ;User stack pointer
10E8 0014       1840 STACK  DEFS 20  ;Interrupt routine stack
1100            1850        ORG  1100H
1100 9110       1860 POINT  DEFW INTR;Interrupt pointer
```

```
ZEAP Z80 Assembler - Symbol Table

10DEH 1750 ASCI          1070H 0890 CINT
0020H 0070 CLOCK         000DH 0100 CR
0BE0H 0080 CRT           10D1H 1650 DISP
10D3H 1660 DISP2         1058H 0660 DIV
0063H 0140 INLIN         1091H 1180 INTR
1082H 1030 LOAD          1063H 0780 MOVE
005BH 0130 MRET          0064H 0150 NUM
0C21H 0160 NUMV          1100H 1860 POINT
0028H 0120 PRS           0020H 0110 SPACE
10E6H 1830 SPSTR         10E8H 1840 STACK
0011H 0090 VECTH         10E5H 1820 YEAR
```

POINTS ARISING

BY RORY O'FARRELL

INMC80-5, page 20. In my review of the Econographics for the N1, where I say "is not acceptable", please substitute "is not unacceptable". The Econographics unit gives a very slight line between solid blocks of white, but it is not objectional. I am also informed by Nascom that the faulty piece of artwork for the track patterns has been withdrawn, and profuse apologies have been expressed for any inconvenience. With any luck, any Econographics units around will now be from the correct track pattern.

INMC80-5, page 58. For microprocessor, read macroprocessor. This is obviously a computer error (GROANS and slow handclap!)

Uncle Dusty's Notes for CP/M users

Dusty Pulver

When you're in the command mode of your CP/M'ed Nascom or Gemini (i.e. when it prompts A>), typing in the name of a .COM file will cause that file to be loaded into the computer at location 100H upwards, and then the instruction at 100H will be executed. How nice it would be if you could load a program without running it, or run a previously loaded program. Well, you can. This is how.

LOADING A PROGRAM

To load a program without running it, type in the program name followed by a space and ^A. CP/M will load the program, then the CCP will look at what follows the space to see what has to be put into the default file control block (FCB). Finding ^A which is an `illegal' character, it will output the symbol that represents ^A (a sort of upside down L in a Nascom) followed by a `?', meaning `I don't understand', and will then return to the command mode, leaving the program that you wanted loaded into memory, but unexecuted, and unaltered.

EXECUTING A PREVIOUSLY LOADED PROGRAM

First of all create a new program called RUN.COM. You don't need an assembler or anything like, just type in the following:

```
SAVE 0 A:RUN.COM
```

This will put the most wholly remarkable program in the entire universe onto the disk in drive A. It is 0 (yes, zero) bytes long, and occupies no room on the disk, apart from one of the available directory spaces. Whenever I format a new disk, this is the second program I put onto it (the first is DISKNAME.007, or whatever I've decided to call that disk. You can't beat a good disk catalogue!).

When in the command mode, typing RUN will execute the program at present in memory, whether you've just put it there as previously described, or whether it's something you've already run, but want to run again, but without the time lag caused by the program being reloaded from disk.

More amazing still, RUN.COM will accept entries for the TFCB, so if you've got, for instance, PIP.COM loaded, typing RUN B:=A:PROGRAM.COM[V] will copy the file PROGRAM.COM from disk A to disk B and verify it, but without the wait while PIP.COM is loaded from disk.

Apart from the time advantage when running the same program several times, but with different parameters, the main use of this program is that you can load a program, change the disks, do a ^C to `log in' the new disks, and then run the program, using data on, and writing to the new disks. For instance using PIP.COM you can copy the whole of one disk to the other, even though PIP.COM is on neither of them. Also you can load a program and then save it onto another disk, without all the hassle of going through DDT or ZSID even in a single drive set-up. Just use STAT.COM to find the number of sectors a program uses, and then load the program as described. Change disks, use ^C to log in the new one, and save the program onto the new disk.

Note... RUN.COM will not work repeatedly with some programs, such as the CP/M user group's FMAP, because they change themselves when they are run.

USING RUN.COM with MBASIC-80

Microsoft's disk based MBASIC-80 alters itself on initialization so that if you exit the program with the command `SYSTEM' or if the program crashes due to a disk error, or you not using `RESET' when you change disks, you can perform a warm start by typing in RUN thereby retaining your program in memory, that a cold start would erase!

FREE MEMORY!

That is, free as in make available. Oh well, with the price of 4116's as low as they are now, I'm sure you can afford to buy them (and just be thankful that you haven't got a ZX81... nearly fifty quid for 8K of RAM!). But I digress.

How much storage space have you got on your disks? According to my calculations, excluding the three system tracks, and the directory, a double sided single density, 5.25" Pertec drive in a Henelec/Gemini G805 system will put 148K onto each disk. My system will only put on 143K. Fiddle! Where's my other 5K? I can access it using D-DOS and my Zap program, so why not with CP/M? A few moments with ZSID gave the answer. The byte in the BDOS which tells the system how much memory the disk will hold was set to 90H, that is 144. Increasing it to 95H gave me back all of my disk. My system was one of the very first Gemini CP/M systems sold, so maybe the fault has been corrected in later releases of the software. None the less, it may be worth your while checking your system. Just use STAT.COM on an empty disk, and see whether is states 143K or 148K free. (It was done deliberately to limit the track access on side 1 of the Henelec system as some read errors were occurring, you work out why!! Ed.) If you want to change it, the disk size byte can be found at the following locations in the Henelec/Gemini CP/M 1.4:

16K System	313EH
46K System	A93EH
CPMxx.COM	11BEH
MOVCPM.COM	123EH

Obviously it's better to change it in MOVCPM.COM, so that all future systems will be correct, but use a back-up copy!

RESET SWITCH

On my Nascom system, I've removed the reset switch from the keyboard, and put it on the front panel of the computer itself. In fact, I've not put on just one switch but two wired as in the diagram below. The idea is that switch 2 resets to 0000H and switch 1 resets to F000H. Switch 2 therefore cold boots CP/M and switch 1 resets NAS-SYS when that's loaded, or gives a WBOOT jump in CP/M, at least, that's the idea, so why doesn't the Warm Boot work? Can anybody help?

- No reply. Watch my lips.... C.a.n. .a.n.y.b.o.d.y. .h.e.l.p.?.

DR.DARK'S DISK BASED NAS-SYS

In the second issue of Micropower (which isn't as bad as I feared it might be), Dr. Dark described a method of adapting NAS-SYS 1 to run in a CP/M environment which is vastly superior to that used by CC Soft in their Disk Constructor Package (excepting that that is NAS-SYS 3). Using his method, it's possible to adapt NAS-SYS 3 using the same instructions, except that the video address references are in a different position, as is the jump table. He has used (P)ut, (F)etch, (D) to exit to CP/M, and (Y) for a copyright notice. Altogether a very nice little patch, for the best debugger around.

The trouble is that I've already used F for Find, and NAS-SYS 3 uses P to display the stored user program registers. So, what do I use? I use D to jump into the disk sub-system, from where I have a choice of (R)ead, (W)rite, (N)as-sys, and ^C to exit to CP/M. An error takes me back to the disk sub-system, but a successful read or write returns me to NAS-SYS. The next job is to eliminate the code in NAS-SYS that's duplicated in SYS. The changes from Chris Blackmores' instructions for NAS-SYS 3 are:

Address	Old Code	New Code
0800	31 00 10	00 00 00
019B	21 0A 08	21 0A F8
0236	11 0A 08	11 0A F8
023E	11 BA 0B	11 BA FB
0254	11 CA 0B	11 CA FB
0265	11 0A 08	11 0A F8
0268	21 4A 08	21 8A FB

The jump table addresses are:

D	0788
F	078C
P	07A0
Y	07B2

and if you want any others, calculate it from these, remembering that they're in alphabetical order, with 2 bytes per letter.

All in all, it's nice to be able to run NAS-SYS programs without having to multimap the computer, and it's wonderful being able to load programs to and from disk, to anywhere in memory, with CP/M looking after all the disk functions. It's far easier than writing your own DOS as some poor fool has done in Computing Today, ignoring the fact that he already has all of the necessary code, bar the 180 bytes that Chris has written.

Incidentally, the Find routine that I use is based on that in the same issue of Micropower, but has been modified

a) to make it work

b) it saves ARG1 to ARG10

then asks you what you want to find, and restores the ARGS when all done, rather than corrupting the values that are used to format the T command.

HERE WE GO AGAIN

Wonderful news for those of you who have just upgraded from CP/M 1.4 to CP/M 2.2. The father of CP/M, Gary Kildall, has let it be known that CP/M version 3 will be with us just as soon as it's debugged, probably during '82. Let's hope that they get it right this time, so it doesn't have to go through several versions before it's OK, as happened with version 2.

Beyond the features of version 2, version 3 will add passwords, file lockout, and record lockout, thereby increasing the security of your programs when others have access to them, plus features to make life easier for us hackers, such as the ability to type ahead of the system, which will catch up with you when it has time (the Gemini IVC and keyboard allow this already - Ed.), the ability to test and write a record, support of a real time clock/calendar, an updated ED.COM which is screen oriented (hooray!), much better documentation (hooray hoorah!), and of course a smaller TPA (booo!).

In readiness for this I'm building a couple of real time clocks, the first using National Semiconductor's 58174 chip with battery backup, and the second based on a receiver for the NPL's radio clock at Rugby. I'll try to write up a report on which is best, and why, for the mag. when I've finished (probably with circuit diagrams and layouts for those of you who want to have a go yourselves).

MAKE D.R. UTILITIES FIT YOUR SCREEN

The utility programs supplied with CP/M are all written for an 80 x 24 character video screen, not the little 48 x 15 Nascom screen that we all know and love. (It's interesting, the Gemini double density implementation of CP/M 2.2, in the version for the Nascom 48 column screen, is 48 x 16, seems no-one's noticed! - Ed.) The result is that the information we want either wraps round onto the next line, or scrolls off the top of the screen. Short of spending a hundred and fifty quid on either the new Nascom Advanced Video

Controller card, or the Gemini Intelligent Video Card, to give us the required screen size, there's no way that we can see what we want to see nicely formatted on the screen. Therefore I was pleased when I found an article by J. Baraclough on page 135 of January '82's PCW giving the necessary changes to ED.COM and DDT.COM for them to fit onto smaller screens. To summarise his article: In ED, the number of lines written by the P command are held in location 161CH (Version 1.4), and 17DBH (Version 2.2). Change this from 17H to 0DH. For DDT's D command, the number of bytes displayed per line is held at 0A15H, change this from 0FH to 07H. The number of bytes in each block displayed is held at 09EDH, so change this from 0BFH (16 x 12 lines) to 05F (8 x 12) or 06F (8 x 14 lines). The number of lines disassembled by the L command is held at 09BCH, it's set to 0CH (12 lines) change it if you wish to 0FH for your 15 line screen.

Spurred on by this information, I sat down and worked on ZSID.COM, and came up with the following: For the D command, 12AFH holds the number of bytes displayed per line, change this from 0FH to 07H for a screen less than 54 characters wide. 125EH holds the number of bytes displayed per block, so change this from 5FH to 2FH for a standard Nascom screen.

AN OPEN LETTER TO RICHARD BEAL

Oh well, Mr. Beal, it looks as though you're going to have to start work now if you're going to get a CBIOS written to support CP/M version 3's new features by the time it's released. When you've finished it, how about writing a utility program to run under CP/M that will give us all the debugging features of NAS-SYS that we know and love, plus a few extras such as Find, Compare, Read and Write programs from/to disk, and Call (like Execute, but CD rather than C3). It doesn't need to have any screen handling or support for the Nascom keyboard, because that's already in the CBIOS, in fact it can lose all the Monitor functions, and there will be no need for us to be able to access any of the subroutines unless we want to write `illegal' programs.

It shouldn't be difficult, or take long, because you've already got most of the source code. All it needs is the ability to relocate itself to the top of the TPA. I'd do it myself, but I'd feel tempted to let other people have a copy, and that would get me into copyright problems, wouldn't it!

MACRO-80, LINK-80, ZSID, AND SYMBOLS

Those of you who are rich enough to own the Microsoft MACRO-80 macro assembler and Digital Research's ZSID Symbolic Instruction Debugger (I once knew a singing duo called the Symbolics. Sym was a really nice girl, but her partner....) or those who are dishonest and know someone who has these packages will know that the symbol table produced by MACRO-80 is incompatible with ZSID.

However, the table of globals produced by LINK-80 is compatible. So, we have to convert all the symbols in the source file for MACRO-80 to globals. This is easily done using ED.COM or DISKPEN. Using the F command replace all the occurrences of `:^I' with `::^I'. This will look for the colons before a tab, and replace them with two colons, thus declaring that symbol to be public. Using LINK-80, the Switch `/Y' when used after `/N' and before `/E' or `/G' will create a symbol table `PROGNAME.SYM' which is compatible with ZSID.

