SOUTH AUSTRALIAN

MICROPROCESSOR GROUP

P.O. BOX 113,
PLYMPTON, S.A. 5038
TEL. 278 7288

INC.

Meetings held at
THEBARTON HIGH SCHOOL
ASHLEY ST., THEBARTON

NEWSLETTER

Vol. 2. No. 2
June/July 1980 Newsletter. Postal Adr. P.O. BOX 113. PLYMPTON S.A.

EDITORS REMARKS

Boy do we have a lot of goodies for you this issue, as you will see we have a number of articles from Megs Micronews (MICROCOMPUTER ENTHUSIASTS GROUP SYDNEY). Our thanks must go to John Whitlock and Bill Bolton for allowing us to reprint their magazine articles in our publication. While on the subject of articles I have not received a very good response on entries from members. I am sure some of you have built projects or written software that would make good articles with little extra work to polish it up.

As the chairman has made comments on future meetings I will complete my remarks with a discovery made by Daniel R. Lunsford of Sacramento, California. He has discovered that the Z80 has some unpublished opcodes the official count being 694 with about 410 extra ones being added. Most involve the twin index regs and consist of things such as "load upper half of IX from any 8 bit reg", or "load lower half of IY immediate. Other instructions include things like "Shift left and set least significant bit.

The key to the new instructions can be seen from the fact that the object code for indexed instructions can be formed by taking the object code for an operation involving HL and adding 0FDH or 0DDH on at the front end (Almost). In fact try this with just H you will find it works ! The following examples use HX as high byte of IX reg and LX for low byte and similarly for IY.

```
DD 67    LD      HX,A

FD B5    OR      LY

FD 84    ADD     A,HY
```

There are some limitations; it is not possible to LD HX,LY. However you can LD HX,LX. The rule is that only one index reg at a time can be referenced.

On page 50 of the Z80 Technical Manual, in the Op-Code column, the object codes for the various rotate and shift instructions are defined in terms of prototypical configurations, with the various codes being differentiated by a 3 bit field. There are only 7 out of the possible 8, the missing one is pattern 6. When it is generated, the results show a "shift left and set least significant bit", the operand is shifted left and the low bit is set rather than reset. The high bit of the operand is shifted into the carry, and sign and parity are affected in the expected manner.

Eric Clarke. *** EDITOR ***

S.A.M.G. LECTURE PROGRAM FOR 1980/81.

A few changes have been made to the Lecture Program since our April/May issue so here is the ammended list.

In particular, notice that the next lecture is on Hardware Implimentation that has been somewhat lacking from our recent lectures. Bob Hourigan has had a great deal of experience with both big commercial systems and the smaller systems we are more familiar with. He should be able to give valuable advice to buyers and builders of uP equipment.

The July lecture will be our first exchange lecture with another group. John Willoughby from the Adelaide University Microcomputer Club will describe and demonstrate a portable Optical Spectrum Analyser that he has developed for the Geology Dept. based on an LSI-11. One of S.A.'s major microprocessor development areas is in Geo-survey equipment. This should have good general appeal.

The meetings continue to be held at 7.45pm on the second Friday of each month as indicated on the program below.

JUNE 13	Hardwire Implimentation. Lecturer: Bob Hourigan
JULY 11	Invited Lecture from Adelaide University Microcomputer Club. A Portable Optical Spectrum Analyser. Lecturer: John Willoughby.
AUGUST 8	The 6502 Microprocessor. Lecturer: Neville Diener.
SEPTEMBER 12	to be arranged.
OCTOBER 10	A Commercial Computing System. Arranged by Bob Hourigan.
NOVEMBER 14	Visit to Angle Park Computer Centre. Arranged by Bob Stunell.
DECEMBER 12	Programming in BASIC including file handling, sorting, games, etc. Lecturer: Ian Fisk.
JANUARY 9	Bring your own equipment night.
FEBRUARY 13	Annual General Meeting plus lecture to be arranged.

****

NORTH STAR COMPUTER SYSTEM

A user review by Bill Bolton.

I recently purchased a North star Horizon 1 computer. Here is a brief report on the operation of the system to date. The Horizon 1 is a Z80A based, S100 bus system and comes with a CPU board, 16K dynamic RAM board, double density disc controller board, 5 1/4 inch minifloppy disc drive, and a 12 slot motherboard with a single serial port on board. The motherboard has provision for a second serial port and parallel input and output ports, together with regulators for a second minifloppy. I'll be adding the components to implement these functions very soon. The serial I/O ports are implemented with 8251 chips and are thus programmable for a number of communications formats including synchronous systems. Similarly the parallel I/O is implemented with latches and has a DIP header to allow a number of different parallel communication conventions. The motherboard also has provision for a realtime clock in the interrupt selection.