ILLOGICAL, CAPTAIN

I find that the Nascom 2 keyboard is illogical. Every time I try to hit the `ENTER' key, I usually hit one of the other keys that it hides amongst (OK, I'm not the best typist around, all those spelling mistakes in this article are really tpynig errros!). I usually hit the one that sticks out on the end: CH/LF. So, one night in a fit of drunkeness, I attacked my keyboard with a blowtorch and some solder, and when I next used my computing machine, I

had rather a nice surprise, because I'd swapped over the wiring to the two keys, so now the 'ENTER' key was by itself on the end. The only trouble was that I'd lost the key cap for it, and jabbing at the thin stalk of plastic was even more difficult than hitting the key in its original place. I searched everywhere, but no sign of the button (I always seem to lose things when I'm drunk. I once lost my false teeth, I think I must have flushed them down the loo after I'd been sick.) (Keep it clean please - Ed.) So a trip was made to my local friendly electronic parts shop, who had only red double width caps. Bliss! I now have a big distinctive key that I can't miss even if I try. See diagram 2 for details of how to change the printed wiring on the back of the keyboard. (If you're using a G805 with SYS, why not change the keyboard table? - Ed.)

SKEWING THE SKEW

I have the use of another CP/M machine besides my Nascom. This is an AVL Eagle, and also uses 18 sectors per track, but with a different skew. Problem? How do I transfer programs from one computer to the other.

First Solution.

Ensure that the system on all disks is of the same size, with source disks containing a CP/M system from the computer that's being copied from, and the receiving disks containing the other system. Log in the source disk, and load the program without running it, as described above. Put in the receiving disk, and do a ^C to change the BDOS in memory... this doesn't alter the BIOS, so now save the program onto the new disk, with the new format. Simple isn't it (provided that you remembered how long the program was, and didn't have to go back and use STAT, and then start again!). And it works, no matter how many sectors there are per track, the only requirement being that both types of disk should work in the same drive.

Second solution.

Write a special program to do the job using direct calls to the BDOS primitive functions, or simpler, adapt somebody elses' program. Practical Computing published a fast disk copy program by T. D. Lee in their Sept '81 issue, which uses a skew table to speed things up. By having separate tables for reading and writing, the skew can be changed on an entire disk very quickly indeed. For your information, the skew table can be found at 311AH in a 16K system, 119AH in a file created by MOVCPM, and 121AH in MOVCPM itself. The number of sectors is at 313AH, 11BAH, and 123AH respectively.

COMMENTS IN SUBMIT FILES

So you've got a submit file, but you can't remember exactly what it does..... Just put in comments as you would in any source file. CP/M will ignore anything from a semicolon(;) up to a carriage return. It will also ignore anything after most terminators (:, \, etc), but Digital Research don't promise to support anything other than a semicolon in future releases.

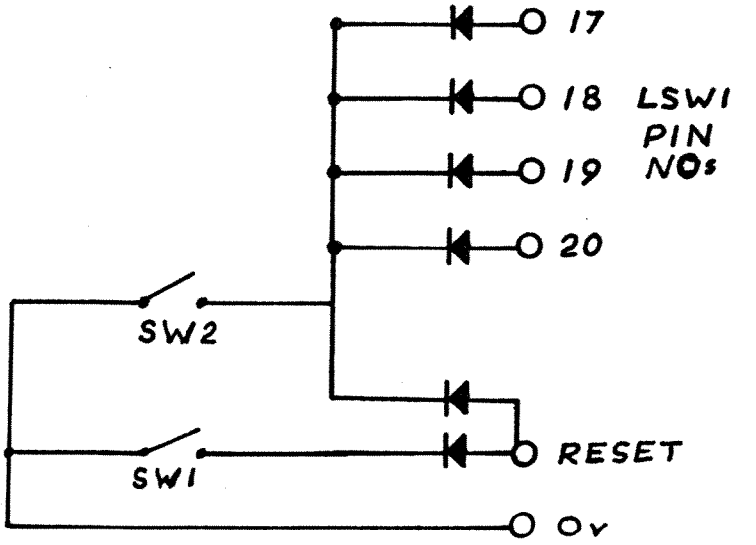
FAKE CALLS TO WBOOT

While on the subject of undocumented features of CP/M, on page 16 of INMC80/2 David Hunt wonders whether making fake calls to WBOOT in order locate the jump vector table, so as to make direct calls to the I/O drivers, is legal. Sorry Dave but it ain't. However, John Pierce of Digital Research has stated that since everybody uses it, they will try to support it in future releases, but it may not be possible, so they will continue to not document how to find the jump vector table.

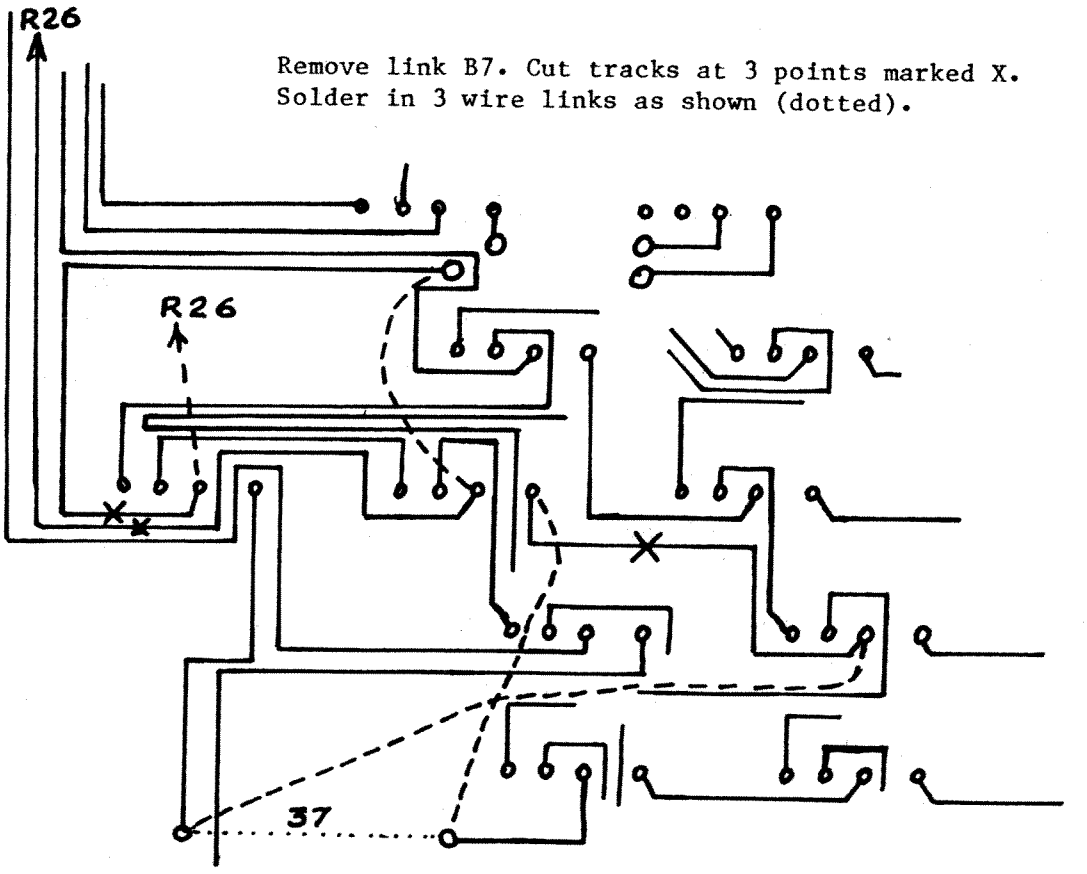
CP/M 1.4 contains two undocumented function codes that are documented in version 2 onwards. These are used to set (28) and monitor(29) the read only status of disk drives.

Uncle Dusty's Diagrams

Diagram 1.
Wiring for two
reset switches.
SW1 to 0000H
SW2 to F000H



Remove link B7. Cut tracks at 3 points marked X.
Solder in 3 wire links as shown (dotted).



DIAG 2.

FREE SOFTWARE?

Last summer I made a trip to my local friendly Nascom dealer to buy a copy of the source of SYS. Rather than PIP each file across by itself, the guy copied the entire disk, and then erased the files that I shouldn't have. These files included DISKPEN.

CP/M doesn't erase the whole file when a file is deleted, all it does is change the first byte of the file's directory entry from 00H to E5H, so when I got home, I just changed the byte on the erased files back to 00H using D-DOS. All I then had to do was find a way to stop the E and W commands from disappearing as soon I used `Insert', because, of course the program hadn't been initialized, and so didn't work properly. It was an easy job, all I had to do was alter a couple of RETurn's to CALL's. Have a look at your disk with the source of SYS on it, if you haven't recorded anything else onto it, you might be lucky too! (Oh know you won't, they're wise to that one, and the SYS source and DISKPEN are now on separate disks, anyway. Watch out, they'll be round to collect the royalties! - Ed.)

A couple of criticisms of DISKPEN. It needs a `continuous change' command, which is to `c' as `I' is to `i'. The `change keyboard case', U, needs to be duplicated by `u'. (Why, you lost the use of your shift key or something, or is it because it was ripped off and you haven't got the documentation? - Ed.) It would be nice if the `find' commands supported `wild' characters, as CP/M's CCP does, using ?? and *.

Incidently, just to save the editor's blushes, I must admit that I don't use DISKPEN. I use Michael Shrayer's Electric Pencil II as a text editor, I find it far preferable to DISKPEN except for it's lack of on-screen formatting, (It will only format what's being sent to the printer), but then, it is three times the price of DISKPEN, and only works with a 64 x 16 character screen.

USING THE TOP LINE

Wouldn't it be nice if we could use the top line of the video display as if it were any other line. Some of that information that gets missed because it scrolls off the screen before we manage to hit ^S could be read, if we could see sixteen lines of text instead of just fifteen. (See earlier `Ed.' - Ed.)

Having the screen handling part of the CBIOS in RAM makes it so easy to adapt. Instructions follow: First go to your local friendly dealer and purchase the source of SYS, well worth the tenner that it'll cost you, and using Electric Pencil, or whatever your favourite text editor is add the following code.

After

```
        VL15    EQU VRAM+038AH
```

add

```
        VL16    EQU VRAM+03CAH
```

Amongst the video switches add

```
        SCROLL16 EQU TRUE                ;16 lines scrolled
```

Just after the beginning of the video routines, you'll find

```
        ;Copy down screen.
                LD BC,VEND-VRAM-16-10-6
```

From this, remove the `-16'

A bit further on you'll find:

```
        ;Scroll-up
        CRT40:  LD DE,VL1
```

Between CRT40: and LD DE,VL1, add the following:

```
                IF SCROLL16
                LD  DE,VL16
                LD HL,VL1
                LD BC,VIDWID
                LDIR
                ENDIF
```

This extra code will cause the old top line to be copied up to the supranormal top line, where you always used to put headings and things, instead of just being discarded as before. The change just after "Copy down screen" causes the clear screen code to clear the top screen as well. The extra code carries a small time penalty, but in most cases this is insignificant. Performing a Clear Screen, or Cursor Home will cause the cursor to return to the beginning of the second line, as before, and not the new top line. Indeed it's not possible to place the cursor anywhere in the top line, but I've never yet found this to be a problem, and the extra code necessary to get it up there just isn't worth the trouble. It may be just one extra line but it makes quite a difference. Try it, you'll like it.

Oh, and while you're playing about with the SYS source, why not change a few things, such as creating a nice friendly sign-on message, and changing the cursor from a bland old underline to a flashing little man, is one of the hoopiest ideas around!

CP/M USER GROUP SOFTWARE

Well lads, one of the reasons that we all spent vast quantities of pennies on CP/M is that there's nearly 8 Megabytes of virtually free software available, isn't it? It's the CP/M user group library, and there's some really hoopy stuff in there, including assemblers, languages, applications software, utilities by the score, and games, most with full source code. It's worthwhile getting hold of several of these volumes, but just which ones depend on your own interests.

The early volumes (that is, upto about vol. 32) have in only a few cases heard of lower case, so you can either modify the source code to convert lower case to upper case, or you can try to remember to use upper case only.

Most of the assembler source is in 8080 code, or if it's for Z80, it rarely uses Zilog mnemonics. In many cases the syntax is for some strange assembler or another, that you probably haven't got, but that doesn't always matter, because there's usually the corresponding .COM file alongside. The later material seems to have standardised on D.R.'s 'MAC' macro assembler, which is upwardly compatible with ASM the assembler which is supplied with CP/M.

In common with most commercial software, the clear screen character is 1AH (SUB) and not 0AH (FF) as is used in a Nascom. I've got round this by altering SYS so that both characters clear the screen. However, 1AH is ^Z which CP/M uses as an end of file character, and this may cause problems, but I haven't encountered any in the three years that I've been using CP/M, so it's probably OK.

The CP/M User Group specify that their software mustn't be sold at a profit, the most that you should pay is the price of the empty disk, plus a small copying charge (which is usually about a pound per volume). The disks are difficult to get hold of on Nascom format disks, except single sided, and four disks per volume costs a fortune, especially if you want all sixty volumes! L.P. Enterpises over at Barking can supply them in Nascom format for 9.20 per volume (inc VAT) plus 3.45 per volume for insurance, postage and packing. Staubig Studio (of 12,Woburn Street, Ampthill, Bedford) can now supply disks in the format used by Gemini G805 (Henelec) single density disk controllers at 8.00 for a volume needing two disks, and 4.50 for some of the early volumes that'll fit onto one. P & P is an extra 75p for the first 4 disks, and 15p per disk for each extra disk. A catalogue of the disks presently available is 2.50.

books

BOOK REVIEWS

By Rory O'Farrell

Build Your Own Z80 Computer

By Steve Ciarcia
Published BYTE/MCGRAW HILL

This is a very readable book, by the author of the well known construction column in BYTE. It describes the various considerations in the design of a simple Z80 based machine and the monitor which would be necessary to drive it. It reprints various relevant data sheets, and discusses problems of driving capability of chips and timing specifications.

Z80 User's Manual

By Joseph Carr
Published Reward books (distrib. Prentice Hall)

This book discusses the considerations in the design of a Z80 based system, and interfacing to various peripherals. It also discusses the instruction set of the Z80 in much the same way as the Z80 Programming manual. Unfortunately, this latter section is marred by faulty proof reading and typesetting errors. As I have the Z80 programming manual, obviously I would prefer that the 200 pages devoted to the discussion of the instruction set were instead an extension of the 100 page section on interfacing and system design!

Interfacing Microcomputers to the Real World

By Sargent and Shoemaker
Published Addison Wesley

This book is similar to the preceeding two in that it sets out to tell you sufficient to design a Z80 based system, but puts particular emphasis on interfacing with external devices. In particular, it discusses A/D and D/A conversion in detail. It must be pointed out that the A/D chips they use are not necessarily the most easily available chips on the market in the U.K., but the general principles are relevant. In addition, they discuss the design of a suitable monitor, and give the listing of DEMON, a DEbug MONitor based on TDL's 2K Zapple monitor.

8080/Z80 Assembly Language

By Alan Miller
Published John Wiley

This book deals with programming both the 8080 and the Z80. In consequence it switches from one set of mnemonics to the other, but it is easy to keep abreast of what is happening. The author develops a monitor for these computers, and considers some of the problems in interfacing routines with CP/M. He is very readable, and the use of the two sets of mnemonics and the optimisations he shows for Z80s over 8080s are interesting and informative.

A few miscellaneous comments

Software Tools, which I reviewed in INMC80-5, has now been published as "Software Tools in Pascal" by Kernighan and Plauger, published Addison Wesley. I have not yet seen this book, but imagine it to be Software Tools rewritten with the programs in Pascal rather than RATFOR and PL/I. As such, I think it would be more relevant to Nascom users than the previous edition.

In using a microcomputer, we find ourselves reinventing the wheel at every step. We must remember that the path we tread towards the development of a usable computer system has already been forged by many trained and brilliant minds. Fortunately for us, they have, in many cases, published their thoughts and results. These findings are now being taught to computer students, and we can learn and indeed save ourselves much work by seizing onto this fact.

Perusal of the computer bookshelf in a good technical bookshop will often reveal titles which your friendly local computer store has never heard of, and which may be most relevant to you and the direction of development of your system. These books are usually text books, and make for very dry reading, but the information is there in them nevertheless.

A good plan is to read them a small bit at a time, allowing time for the information to sink in. They are not written in general about specific machines, and discuss instead the general concepts, which are applicable to all computers.

It would be beyond my power to review in depth a number of these books. I would like instead to mention some of them, with a short comment on the contents or field of applicability, in the hope that you can find a copy in your local bookshop or library to help you decide if it is relevant to you before purchase.

Algorithms + Data Structures = Programs — By Niklaus Wirth
Publ. Prentice Hall

This is by the father of Pascal, and is regarded as being one of the seminal books on the design of computer programs. Wirth is of the opinion that correct computing springs from a correct structuring of the data, and application to this of the best algorithm. His langage Pascal is based on this postulate. This book is regularly used as a teaching book in universities - even those who do not teach Pascal!

An Introduction to Data Structures with Applications — By Tremblay and Sorenson
Publ. McGraw Hill

This is perhaps the cheapest of the standard books on data structures, being available in a McGraw Hill Student edition for about 10.00 (some other data structure books run to 30.00!). It gives examples in PL/1, which is quite close to Pascal or Structured English. I'll quote its chapterheadings to give you an idea what it is about.

Information and its Storage Representation
The representation and manipulation of strings
Linear data structures and their Sequential Storage Representaion
Linear data structures and their Linked Storage Representation
Sorting and Searching
File Structures.

It is a hefty volume, running to just over 700 pages, and I am wading through it. It is not the lightest reading in the world, but it seems very complete.

Fundamentals of Operating Systems — By Lister
Publ. Macmillan

This book is concerned with the design of the operating systems (the monitor) for large or multiuser computer systems. With the advent of disks, and other expansions, such as possible multiple processors on a bus or multiple users on the same machine, as might happen in schools, this subject will become more and more relevant.

Closing Comments

Please bear in mind that these books are text books, and consider them well before you rush over to the cashdesk to purchase them. Perhaps you may find them absolutely indigestible! If you are minded to buy a book with a view to construction or design of some project, then I'd recommend Ciarcia or Sargent & Shoemaker. If the higher theory would be of interest, then I think Software Tools in Pascal would be a good starting point. I have not seen this particular volume, I stress again, but am quite confident of its general usefulness. In addition, we have two excellent Pascal Compilers available for the Nascom, the Hisoft Naspas and Polydata's. With a little bit of adjustment I feel sure the Software Tools programs will run on these.

Disks

DISKS on a NASCOM

by Bob Gilchrist

Most micro users would agree that disk storage, whether minifloppy (5.25") or floppy (8"), is an attractive if expensive proposition. In this article I am going to tackle some of the pros and cons of going to disks, with specific reference to my own experience with the Nascom II. It will end with a description of the system I eventually chose.

The advantages of disks I think are well enough known, but without labouring the point, the chief gain is in speed of access to programs and/or data. In many cases, this makes the machine capable of far more useful work. One only has to edit a few 28K byte articles on a cassette based system to begin to realise the limitations of that method of storage. Cassette files, particularly on the Nascom, are also difficult to keep track of. There is no index or directory to tell you what is on a particular tape, and most people have had to resort to manual methods of book keeping. A tape counter helps, but you very soon find yourself with an awful lot of tapes, cluttered up with things that you'll rarely use but which would take an age to sort out. A well known law also states that in the process of tidying up tapes you will delete the only copy of your most precious creation. You know the story.

So what are the choices for the Nascom II owner? At present there are three disk systems available for the Nascom, which, to avoid confusion, I'll refer to as the Henelec, Gemini and Nascom systems in order of their release.

First, the Henelec. This has been around the longest and is a really cheap system consisting of a small pcb which connects to the on-board PIO socket and then to the drives. The controller chip is the Western Digital WD1771, which has been around for some years now. One to three drives can be used. The software is written to support the Pertec FD250 drives, but minor alterations to the parameter table at the start of the FDC EPROM allows almost any Shugart compatible 5.25" single or double sided drive to be supported. The Pertec FD250s are 5.25" drives and they are double sided (there are recording heads on both sides of the disk). In common with most single/double density drives they have 35 tracks recorded at a density of 48 tracks to the inch. Using the Pertec drives with the Henelec system gives a formatted disk capacity of about 160K per drive. The Pertec drives are capable of storing data in single or double density, a possible advantage if future upgrading is a consideration. Obviously, double density stores twice as much as single density. However, due to the simplicity of the Henelec hardware only single density operation is possible. The board can be purchased as a kit (with the FDC software) and the drives can be obtained without a case or PSU (power supply). With a certain amount of `do it your self', a second hand drive, and D-DOS, you could provide a disk system for quite a bit less than 200.00. This is the cheapest route to disks. Gemini also sell this system, packaged in a nice case with the PSU, and known as the G805, costing about 395.00 + VAT, if you can afford it.

Four alternative operating systems are available for the Henelec: D-DOS, (`DOS' stands for Disk Operating System) a very simple system rather like using a cassette tape, (you still have to note where everything went) this is very cheap but a little tedious to use. DCS-DOS is a lot more elaborate and overcomes all the disadvantages of D-DOS - where a dedicated Nascom DOS is required DCS-DOS has a lot going for it. POLYDOS 1 is in many ways similar to DCS-DOS (but costs more). It has by far the best documentation of any DOS yet seen (even the professional ones) and has many nice enhancements over DCS-DOS. All three DOS'es leave the Nascom memory map alone and in most cases all existing Nascom software will run. Lastly there is CP/M 1.4 and latterly CP/M 2.2, costing about the same as POLYDOS. This changes the

memory map, meaning that none of the Nascom software will run without alteration, but is a de facto professional system, with built in software compatibility with other CP/M machines and lots of `off the shelf' software. A lot has been written about CP/M elsewhere.

The next is the Gemini double density system. This uses a NASBUS/ 80-BUS compatible 8" x 8" disk controller card which plugs into the bus. The controller chip is the Western Digital WD1797 and its attendant support chips, all very new devices. Port decoding is done on board and may be decoded to either ports starting at C0H or ports starting E0H. The card is capable of either single or double density operation. The card is called the G809, and is one of the Gemini `Multiboard' range. Software has been optimised for the same drives as above, but due to more sophisticated (and expensive) hardware, it is capable of operating in double density. This gives 350K bytes of formatted storage in each drive. Other drives, including 8", can be used but software is only available for the Pertec 5.25" drives. Gemini also supply drive boxes fitted with one or two Pertec drives and PSU. For those capable of `knocking up' a PSU and box, the drives and PSU being available separately.

There are two choices of software, POLYDOS 2 for Nascoms, which is similar to POLYDOS 1 (above) but with different disk controller software for the G809 card. Then there is CP/M 2.2 in two versions, one for the Nascom and one for the Gemini `Multiboard'. The reason for the two versions is because of the different video and keyboard requirements and because the I/O support (serial and parallel input/output) is different between the Nascom and Gemini machines. When using CP/M, media compatibility between the Henelec system and the Gemini system is maintained as the Gemini CP/M is `auto density seeking'. Put simply, the Gemini double density system, when presented with a disk it can not read in drive B, switches automatically to single density to retry before rejecting a disk as being unreadable. Thus the Gemini G809 system is capable of reading disks from either system.

The newest system on the market is the Nascom disk system. Originally dreamed about by Nascom in pre-receivership days and finally got into production by Lucas two years after the first specifications were drawn up. Like the Gemini this is a Nasbus 8" x 8" inch card which plugs into the bus. This card uses the Western Digital WD 1793 controller chip which has been around some time. The card is capable of either single or double density operation. Unlike the Henelec or Gemini, the Nascom system uses `double track density' drives, which write 80 tracks at a density of 96 tracks to the inch. The drives being only single sided, squeeze roughly the same amount of data on the one side of the disk as the Pertec squeezes on two sides. The drives are the TEAC TD-50C single sided drives and are supplied in a box with a built in PSU. Nascom have exhibited a drive box which matches their new Nascom case, but to date drives have been supplied in square metal boxes which, if you peel the labels off the box, you will discover were bought complete from Cumana in Ilford (they might be cheaper if you tried them direct). There is little advantage in double track density to the end user, except the drives are marginally cheaper than double sided drives (to buy in quantity - Nascom charge the same as Gemini for single drives bought over the counter). At present double track density causes media compatibility problems between the competing systems, as no control software has been written (yet) to allow the double track density drives to `double step' (96 tpi being double 48 tpi) thus allowing them to read single track density media. Even if double step control software were written it would not be possible for the Nascom system to write disks which could be read by single track density drives. No doubt some enterprising dealer will either tag a Pertec drive on a Nascom system or a TEAC drive on to a Gemini and write the necessary software to allow `cross fertilization' and charge for the priviledge. A complete Nascom disk system is almost exactly the same price as the equivalent Gemini system.

There are two DOS'es for the Nascom system. NAS-DOS is the earlier mentioned DCS-DOS with rewritten disk control software and allows existing Nascom software to be run. Then there is CP/M 2.2 for the Nascom system.

The choice between these three depends on your budget. You might perhaps be more inclined, as I was, toward the Gemini system as part of the whole "Multiboard" idea. I thought long and hard about these options. I also talked the whole thing over with fellow Nascom II owner. We agreed on the advantages of disks, but a definite problem emerged: Having disks opens the door to another operating system from NAS-SYS, namely CP/M. CP/M, if you didn't know, stands for Control Program for Microcomputers. You cannot go far in computing without hearing about CP/M sooner ar later. It has many followers, particularly in America. It was written 7 years ago and there is an enormous `user base' and a very efficient User Group, even in the UK. I don't propose to go into great depths about it here but briefly it does all of the huge amount of `house keeping' for you. True that NAS-DOS, DCS-DOS and POLYDOS also do the housekeeping equally as well. With D-DOS, you'd have to resort to pen and paper to record what is on your disk, and more to the point, where to find it! In my case, the final arbiter was the huge software base for CP/M.

CP/M is a good program, but it's not NAS-SYS. We all love NAS-SYS because of its simplicity, ease of use and modification, and screen editing. Can you give that up? I didn't want to. However, if one contemplates spending a good deal of money on a disk system, one wants to get the best out of it. CP/M has a good deal of free software in its user libraries as well as an exciting range of commercial software. Most of this expects to display output on a standard 24 x 80 VDU. Of course the Nascom display is only 16 x 48. This, in my opinion, was not on, so I considered the Gemini Video Board which gives 25 x 80. The decision to use this solved some of the operating system dilemma. There seems little point in hanging on to NAS-SYS with the video board, as they are not compatible. So I made the break to CP/M.

My choice was between the single density Henelec system and the double density Gemini as the Nascom one still hadn't appeared at the time I actually had the money in my pocket. Had the Nascom system been around the choice would have been even more agonising. I reasoned that I would save over 100.00 if I bought the bare drives and boards and built the power supply and mountings myself. Over a week of solid full time work (in my holidays) nearly disabused me of this idea, but it worked, well almost. When built, my Nascom II, now sporting an additional 64K RAM, the Gemini IVC and FDC cards and two Pertec drives, refused to work. A good deal of time was wasted looking for non-existent faults before I finally gave up and called the supplier of the bits, Henry's Radio. This is where I had a surprise. I imagined I was in for an arrogant rebuff; after all, I had not bought the recommended packaged system but had bought the bits and these had all been tested before sale. To my relief, they were very sympathetic. "Bring it in" they said. So I did, and they spent the best part of an entire day on it. They fixed it and it wasn't even my fault! There was a track fault on the Nascom II under the solder resist. It hadn't shown up before because it affected a signal (NASIO) that I had never used before. I was a happy fellow that day.

Readers of Computing Today may have read of one man's dissatisfaction of CP/M. Much of what he says is fair comment - it isn't easy to give up one operating environment for another, and I also mourn the loss of Debug. However, all is not lost; screen editing for example, is still possible under Gemini's implementation of CP/M, thanks to a few familiar names like Richard Beal and Mr Parkinson.

No mention of the CP/M system is complete without a reference to its extensive software library. As there are something like 300,000 users of the system, much software has been written over the years. There is a large club in the US and they have collected a library of 51 8" diskettes (I'd guess about ten megabytes worth!) of programs. Inevitably, not all of them appear useful. Some of the early programs have been superceded by improvements on later disks and thus there is some repetition. There are games and music programs as well as many utilities and languages. There is also a "Special Interest Group" (SIG) in the US with a wealth of interesting programs. Any of these disks are available through the UK CP/M Users Group, providing you are a

member (current cost 5.00 per annum). Incredibly, all you pay for this software is the media cost (the disk), a nominal copying charge (1.00) and the postage. You can even send in your own blank disks, in which case you pay only 1.50. I have in fact just done this and purchased disk No. 50, which contains, among other things, a Pascal compiler! I should mention that the speed of service is not far short of "by return post" and that the UK group themselves have four disks in their own library. In case you are wondering how I can read the 8" disks, the answer's simple: The library is also available on 5.25" disks, but in a single density format called "SD" format. This is the very format chosen for the Henelec system when it was originally designed, and is also the alternative format on the Gemini system. The Gemini system defaults to double density, but allows single density disks in the second drive. Thus all the library disks are easily read by these systems. In the case of the Gemini, you have the added advantage that you can copy the SD disks on to double density ones. In some cases, two of the library Volumes (each 8" disk is called a Volume) can fit on a single 350K 5.25" disk!

It is this compatibility that I find most exciting about my chosen system. I now have access to a great deal of software most of which is streets ahead of any that I have written. As Dave Hunt, Richard Beal and "Parky" all operate similar systems, a user of the Nascom/Gemini stands to learn much from these people who are, for many of us, mentors in microcomputing.

To summarise the system that I eventually ended up with:

Nascom II
2 Nascom 32K RAM (A) boards
Gemini IVC board
Gemini FDC board
2 Pertec FD250 5.25" drives

Costs for these can of course be found in the mags. In case I did not make it clear above, it is necessary to add extra transformers in the power supply to cope with the disk drives if separate drives are purchased without power supplies.

Anyone who has looked at or used CP/M will know that it was created to be essentially independant of particular hardware. In consequence, the parts of the system that handle the keyboard, disks and display etc, are all lumped together in one area called the Basic Input/Output System (BIOS). CP/M itself is ignorant about the particular hardware in use. A program run under CP/M doesn't know if the computer is a Heath, Superbrain or Horizon. It only knows that it has an 8080 or Z80 CPU. It is insulated (by the BIOS) from hardware dependant routines. The success of a CP/M implementation relies on a well written BIOS. You cannot screen edit on a Horizon, for example. When I say that there are two BIOS's for the Nascom/Gemini, and that one was written by Parky and the other by Richard Beal.....

The wonderful RB has also made the source of his BIOS available, which makes it easy to modify and reassemble. It has many useful features. The "Parky" BIOS is the one that you get with the Gemini and it, too, is every bit as good as RB's, except that the source code is not provided making it difficult to modify certain aspects of it, although a `patch' area is provided for easy selection of serial/parallel printers etc. Gemini say that the listing will be available `soon'.

I hope my efforts to describe the competing systems will be of help to somebody, and if asked to give an opinion, I think the best of all versions is a Nascom II fitted with the Gemini disk system, if only because the source for the alternative BIOS, SYS, is available. It makes a very worth while and powerful system. In all it must have cost 1500.00 but you name another machine with as much for less than double that price. (Sorry, we can name a couple, but none are so much fun as one you build yourself. Ed.)

CB

Breaking Computers

D. R. Hunt

Somewhere in this issue Dr. Dark mentions the use of CB for data transmission. The editor pointed this out to me, knowing my interest, and suggested I might like to write a reply. I, like a mug, said I would, as it would be the only words from my Nascom in this issue, part 6 of the Dodo's guide being only half written at this stage. No, I haven't bowed out because I committed a ricket in the last Dodo's Guide, that one was a major disater. If you're one of the 5,000 odd people who DIDN'T write to me to tell me, you'd better read issue 5 again. Anyway, I set out to examine the practicalities (or otherwise) of the idea of using CB for data transmission.

Well first of all lets look at the snags. The major snag to the idea is that under the terms of the CB licence, it's illegal!! Paragraph 6 of the expensive piece of blue paper says:

"6. All transmissions other than selective calling signals or digital transmissions designed solely to identify the transmitter, shall be in plain speech only."

If being outside the law doesn't dampen the enthusiasm let's find a few more areas of trouble. Effective range, I haven't done any calculations to find out what would be a tolerable noise level, but for acceptable data recovery without the use of clever and complicated equipment, the received transmission must be at least reasonably free from interference. Now with the legal FM mode of CB transmission employed in this country this is slightly less of a problem than it would have been with an AM system, as the FM capture ratio ensures that if a signal is a minimum of 3dB better than any interfering signal then the required signal will be received to the reasonable exclusion of the interfering signal. Comforting news, so where's the problem? Well around Dr. Dark's `twenty' (location or area for those uninitiated in CB slang) this may well not be a problem. I have heard reports of interference free reception over fifteen miles out in the sticks, but around North West London, fifteen metres is nearer the mark. Competition for channel space is both fierce and ruthless to the exclusion of good radio operating practice and good manners. With the exception of channel 9, the voluntary `Emergency channel' (which round our way remains remarkably and inexplicably free), each channel may have anything up to four simultaneous transmissions at various receivable ranges going on at any one time throughout the 24 hours. This limits the practical range, bearing in mind the effective capture ratio effect, to about two miles radius. I know of about seven Nascom owners in my area which would be within acceptable range but I think my situation would be the exception rather than the rule.

Now we come to the ungentlemanly behaviour of a minority of the local `breakers' (CB users). I have to say `minority' or I'll be drummed out of the Lima Bravo Breakers Club. (The LB club members, myself included, consider ourselves a cut above the rest, being firstly `more mature in age' and secondly, being mainly engineers, bona fide radio amateurs, and the like.) In my opinion, round our way the `minority' is about 70% of the whole. I don't need to try it to surmise that any attempt at data transmission round my way would cause the more antisocial elements within the fraternity to commence jamming operations within minutes, or at the very least, it would result in numerous well intentioned but uninformed `breakers on the side' (someone trying to get in on the transmission) coming up trying to find out what the heck was going on. In either instance, with it's attendant effect on the degradation of the signal.

So, for the determined, how may these problems be tackled. Well the first obstacle is the legality, plainly it would be illegal, and far be it from me to incite anyone to break the law, so what follows are my thoughts, AND ONLY MY THOUGHTS on how the other obstacles may be overcome, sadly, without due regard to the law. It's obvious from the above lots of power would be an advantage and that some disguise must be adopted to avoid deliberate interference.

Apart from the illegality of sending data over the air, it's plainly obvious that other conditions of the CB licence are being openly flouted round my way, judging

by the ever increasing number of half wave dipole aerials which are appearing on the roof tops. It surprises me that Busby hasn't done himself a severe damage by getting his undercarriage entangled in the ever growing forest of `big twigs' which are sprouting and pointing skywards. Now the reason for the increasing number of large aerials is because the local breakers are taking it upon themselves to overcome interference from other local stations. The regulations restict the user to an inefficient aerial system, a base loaded 1.5 metre dipole. Now a halfwave dipole (which is some 5 and a bit metres long) is essentially the `fundamental radiator' of radio signals and is very efficient. By buying a half wave aerial, you obviously `get out' (radiate or transmit) better, meaning that the person you are in contact with will receive you better as your signal is now stronger that the others and thus overcoming the others by taking advantage of the capture ratio effect. However, this simply aggravates the problem as your strong signals will blot out the signals from the more legal breakers around (with 1.5 metre base loaded aerials), who in turn will be persuaded to go and equip themselves with larger aerials likewise. And so we the problem gets worse not better.

The same thing happens when `boots' or `burners' are used (linear power amplifiers), the local CB emporia are chock full of linears ranging from 10 to 500 watts. Up to about 50 watts a linear is relatively cheap, certainly not much more than the cost of a couple of tanks full of petrol (which will be burnt by a mobile breaker cruising round the country side trying to get good `copies' (contacts) over the following few weeks anyway). So someone puts 25 watt boots on his big twig, and surprise surprise, for a few weeks he can get out over the top of the rest. The average conversation goes as follows:

"On the side. On the side. Who's the breaker bending my needle."
"You got Bulltwit, good buddy. What's yer handle?"
"Ya got Brain Damage here. Say, thats some signal you're putting on me. What's yer twenty?"
"Willesden. What's yours?"
"You're coming across wall to wall in Finchley, good buddy. What's your rig and you pushing `wellies' or somethin'?"
"Roger-dee. The rigs a Harrier and I'm poking 50 watts up a Procom GP27 half wave."
"That so, Bulltwit, Yeah, I'm thinkin' of getting some `wellies', got any ideas?"
..... and so on

I'm not saying that that is a typical conversation, but it does follow sufficiently well established lines for us to call these types `Three What Breakers'; that is the `Three Whats', Whats your handle?, Whats your twenty? and Whats your rig? ... end of conversation. So not only do all the locals start running large aerials, but the power goes up progressively as well. This would seem to be a never ending spiral, limited only by the necessary funds for the equipment and the (unlikely) intervention of the Radio Regulatory Division of the Home Office.

To digress for a moment, what I'm about to write is perfectly true. A breaker round our way who was already running 200 watts decided to increase this already over adequate power further by fitting a `Bremi 500', about the biggest 500 watt linear around. It worked, so he thought why not drive the Bremi 500 from his existing 200 watt linear which if it worked, would, according to his simple mind give him something like 10KW output. Well he didn't expect the arrangement to be very efficient, but thought he might get a couple of kilowatts if he was lucky. What he did get was a lovely explosion and following fire which needed the attention of the local fire brigade. It completely gutted his living room.

So the first requirement would be an efficient aerial system and bags of power to punch the data over and above the rest.

Now what about the locals who are not likely to tolerate data transmission anyway. This can be overcome by the mode of transmission employed. Now it would be quite feasible to transmit the Nascom tape signal. The audio bandwidth required would be 2.4KHz which is approaching the limits of the capability of the CB system, but provided the deviation were kept to sensible levels the modulation index would not be exceeded thus ensuring adjacent channel interference would be kept within the prescribed limits. The data rate had better be kept to 300 BAUD for greater data integrity, but at short range this should be acceptable (excepting that it might be

quicker to walk round to the person concerned and give him a tape of the data than to go to the bother of setting up a CB communications link).

As already proposed, straight transmission of data is vulnerable and likely to attract the attention of jammers, etc, if it is obvious. So a more sneaky method is required. One method would be to transmit the data as a sub-audible Pulse Code Modulation signal underneath a conversation. This would require the construction of extra encoding and decoding equipment to attach to the computer, and would cause the generation of extra sideband energy which in turn would require a reduction in deviation if the modulation index were not to be exceeded. However, it has the advantage of being undetectable unless someone is looking for it. Because of the very restricted bandwidth available to the PCM signal the data rate would be necessarily be slow. That means that who ever is talking on the speech part of the carrier had better have a lot to say as the data transfer could last from minutes to hours.

Another method would use a duplex mode of transmission where the two computers would work a kind of handshake. First, it should be noted that most CB rigs use Binary Coded Decimal channel change logic, so the channel switch could be removed and the channel change circuitry fed from the computer's PIO device instead. The transmit - receive circuitry must also come under computer control from a port. The system must use a synchronized clock, and when this was discussed some time ago, it was suggested the the Rugby MSF time transmission be used for the purpose. At that time the only circuit around (short of designing ones own) was one published in Wireless World about three years back. This had the disadvantages of being both complex and expensive. However, Radio and Electronics World, April 82, have published an interesting design which should suit. (The R & EW article is good reading because the MSF receiver feeds another board which decodes the signal, and it uses a Z80 to do it. Food for thought??) Anyway, here's how the idea works:

The programs in both computers are set to a standard time point for the data transfer to commence, say on the hour.

At that time both computers look up the same number from identical tables of random numbers between 1 and 40 and select the channel chosen.

The sending computer switches the rig to transmit and sends a burst of data, staring with a numbered header, for less than one second.

At the next second time blip from the MSF receiver, the receiving computer switches to transmit and acknowledges the receipt of the data, correct, otherwise or not at all.

On the next second time blip, the two computers look up the next channel number in the random table and change channel.

The sending computer now knows one of three conditions:

1) Data received correct
2) Data received but incorrect or no data received
3) No acknowledgement received from the receiving computer

In condition 1, the next block of data is sent and the sequence repeats

In condition 2, the same block of data is sent again and the sequence repeats

In condition 3, as in condition 2

So it can be seen that the data will eventually be sent, albeit slowly, by the use of all the available channels for a minimum period only. Minimum interference would be caused to other users, and any attempt at jamming would be futile as the random channel table would be sufficiently long to make following the signal impractical. Also, because of the short duration of the data and the constantly changing frequency, it would be very difficult to pinpoint the actual location of the tranmitter without using broad band direction finding equipment. This method of transmission would probably be slower than the sub-audible PCM method, but is simpler and could be used overnight for instance. Don't forget, computers are patient beasts and will keep trying until the job is complete.

So what is my opinion? Well I don't think it's worth trying, round here at any rate. Does anyone else have any sneaky ideas which would pass unnoticed. I suppose the only answer would be to take the RAE exam and get an amateur ticket. Although that won't help much, amateurs are restricted to Baudot code at a maximum of 50 BAUD, and then plain text only. Mind you, it would be an interesting exercise to think of a way of cramming eight bit data into the five bit Baudot code, some kind of four level shift I suppose.

Multi-Map

MULTIMAPPING FOR NASCOM 2 - UPDATED

C. Bowden

In INMC80-3 I described my method of switching between the NAS-SYS and CP/M operating systems. Since then, there have been a number of exciting developments in the hardware field that have greatly increased the power and flexibility of the NASBUS/80-BUS system. I refer in particular to Gemini's IVC card, Double Density FDC card and 64K RAM card.

At the same time, one should not forget that for medium scale software development, with a more `intimate' and `operator friendly' feel, the standard Nascom packages take a lot of beating. The larger characters on the 48 x 16 screen are also better when introducing the computer to newcomers or older people. There are also a number of interesting new pieces of software like POLYDOS that make the old system far from redundant.

In INMC80-4 Mr David Parkinson described his very powerful system for `Booting' all firmware from a `paged' EPROM card. This system permits one to run all ROM/EPROM Software including BASIC and NAS-SYS in RAM. For full implementation 64K of RAM is needed, but variations are possible. I have combined Mr Parkinson's idea, and the core of my original scheme to form a simpler yet more powerful method of switching between systems and formats.

The following attractive features are possible:

1) Very little extra `wiring' needed to permit system switching.
2) Switch selection between NAS-SYS, CP/M 48 x 16 and CP/M 80 x 25 video formats.
3) A full 64K of RAM may be used.
4) The RAM has no `gaps' in it due to ROMs.
5) With one EPROM card, 32K of EPROM and 8K ROM may be used.
6) No WAIT states need be used on the CPU card.
7) It is possible to `overwrite' parts of firmware on a temporary basis.

The essential components required to carry out this scheme, at least in part, are:

1) An EPROM card. (Page Mode).
2) Access to an EPROM programmer and ideally, an eraser. I use a small `Germicidal' type of UV tube. (CAUTION - DO NOT VIEW THE LIGHT).
3) A copy of the article in INMC80-4, and understanding how it works.

The first requirement when using two or more completely different types of operating systems is that RESET takes one back to the system in use, and not the alternative(s). This is achieved in my system by using two `BOOT' EPROMS, and changing the RESET jump. The EPROMS are:

BOOT EPROM A

located at 2000H on EPROM card. This EPROM copies `SIMON' to RAM at address F000H, and then jumps to F000H, paging out the EPROM card on the way.

BOOT EPROM B

located at 4000H on EPROM card, which copies NAS-SYS to RAM at 0000H, and then jumps to it, paging out the EPROM card.

It is a very simple matter to arrange the RESET jump to be switched to suit the system BOOT required, if the BOOT ROM's are located at EPROM CARD address 2000H, 4000H or 8000H. Other addresses will require more complex switching.

The EPROM holding the NAS-SYS BOOT software also holds the 'MENU', key selection and 'copy across' software for the selected program. In my case this is in a 2716, and there is over 1K still free so I have incorporated a number of 'utility' routines, some of which are described below. The program required to copy 'SIMON' across is very short and the remainder of the EPROM may be used to hold whatever other software the user desires. (This may be software not relating to CP/M, such as more utilities). SIMON is the BOOT software for the Gemini CP/M 2.2 implementation which may be overwritten once it has done it's job. Users of CP/M 1.4 could copy the relevant software across, but it has to be available the whole time, and so the top 4K of RAM would have to be left free.

Whilst it is normally the practice to copy programs across to RAM, it is quite possible to 'page in' the EPROM card and run software on it (provided it's address is correct), but if there is any quantity of software on the card, this soon becomes impracticable since all software on the card will be on at the same time.

The Gemini G809 CP/M system, once booted, needs no further interaction with the EPROM card, since all the software including the operating systems are on disk. The NAS-SYS system needs frequent access to the card. Indeed the speed of access and clean memory map are amongst the main reasons for using the system. It is therefore necessary to be able to access the 'MENU' easily. One method is to use the 'E' command to access the card via a small 'turn on' routine. A better method is to change the allocation of a command key. This can be done either by modifying the EPROM card copy of NAS-SYS , or by adding a little routine to the software in the BOOT EPROM that alters the copy of NAS-SYS that it has just copied into RAM. I decided to use key 'D' to access the MENU since the 'cold start' of ROM ZEAP is now redundant, so location 0789H needs to be changed to 08H. (NS1 users change 078E and 078F to 00H, 08H respectively). The software needed to turn on and jump to the MENU is only 7 bytes long (p44 INMC80-4). I placed this routine in the VIDEO margin at 0800H - 0806H, and have not had any problems by so doing.

To give an idea of the versatility of the system, the 'utility' routines that I can access from the MENU at present are:

1) Handshake for the Imp printer (see below).
2) Memory test (see below).
3) Basic renumber.
4) Set PIO ports.
5) EPROM programmer (home-brew).
6) Block compare.
7) Find bytes.
8) Improved cassette read.

(The last three were described in recent issues of the MICROPOWER mag.)

Firmware that can be instantly down loaded includes at present, T4 (so that the old favourites like LOLLIPOP LADY can be run to amuse the juniors), ZEAP (RAM version), NAS DIS-DEBUG, and NASCOM ROM BASIC. In the near future I hope to add POLYDOS 2 to the firmware; and a real time clock and auto line numbering for BASIC to the utilities.

HARDWARE IMPLEMENTATION

Refer to the circuit diagram. It will be seen that the wiring is very simple. In my equipment, a three pole three way switch (obtainable from most electronics catalogues) is used to perform three functions:

A) S1A. Position 1: VWRAM to 0800H for NAS-SYS with 48 x 16 screen.
2: VWRAM to F800H for CP/M with 48 x 16 screen.
3: VWRAM disabled. IVC card in use.

B) S1B. Position 1: VIDEO O/P socket selected to NASCOM 2 by relay.
2: As position 1.
3: VIDEO O/P socket selected to IVC card by relay.

C) S1C. Position 1: RESET JUMP to 4000H to 'boot' NAS-SYS.
2: RESET JUMP to 2000H to 'boot' CP/M.
3: As position 2.

LSW1/1 and LSW1/4 Up
LSW1/2 and LSW1/3 Down

The arm of S1A should be connected to LKS1 pin 2.
Pole 1 to LKS1 pin 15.
Pole 2 to LKS1 pin 13.

The arm of S1C should be connected to common negative.
Pole 1 to LSW1/2 pin 19 (or IC2 pin 6).
Poles 2 and 3 to LSW1/3 pin 18 (or IC2 pin 13).

S1B is unnecessary if no IVC card/VIDEO monitor is used.

(N.B. From chats with other N2 owners it is clear that quite a few do not realize that it is quite permissible to use 64K of RAM and to overlap this with RAM or ROM located on the CPU card. The NASCOM 2 produces a signal that disables any RAM external to the CPU card that is decoded at the SAME address as RAM or ROM on the CPU card. BUS conflict is thus avoided. It would therefore be quite in order to insert even more ROM's onto the N2 card and to access these by extra switching.)

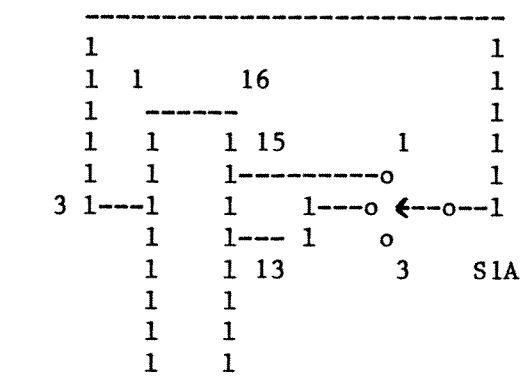
SOFTWARE NOTES

A few points that may be of interest:

Memory test. The test that I use is the one supplied with the RAM B and 64K RAM cards. A full assembly listing was supplied with RAM B (the assembly was correct but there was an error in the HEX listing that I received). This test is very quick and useful but there is a snag. It will not run with NAS-SYS in RAM as it stands. (The one with the 64K RAM card had the 'bug' removed - Ed.)The reason for this may be found near the end of part 1 of the test. At location 0CD9H is an LDI instruction. This moves bytes pointed to by HL to the location pointed to by DE, whilst counting in BC. In the memory test, DE is not being used for anything relevant at this time, but it is pointing to low memory. When NAS-SYS is in ROM no ill effects occur, but when it is in RAM it gets corrupted and as soon as the program reaches the RST PRS at 0CDEH 'CRASH'. The problem may be resolved by removing the LDI and replacing it with:

```
INC HL
DEC BC
LD A,B
OR C
JR NZ,ZCHK
```

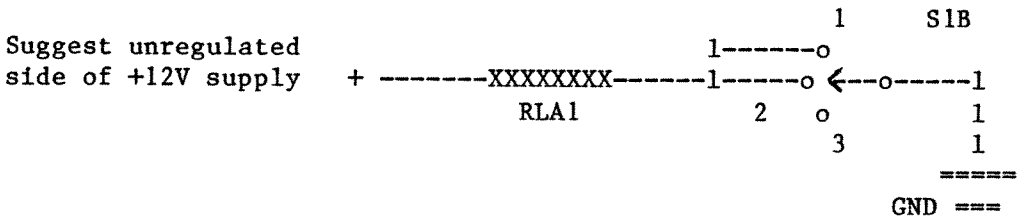
MULTIMAPPING REVISITED - CONNECTION DIAGRAM

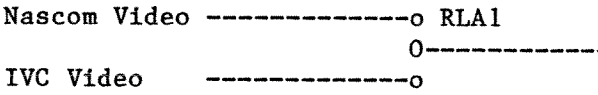


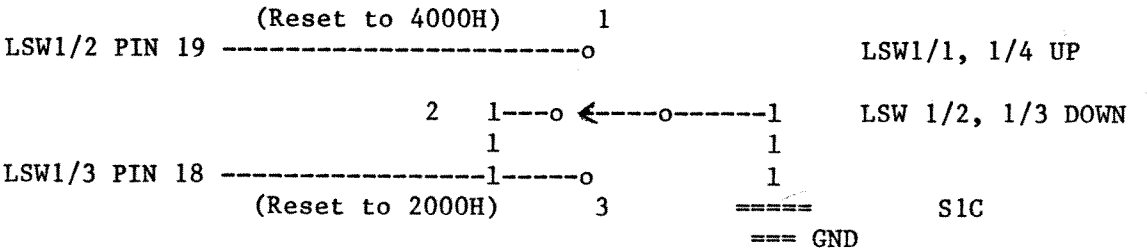
S1 - 3 POLE, 3 WAY ROTARY

RLA - 12 volt with at least 1 changeover contact. e.g. RS 347-832

LKS1 - No other connections needed.







Position 1 - Nas-Sys and normal Nascom Video/Workspace
" 2 - CP/M " Nascom Video/Workspace at F800H
" 3 - CP/M " Gemini GM812 IVC card

This is for use with 64K of RAM, however made up. If a lesser amount of RAM is used, certain firmware would have to reside on either the Nascom 2 board, RAM A card or a 'Paged in' EPROM card, and switching would need to be modified accordingly.

If you have the assembly listing you can type it in, modify it and reassemble. Or you can put a 2 byte JR to the end of the program in place of the LDI, and put the new code at the end with conditional and unconditional jumps back to the required places. If you adopt the latter course, note that the code proper ends at 0D48H with a C9, but the next 2 bytes are used as a store.

BASIC IN RAM. This one led to some embarrassment for me because I overlooked the cause and made a long distance telephone call to someone who kindly pointed out my `basic' error. When BASIC is executed it asks how much RAM to use. If you do not wish to reserve any and press enter, BASIC will `measure' the RAM. If there is no ROM or `empty' RAM at the top, just below BASIC, the NASCOM ROM BASIC will corrupt itself as it is not protected against this possibility (since it expects to be in ROM). The solution is to enter a limit of say 55000 in response to the query SIZE?.

NAS-SYS HANDSHAKE. One of my very few `hates' about NASCOM Software has been the need to have a number of `USER' routines, each one marginally different, to enable printer handshaking to ZEAP, NASPEN, NAS-SYS etc. Since it is now easy to overwrite parts of the firmware on a temporary basis, I have included a routine, accessed from my MENU, that provides H/S by altering the SRLX routine in NAS-SYS. The SRLX routine is used to send bytes to the serial port, so it is used by all devices talking to a serial printer. I use an IMP printer at present, so my routine is written to suit this printer but the software can easily be changed if necessary. The space allocated to SRLX is not sufficient to allow the modified routine to be put in the same locations, and the next routines may be needed by the Programs that are using the Printer. One routine that is most unlikely to be required during the execution of a program is the `M' routine. My overwrite routine therefore looks like this (NAS-SYS 3):

```
005BH   C3 80 02        SRLX:   JP SRLXA        ; Alter start of SRLX

0280H   F5              SRLXA:  PUSH AF         ; Overwrite start of `MODIFY'
0281H   DB 00           BUSY:   IN A,(00)       ; See if busy
0283H   E6 80                   AND 80H
0285H   28 FA                   JR Z,BUSY
0287H   F1                      POP AF          ; Restore byte
0288H   D3 01                   OUT (01),A      ; Send it
028AH   DB 02           WAIT:   IN A,(02)       ; Wait until gone
028CH   CB 77                   BIT 6,A
028EH   28 FA                   JR Z,WAIT
0290H   C9                      RET             ; Back to program
```

At the end of the program, a RESET restores NAS-SYS to its unmodified state. The same overwrite routine could be used with NS1 but addresses would have to be changed.

This system has been in use for about three months now and the ability to switch rapidly between systems and formats has been very useful. Particular advantages over the old MULTIMAPPING system have been the ability to use 64K of RAM, to load NS firmware and utilities instantly and to overwrite NAS-SYS and to restore it at will. It is also very useful to be able to load software into memory under one operating system, and to switch to the other system although this was also possible with the old MM setup.

N1 Graphics

Still Cheaper Nascom 1 Graphics — by Theo Zelders, The Netherlands

The Nascom 1 character generator is in fact effectively a 2 kByte ROM. When it is replaced by a 2732 EPROM (and this is very easy, as you will see) there is room for another 2 kBytes to accept the information of the Graphics ROM. The 128 characters of the character generator are addressed with 7 bits only. The two lines necessary for input and output of the 8th bit (usually called bit 7) can be connected to the two NC pins of the original 6576 socket. Only one line on the Nascom 1 board has to be broken, and in case you do not like that, you can always bend pin 6 of IC 15 outside and leave the board as it is. This pin 6 of IC 15 has to be connected to pin 10 (NC) of the char. gen. socket. It is the output of the 8th bit. A few extra connections have to be made: Pin 12 of IC 20 (8th data bit) to pin 18 of IC 17 (74LS273 byte latch); pin 19 of IC 17 (output of 8th bit) to pin 14 of the char. gen. socket. That's all for the board.

Now you have to make an adapter socket for the 2732 EPROM. The pin layout's of the two IC's are totally different. The easiest way to solve this is to mount two 24 pin wire wrap sockets alongside each other in such a way that they fit best to the Nascom. Later on you will cut the pins of the 2732-accepting socket so you can push the other socket (with the long pins) into the socket of the char. generator, and you will find the other socket floating somewhere over your Nascom board.

Now interconnect the two sockets according to the following scheme:

long pin socket	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
short pin socket	-	24	-	19	15	13	10	22	23	17	1	2	12 18 20	21	4	3	9	11	14	16	8	7	6	5

That is all, apart from programming the 2732. How do you get the bytes for this EPROM? The easiest way is by copying it from someone who did the job earlier. But some one must be the first. The most difficult way is programming it by hand by coding each byte in turn from the Nascom-documentation and the characters published in an earlier Newsletter. This is easy but time consuming. I did it earlier, at least partially, and it turned out to be fun. The characters are built from 16 consecutive bytes in the EPROM. Each byte defines a line of eight dots. A bit is set when the corresponding dot is white; zero in case the dot has to be black. Highest order bit corresponds with leftmost dot. Each character is addressed with it's own number from 0 to 255.

There is one other way to at least get the original contents of the character generator IC. Before you cut the pins of the 2732 socket, push it into a 2732 socket somewhere on your system; connect the two extra supply potentials (+12 and -5) to the appropriate pins of the 6576 socket and read the characters into computer-memory or onto tape. Also a 2716 socket will do. (By the way, I converted the 4 2708 sockets on my memory board into 2732 sockets and now I can choose by switch which of the EPROMmed utilities in the 4 2732's I will use by simply pulling the chip-enable down when I need one).

I wish you success, and fun, in programming chess and other symbols into your Nascom.

Really??

RANDOM RUMOURS (& TRUTHS)

By S. Monger

It would appear that Quantum Micros decided that they were so busy with their Quantum 2000 (a rather smart looking three (yes 3!!) drive system based on Gemini MultiBoard) that they have handed-over/sold/given (??) rights of their I/O card to Gemini. This card, given the Gemini magic number of GM816, will come built and tested with 3 Z80A PIOs, 1 CTC, and a battery backed Real Time Clock. (Seems that those blasted RTC things are about to appear everywhere, as Gemini are also about to produce a mini one that attaches to the Nascom/Gemini PIO.) The rather neat thing about the Gemini (nee Quantum) I/O board is that, as all the I/O decoding and buffering has already been done, it contains an on-board 50 odd way bus, and holes to take 'piggy-back' sub-boards. This gives the potential of all sorts of natty I/O, the only snag being that you probably can only have one 'piggy-back' per I/O board. Gemini say availability of both the I/O and RTC boards will be late March/early April.

All sorts of rumours are floating about re. the Nascom AVC (Advanced Video Controller) board. Production has been put back several times, and the latest 'promise' appears to be late April/early May. One thing for sure is that it was not to be seen at a recent London show. The provisional (rumoured) spec. sounds pretty impressive, but the complexity of the board seems to have turned it into a 10"x8" board. Now let's see, if we take a Nascom 2 (12x8), an AVC (10x8), a RAM card (8x8), and an Arfon speech board (5x8), we can design a 'Pyramid' case!! The board is said to have 48K of RAM on board (16K per colour) which should give very pretty pictures (software allowing), but there seem to be conflicting reports as to whether it can handle text or not - hmmmm.

Any visitor to Gemini a couple of weeks ago may have recognised the faces of most (all?) of the 'MicroValue' dealers. (By the way 'Micro' means very little/small doesn't it.....) It seems that they were picking up their first production Gemini Galaxy 1 units, or was there some scheming going on? The Galaxy, by the way, is a collection of MultiBoard cards (CPU/RAM/IVC/FDC), 2 Micropolis 400K drives, and a switch mode PSU in a metal case. The keyboard is in a separate unit. The whole thing looks as sturdy as a brick ****-*****, and comes complete with a fair range of software. Can a British company out-Osborne Osborne? At the price (1450 + Maggie's bit) it seems to be priced only a little lower than the sum of the bits, but of course there are lots of nice looking connectors on the back (they generally cost quite a bit) and you don't have to assemble the lot yourself. Going rather up-market for us hobby-types, though.

Nascom 3 also seems to have been sneaked out fairly quietly. Lot's of rumours were floating around about that, but none of them appear to have been right. It is in fact a Nascom 2 in a box, although you do get the (obligatory) Graphics ROM and also Nas-Sys 3. The time was well overdue that a cased Nascom become available, (if you remember, 'System 80' would have arrived, had the receiver not...) and the case is also going to be available to those who want to construct their own N3s. (By the way, MultiBoard owners will be able to get hold of the Galaxy case.) Will Nascom 4 be Nascom 3 with an AVC??

Watch this space for further ill-informed rumours.

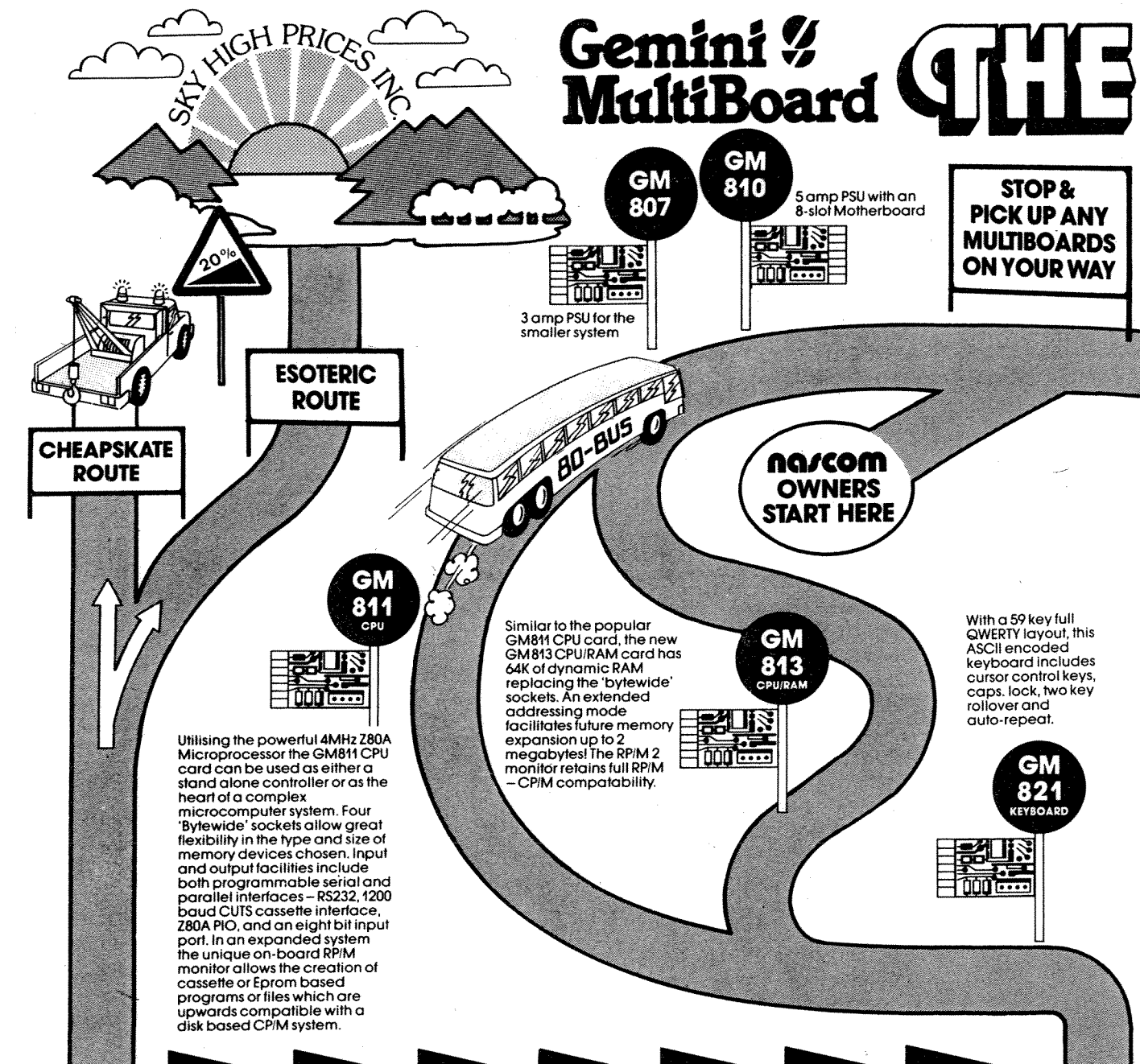
N2 program

```
100 REM **  The TOWERS of HANOI  **
110 REM     *  by John Waddell  *
120 REM
130 CLS:GOSUB 950
140 SCREEN 2,2:PRINT "5 discs, all different ";
150 PRINT "in size, are placed in"
160 PRINT "order on a fixed peg, with the ";
170 PRINT "largest at the":PRINT "bottom. ";
180 PRINT "There are 2 empty pegs in line with"
190 PRINT "the first, and the task to be ";
200 PRINT "accomplished is":PRINT"to transfer";
210 PRINT " all the discs to peg 3 so that at"
220 PRINT "no time is a larger disc resting ";
230 PRINT "on a smaller.":PRINT:PRINT
240 REM  ***  Set up pegs and discs  ***
250 INPUT "When ready, press ENTER. Ready";A$
255 CLEAR 500
260 CLS:DIM D(5),D$(5),P(3):GOSUB 950
261 REM * Array D for discs, P for pegs *
262 C$=CHR$(128):E$=C$+C$
263 SP$="       ":B$="                "
264 FOR I = 1 TO 5
265 SP$=LEFT$(SP$,LEN(SP$)-1)
266 C$=C$+E$
267 D$(I)=SP$+C$+SP$
268 NEXT
270 FOR I = 1 TO 5:D(I)=1:NEXT
280 P(1)=5:P(2)=0:P(3)=0
290 FOR I = 0 TO 2:SCREEN 6+16*I,2
295 PRINT "Peg";I+1:NEXT
300 SCREEN 1,10:FOR I = 1 TO 47
310 PRINT CHR$(129);:NEXT:PRINT
320 FOR Y = 8 TO 26
325 SET(14,Y):SET(46,Y):SET(78,Y)
330 SET(15,Y):SET(47,Y):SET(79,Y):NEXT
340 FOR A = 9 TO 5 STEP -1:SCREEN 2,A
350 PRINT D$(A-4):NEXT
359 REM *  Main loop  -  to get moves  *
360 I=0
370 I=I+1:SCREEN 1,13:PRINT SPC(40)
380 PRINT:SCREEN 1,12:PRINT SPC(45)
390 PRINT:SCREEN 1,12:PRINT "Move";I,
400 INPUT "From peg";Q$:Q=VAL(Q$)
410 IF Q=1 OR Q=2 OR Q=3 THEN 430
420 PRINT"Be sensible!                ":GOTO 380
430 H1=P(Q):IF H1 > 0 THEN 460
440 PRINT "There is no disc on peg";Q;"."
450 GOTO 380
460 PRINT:SCREEN 32,12
470 INPUT "To peg";R$:R=VAL(R$)
480 IF R=1 OR R=2 OR R=3 THEN 510
490 PRINT "That is not possible.       "
500 SCREEN 30,12:PRINT SPC(15):GOTO 460
510 IF R<>Q THEN 540
520 PRINT "That means stay as you are!"
530 GOTO 460
539 REM  *  Check whether to permitted peg  *
540 H2=P(R):FOR K = 1 TO 5
550 IF D(K)=Q THEN 570
560 NEXT
```

```
570 FOR M = 1 TO 5
580 IF D(M)=R THEN 600
590 NEXT
600 IF K < M THEN 630
610 PRINT "This move is forbidden.      "
620 GOTO 380
629 REM  **  Set up disc removal  **
630 B=10-H1:A=2+16*(Q-1)
640 FOR F=1 TO 400:NEXT
650 SCREEN A,B:PRINT B$
652 Y1=29-3*H1:X=15+32*(Q-1)
654 FOR Y = 8 TO Y1
656 SET(X,Y):SET(X-1,Y)
658 NEXT
659 REM  **  Set up disc reappearance  **
660 FOR F = 1 TO 400:NEXT
670 B=9-H2:A=2+16*(R-1)
680 SCREEN A,B:PRINT D$(K)
690 D(K)=R:P(Q)=P(Q)-1:P(R)=P(R)+1
700 IF P(2) <> 5 THEN 730
705 PRINT:SCREEN 12,13
710 PRINT "You have ended on the WRONG PEG!!"
720 FOR J = 1 TO 2000:NEXT:GOTO 740
730 IF P(3) <> 5 THEN 370
739 REM  ***  There at last!  ***
740 CLS:FOR J = 1 TO 5
750 SCREEN 12,8:PRINT SPC(35)
760 FOR K = 1 TO 500:NEXT
770 PRINT:SCREEN 12,8:PRINT"You have DONE IT!"
780 FOR K = 1 TO 1000:NEXT K:NEXT J
790 PRINT:PRINT TAB(12);"It took you";I;"moves."
"
800 IF I = 31 THEN 830
810 PRINT TAB(10);
820 PRINT "It CAN be done in 31 moves."
830 PRINT:INPUT "Another";A$
840 IF LEFT$(A$,1)="Y" THEN 130
850 END
949 REM  **  Heading  **
950 T$="*  The TOWERS of HANOI  *"
960 FOR I = 1 TO 25
970 POKE 3027+I,ASC(MID$(T$,I,1)):NEXT
980 RETURN
Ok
```

FOR SALE: TELETYPE MODEL 33 page printer (110 baud ASCII) with manuals as new. 90.00 o.n.o. David Warwick, tel. 0723 862160 (evenings).

FOR SALE - NASCOM 2, 32K RAM, Naspen, Toolkit, Nasdis, D-Bug, & Zeap, all in ROM. In case with PSU, all doc. and software.
Total cost 700.00+, selling for 399.00.
CENTRONICS QUICK PRINTER P1, as new with cable and software for N1.
With paper. Tel: WELWYN 4237 (Evenings).



80 BUS STATION

ROUTE

The Gemini MultiBoard concept is the logical route to virtually any microcomputer system you care to name. Whether you require a business system, an educational system, a process control system or any other system, there is a combination of MultiBoards to fulfil that function.

This concept ensures maximum flexibility and minimal obsolescence. Maintenance and expansion is greatly enhanced by the modular board design. MultiBoard is based on the 80-BUS structure, which is finding increasing acceptance among other British manufacturers; thus broadening the product base.

FARES

Hardware (Built & tested)

GM802	64K RAM card	£140
GM803	EPROM/ROM card	£65
GM807	3A PSU	£40
GM808K*	EPROM programmer	£29.50
GM809	FDC card	£125
GM810K	5A PSU/8 slot motherboard	£69.50
GM811	Z80 CPU card	£125
GM812 (*Kit)	Z80 IVC card	£140
GM813	Z80 CPU/64K RAM card	£225
EV814	IEEE 488 card	£140
GM815-1	Single drive disk unit with PSU (350K)	£325
GM815-2	Double drive disk unit with PSU (700K)	£550
QM816	Multi I/O board	£140
AM819	Speech board	£85
AM820	Light Pen	£35
GM821	ASCII keyboard	£57.50

Software

GM512	CP/M 2.2 for Multiboard	£90
GM517	Gem-Zap edit/asm tape	£45
GM518	Gem-Zap edit/asm disk	£45
GM519	Gem Pen editor/ text formatter tape	£45
GM520	Gem Pen editor/ text formatter EPROM	£45
GM521	Gem Pen editor/ text formatter disk	£45
GM524	Gem Dis disassembler/ debugger tape	£30
GM525	Gem Dis disassembler/ debugger disk	£30
GM526	Comal-80 tape	£100
GM527	Comal-80 disk	£100
GM528	APL disk	£200

LOGICAL ROUTE

GM 812 –IVC

The GM812 Intelligent Video Controller card features an on board Z80A processor to provide independence of the host processor and the ability to redefine the functions and parameters of the display.

Normally used in an 80 × 25 mode the card contains a programmable character generator allowing three additional modes of operation – inverse characters, 160 × 75 block graphics, or user defined characters.

A keyboard socket allows buffered character input, and a light pen socket is provided for specialist applications. Being I/O mapped the card does not occupy any system memory space.

GM 809 FDC

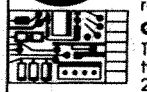
GM 815 DRIVE UNIT

GM 809 FDC

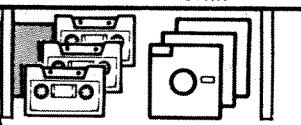
The GM809 floppy disk controller card can support up to four disk drives in either single or double density modes. The card uses the Western Digital 1797 controller and has variable write precompensation and phase locked loop data recovery circuitry.

GM 815 Drive unit

The GM815 floppy disk housing contains one or two 5¼" double density, double sided Pertec FD 250 drives. This gives a storage capacity of 350K per drive. Power for the drives is provided by an integral supply unit.

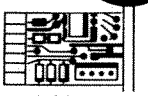


AUTO-EXCHANGE
All your RP/M software automatically transferred to CP/M



GM 802 RAM

The GM802 RAM board provides a full 64K of dynamic memory. The 80 BUS RAMDIS signal is fully supported so that any EPROM in the system is given priority over the RAM, preventing any possibility of bus contention. Page Mode is also supported by the card which, with the appropriate software, allows up to four memory boards to be used in a system.



FILL-UP WITH SOFTWARE

RP/M

RP/M software is available on tape and includes Editor/Assembler; Text Editor/Formatter; Disassembler/Debugger; Pascal and Comal-80. These packages can also be run under CP/M.

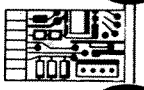
CP/M 2·2

A CP/M 2.2 package is available with the GM 809 card and Pertec drives. On-screen editing auto single/double density selection and parallel or serial printers are supported. Running under CP/M is a wide range of utilities, application software and languages.

ONE WAY

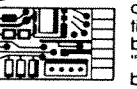
GM 803 EPROM BOARD

The GM803 Eprom Board will accept up to 16 2708 or 2716 Eprom devices. This allows the addition of up to 32K of firmware to the system. The board supports the Page Mode system and consequently need not occupy any memory space when not in use.



QM 816 I/O BOARD

The Quantum Micro I/O board provides a unique solution for interfacing to "the real world". The board contains 3 PIO's, a CTC and a real time clock with battery back up. "Daughter" boards may also be added and these include A-D, D-A, opto-coupling and serial interface boards.



A number of manufacturers are busy working on additional 80-BUS boards which will progressively increase the potential of your MultiBoard system.

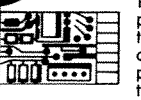
MEN AT WORK

PROTO-TYPING BOARDS

80 BUS compatible prototyping boards are available from both Vero and Winchester Technology. These allow the user to easily add a card of their own design to the system.

GM 808 EPROM PROGRAMMER

The GM808 Eprom programmer connects to the PIO on the CPU card and allows the user to program 2708 or 2716 type Eproms.



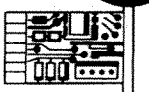
AM 819 SPEECH BOARD

AM 820 LIGHT PEN

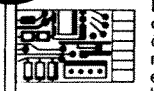
This low cost light pen can be used with the GM812 IVC for many applications, including answer selection, editing, menu selection and movement of displayed data blocks.

EV 814 IEEE 488

The EVC IEEE 488 Controller card has been designed to fully implement all IEEE 488 interface functions. This card gives the user a very versatile method of controlling any equipment fitted with a standard IEEE 488 or GPIB interface at minimal cost.



The Arfon Microelectronics speech board utilises the National Semiconductor Digitalker chip set. This gives a vocabulary of over 140 words and sub sounds. Output is from an on-board speaker.



GEMINI MULTIBOARDS – BUY THEM AT YOUR LOCAL MICROVALUE DEALER

All the products on these two pages are available while stocks last from the MicroValue dealers listed on right (Mail order enquiries should telephone for delivery dates and post and packing costs.) Access and Barclaycard welcome.

INTERFACE COMPONENTS LTD.
Oakfield Corner, Sycamore Road, Amersham, Bucks.
Tel: (02403) 22307. Tlx: 837788.

COMPUTER INTERFACING & EQUIPMENT LTD.,
The MICRO-SPARES Shop,
19 Roseburn Terrace,
Edinburgh EH12 5NG
Tel: (031) 337 5611

ELECTROVALUE LTD.
700 Burnage Lane, Burnage,
Manchester M19 1NA.
Tel: (061) 431 4866.

28 St Judes, Englefield Green,
Egham, Surrey TW20 0HB.
Tel: (0784) 33603. Tlx: 264475.

SKYTRONICS,
2 North Road, The Park,
Nottingham.
Tel: (0602) 45053/45215

TARGET ELECTRONICS
16 Cherry Lane, Bristol BS1 3NG.
Tel: (0272) 421196.

BITS & PC'S
4 Westgate, Wetherby,
W. Yorks.
Tel: (0937) 63774.

HENRY'S RADIO
404 Edgware Road, London W2.
Tel: (01) 402 6822.
Tlx: 262284 (quote ref: 1400).

LEEDS COMPUTER CENTRE,
62 The Balcony,
Merrion Centre, Leeds.
Tel: (0532) 458877

Nascom & Gemini USERS

NEW 32K C.M.O.S. BATTERY BACKED RAM BOARD

FEATURES:

EFFECTIVELY REPLACES EPROMS.
Does away with the inconvenience of **EPROM** programming and the compromise of assigning valuable address space to **ROM.**

ON BOARD RE-CHARGEABLE Ni-Cad BATTERY RETAINS MEMORY FOR OVER 1000 Hrs.
Battery is automatically charged during power-up periods.

HIGH SPEED OPERATION up to 6 MHz WITHOUT WAIT-STATES.

FULLY NASBUS[1] and GEMINI-80 BUS[2] COMPATIBLE.

PAGE MODE SCHEME SUPPORTED.
The board can be configured to provide one 32k byte page or two completely independent 16k byte pages.
Complete pages of 64k bytes are simply implemented by adding more boards on to the bus.

SOFTWARE and/or HARDWARE READ/WRITE PROTECTION.
4K blocks in either page are link selectable to be aligned on any 4K boundary.

FULLY BUFFERED ADDRESS, DATA AND CONTROL SIGNALS.

MEMORY I.C. SOCKETS ARE LINK SELECTABLE TO SUPPORT ANY 24 PIN 2k byte MEMORY I.C.s.
Thus the board can support up to 32k bytes of any mixture of cmos, nmos rams or 2716/2516 eproms.
All options are link selectable using wire links plugged into gold-plated socket pins, avoiding the risk of damage and the inconvenience caused by soldering links directly to the board.
The printed circuit board is manufactured to the high quality demanded by industrial users and conforms to B.S.9000.
The board comes assembled and tested and is supplied with a minimum of 2k bytes of low-power cmos ram. Fully documented.

AVAILABLE NOW!

PRICES:

Board with 2k bytes	**£80.00**
Board with 16k bytes	**£120.00**
Board with 32k bytes	**£170.00**
Bare Boards (circuit diagram supplied)	**£37.50**
HM6116-LP/3 Very low power 2k cmos memory I.C.	**£6.50**

Please add 95p P&P and VAT @ 15%

[1]nasbus is a trademark of nascom microcomputers a division of LUCAS LOGIC
[2]trademark of GEMINI MICROCOMPUTERS LIMITED

Processes cheques and P.O.s to:
41a MOOR LANE, CLITHEROE, LANCS.
BB7 1BE. For further information Phone (0200) 27890

Please supply me with the following:

Price

Total enclosed £

Address:

POST CODE

EDAC ENGINEERING

Two products for Nascom 2, to make your's a Super Nascom 2, and give your screen that professional appearance.

SCREEN FLASH ELIMINATOR Totally removes screen flash even under the most extreme circumstances of repetitive screen access. It is really great for games programs. Slows programs by less than 1%. ... Price £14.75

SCREEN WEAVE ELIMINATOR Corrects the frame frequency on the Nascom 2, and thereby removes the weave which is so noticable on some televisions and monitors. Price £8.75

SPECIAL OFFER !! both units purchased together for only £22.50

Also available SCREENPLUS for comprehensive reverse video and flashing words, all under program control. ..... Price £32.50

All three are NASCOM APPROVED PRODUCTS. They do not interfere with any Nascom or Nasbus facilities. They come assembled and tested, together with full instructions and circuit diagram, are easy to fit, and require no changes to your Nascom.

SAE for full details
EDAC ENGINEERING
257 Orphanage Road,
Erdington,
Birmingham, B24 OBD
Tel. 021-373-1260

nascom approved product

ATTENTION ATTENTION

NASCOM USERS

£99 £99

Ikon Computer Products pioneers in product development have broken the £100 barrier in data storage systems.

The HOBBIT CASSETTE system is available at only £99 plus VAT. This ready built, but unboxed, data storage system is based on a Phillips Mini Cassette with drive unit, interface board, firmware (in 2 2708's), manual, cassette with 100K storage, transfer speed of 750 bytes/sec The Hobbit is compatible with all Nascom Computers and Monitors.

IKON – Sophistication at a realistic price

£113.85 (incl. VAT) for the Hobbit Unit available from:
IKON Computer Products
Kiln Lake
Laugharne
Carmarthen
DYFED
Tel: (099 421) 515

LEVEL 9 QUALITY SOFTWARE for NASCOM 2 and NASCOM 1 (Nas-Sys/Cottis B.)

EXTENSION BASIC

This program enhances Nascom ROM BASIC, adding over 30 extra statements and yet using only 4K of memory. The new statements are used as if they were an integral part of ROM BASIC: it is as if you had bought a more powerful BASIC interpreter that was fully compatible with standard ROM BASIC. There is NO mucking about with USR calls.

Because Extension Basic is fully compatible with standard BASIC, you can use it either as an interpreter in its own right, or as a superb toolkit for producing programs to run under ROM BASIC alone.

The additional keywords include:

Editing Commands:
- AUTO — automatically number new lines;
- EDIT — edit long lines (up to 700 characters long);
- DELETE — delete a range of lines;
- REDUCE — discard comments and/or spaces to reduce size;
- RENUMBER — renumber lines and GOTO/GOSUB/ON/RESTORE references.

Debugging Commands:
- DEC & HEX — convert between decimal and hexadecimal numbers;
- FIND — find any text string (even if it includes keywords);
- SPEED — set keyboard auto-repeat speed;
- TRACE — follow program execution, optionally slowed down, showing line numbers and selected variables etc.
- XLIST & XREF — cross reference listing of selected line(s).

Input Statements:
- GET — get value of key pressed (0 if no key pressed);
- INKEY — wait for key to be pressed and return its value;
- INLIN — return entire line containing cursor, on ENTER;
- TEST — test whether selected key is up or down.

Screen Handling:
- COPY — copy block of memory (e.g to move the screen);
- LINE — set/reset/reverse a line between 2 points;
- PRINT @ — print at specified position;
- VDU — print string or string variable, even on top line;
- WRAP — prevent any word being split between 2 lines.

Structured Programming:
- IF..THEN..ELSE — do one statement if true, another if false;
- REPEAT..UNTIL — loop until condition true;
- WHILE..WEND — if condition true, loop while it remains true.

Extension Basic is supplied with a 25 page manual, which includes instructions on adding your own keywords to those provided, and a demonstration EB program which includes the games Sweeper and Demon Driver. The cassette version comes with a relocator program so that you can place EB anywhere in memory. EB works with Nas-Sys 1 or 3.

EXTENSION BASIC on cassette £15
2 * 2716 ROM (specify start address) £25
4 * 2708 ROM (specify start address) £25
to convert to ROM, having bought EB on cassette £12

Q-DOS

Q-DOS is a high performance file system for the Gemini G805 floppy disk drive. The 'Q' stands for Quick, for example saving a 32K program takes only 11 seconds, and loading it again takes 9.

Q-DOS provides facilities including:
- save BASIC or ZEAP programs or memory to named files;
- load files back into memory;
- display directory;
- erase, rename or protect files etc.

Also, Q-DOS allows machine code programs to read or write data to sequential files, with Q-DOS providing automatic buffering to optimise performance.

Q-DOS is supplied with complete documentation and utilities for:
- initialising disks;
- unerasing accidental deletions;
- coping with any hardware problem;
- saving all/selected files to tape in Nas-Sys format (you could load them individually using Nas-Sys R) and reloading a disk from tape.

Q-DOS uses 4K + 4K for buffers and D-DOS is not required. It works with either Nas-Sys 1 or 3.

Q-DOS on cassette £25
in ROM: 2*2716 or 4*2708
(specify start/buffer) ... £35

GAMES

Missile Defence 16K,MC,C £8
Shoot down enemy ICBMs, bombers and cruise missiles with ABMs from your 3 bases to protect these bases and the civilian population. This high-speed game is similar to the arcade game, and includes the unique option of a 2 player game: playing Ron and Leo you try to 'win' world war 3.

Fantasy 81 16K,B £8
An adventure game in which you must compete with the other 25 characters in a Gothic mansion to join the League of Heroes. The other 'players' behave realistically and you need cunning to win, rather than all-out aggression.

Space Invasion 16K,MC,C £7
The aliens are back armed with 6 types of missiles and assisted by commando raids. 10 speeds, 10 skill levels (easy - impossible) etc.

Bomber 8K,MC,C £5
Flatten your least favorite city to clear a landing strip for your bomber aircraft. This fast action game allows you to destroy London, Paris or New York.

Games Tape 8K,B & MC,C £6
5 games on a single cassette:
- Double Breakout. Break through 2 walls by knocking bricks out with a ball. 4-way bat movement and a choice of speed and bat. Winning earns a faster game for more score.
- Gunner. Shoot down enemies with angled missiles while avoiding the bombs they drop.
- Surround. Trap your opponent with your tracks. For 1 or 2 players.
- Wumpus. Hunt the loathly Wumpus in a maze. Then go after the others..
- Minefield. Cross a deadly mine-field using a dodgy detector.

DETAILS

Postage & Packaging is FREE, and there is no V.A.T to add.

Please send any order, or a large S.A.E for our ILLUSTRATED CATALOGUE to:

Level 9 Computing,
229, Hughenden Road,
High Wycombe,
Bucks. HP13 5PG

PROGRAMMER

Versatile programmer with own Nascom offers his skills to those with a software problem. Cheap rates if I like the job. Also willing to exchange software skills for hardware or hardware skills. Any hardware wizard in South London looking for a collaborator?
Contact: Mike York, 9 Rosehill Road, LONDON, SW18. Tel. 01-874-6244.

NASCOM 2 - 'TOUCH' TYPING TUTOR

Increases speed, eliminates errors. Two programs, include speed and accuracy tests. On N2 cassette (1200 baud). Needs 8K and graphics ROM.
Lloyds, 35 Magheraboy Road, Portrush, Co. Antrim, BT56 8NX. (0265) 823101.

DOCTOR DARK'S CP/M SMOOTHWARE SALE!

MARVIN.BAS is a program that enables you to have a conversation with your computer. Unlike many such programs, this one is capable of learning from your input! A way of making Marvin "forget" things he shouldn't know is also supplied. Written in EBASIC, the programs and data files can only be supplied on Gemini-CP/M compatible discs in Single Density format (YES, OK for G815's with 2 drives, as well as G805's with 1 or 2 drives)

And it's only £10

CHRIS BLACKMORE
31 HERNE RISE
ILMINSTER
SOMERSET
TA19 0HH

interface components

SOFTWARE FOR GEMINI DISK SYSTEMS from D. Parkinson, author of NasDis, Imprint and Gemini disk BIOSs.

LIST - A program to replace the CP/M TYPE command. The user may specify an ambiguous file name, and list the programs to the screen and/or printer. All output is paginated, with headings that may either be specified by the user, or will default to the file name. Lines may optionally be numbered. Listing may be interrupted and then skipped forwards or backwards by any number of pages. Alternatively, listing of any program may be aborted, in which case listing of the next file specified will commence.

REPAIR - Specifically for systems using the Gemini GM809 double density FDC card, this program allows the user to read and write sectors of a disk directly. It can be used to recover files that have been accidently erased, provided that they are still complete. It can also be used to recover source files from a disk on which the directory has been destroyed or corrupted.

COPYSB - For systems using the GM809 card and Pertec FD250 drives, this program will copy entire disks between Gemini double density format and Superbrain DD or QD formats. Transfer may take place in either direction, and the program will also format disks to Superbrain formats.

Prices: LIST & REPAIR on a Gemini DD/DS format disk - 25.00 + 1.00 p&p + VAT
COPYSB " " " " " " - 35.00 + 1.00 p&p + VAT

INTERFACE COMPONENTS LTD.
OAKFIELD CORNER, SYCAMORE ROAD, AMERSHAM, BUCKS HP6 6SU
TELEPHONE: 02403 22307. TELEX 837788

NASCOM & GEMINI OWNERS - SOFTWARE FROM CCSOFT - THE AUTHORS OF DEBUG.

We will be introducing three new packages during May, these are:

NAS-GRAPHPAC
Based on software that was originally developed for Nascom Microcomputers Ltd., this package links with the standard ROM BASIC to provide twenty additional commands for controlling the Nascom 2's low resolution graphics. This we feel will become the standard graphics package on the Nascom in the future as we will be supporting the new Lucas Logic high resolution colour card when it eventually arrives. Price £20, supplied on N2 cassette.

GEM-GRAPHIC
This relocatable package links with Microsoft's BASIC-80 and provides over twenty extra commands for controlling the Gemini GM812 IVC, making it a much more viable proposition on a disk based system. Price £35, supplied on disk.

GEMINI ECONOMY BASIC
Running under RP/M or CP/M on a system fitted with the Gemini GM812 IVC, this package was written with the cassette based Gemini system in mind, as it is less than 8K in length. The BASIC supports floating point numbers and also incorporates GEM-GRAPHPAC. Price £25, supplied on Gemini format cassette, or £28 on disk.

Include £1 p&p per order (no VAT yet) and specify exact system hardware please.
Send for Data sheets on the above software now!!!

Mail order to: CCSoft, 83 Longfield Street, London, SW18.
or see your local Nascom or Microvalue dealer.