The quality of the North Star hardware is really excellent. The mainframe is sturdy and comes with an attractive wood cover. The card cage takes up approximately half of the chassis while the disc drives and substantial power supply take up the other half. A fan is provided for forced ventilation, power comes in through a standard IEC 3pin connector and the mains and reset switched are on the back panel. Hardware documentation is also excellent with clear descriptions and circuit diagrams. One thick manual covers the motherboard, disc controller, power supply and chassis assembly, while separate manuals cover the CPU board, RAM board and disc drive.

The computer comes with Release 5 North Star software which comprises a good DOS, Extended Disc BASIC and a number of utilities including a monitor. All North Star software is designed to run from 2000H (hex) upwards which is a bit of a pain, however it is GOOD software. The DOS has a bootstrap ROM at E800H which loads it at 2000H-2CFFH. All other disc programs which will occupy more than 8K of memory have to load between 2D00 and E800, which is a 47K memory space and should be adequate for all but the longest programs. Memory space tends to become less significant in a disc based system as programs can be run in modules off disc if necessary. The DOS is probably the most widely used next to CP/M and there is a very great amount of software available for the North Star. It supports named files and all the other goodies one expects of a good DOS. A number of disc utility programs are supplied for disc testing and copying, including converting single density diskettes to double density format. The DOS can automatically detect single or double density format and can handle double sided drives when they become available.

The other major utility is a monitor. Versions are assembled for several different locations in memory from 0000H to F400H, including one version which has its own I/O routines so it can communicate without using DOS during memory testing. Most programs use the DOS I/O routines. The monitor supports a good range of commands including Fill Memory with

constant, Block Move (including overlapping moves), Memory Display, Memory Display and change, Jump to Memory location, Memory Test (a good one too!) and warm or cold DOS boot. Better still, the monitor command Jump table has space for up to 4 extra monitor commands so you can implement your own routines, and they give explicit instructions on how to implement those extra routines. I've already implemented a memory location routine in my monitor.

The disc contains two versions of North Star (Extended Disc) BASIC. One is self contained while the other is designed for use with the North Star Floating Point Hardware board (it operates faster and uses approximately 700 bytes less memory.) North Star BASIC is one of the most popular disc BASICs around and is well supported. The Release 5 version has some minor changes from the previous versions, including some minor redefinitions of reserved words. I can't comment more on it at present because I've only got 16K of RAM and that leaves just 43 bytes free when DOS and BASIC are loaded. I've ordered some extra memory from Thinker Toys (through Automation Statham- one of the "nice guys" in the computer shop field) [How did that plug get in?-ed.] and will comment further on NS BASIC once I've had a chance to use it. The message is plain-- don't buy a North Star system with only one 16K memory board.

The software documentation is excellent. One thick manual covers the lot, but don't expect much in the way of source listings. There is a good section on various patches to enable or disable various features of the software and a complete source listing of the I/O routines and initialisation routine but next to no source listings apart from that. Still I'm not complaining as they do give enough information for the dedicated hackers to find their way around when looking for specific pieces of code. The sections on getting the system up and running are particularly good. NS BASIC is fully explained, with a discussion of each aspect and function including a good index to quickly locate data on any key/reserved word.

I've only discovered one problem with the software and that was really a problem with the documentation. They give the wrong location for the DOS byte which sets the listing lenght to suit whatever video display is being used. I wanted to set this byte to suit the 16 lines of my terminal. this means that DOS directory listings continue for 15 lines, then the computer issues a prompt "PRESS RETURN TO CONTINUE" so that listings can be easily examined before scrolling off the screen. When RETURN is pressed another 15 lines of listing are displayed. I can't remember the exact fix just now but the byte is NOT at 2028H as specified in the documentation. It is a few bytes higher up in memory; it's not hard to find using a monitor memory dump. The people in the Melbourne Byte Shop (where I bought the system) helped me with the fix for that problem. Otherwise I've had no problems with the NS software that weren't of my own making.

To date the only software that I have written and run on the system is extensions to the monitor and some "tiny" languages such as SIMPLE, however the enormous power of the disc based system is very evident and I can hardly wait to get some more powerful software running. Double Density CP/M for North Star turned up last week but once again there is

not enough memory to support it as the Lifeboat Associates version comes configured as a 22K CP/M system (actually requiring 24K RAM to support North Star patches). A US dealer has advertised a patching program to interface North Star software to the Thinker Toys Discus 2D 8 inch floppy disc system and that is looking very tempting at the moment. There is no doubt that with both North Star DOS and CP/M DOS there is very little in the way of commercial software that I won't be able to run. The catch is that the interchangeable medium for CP/M is 8inch floppy and while I can support CP/M on 5 1/4 inch minifloppy a lot of the most desirable software is only available on the 8inch media.

The only other bit of S100 hardware that has arrived to date is a Mullens extender board. This board has a bit of active circuitry which will drive an onboard 7 segment display to display a H, L, or P when the probe is put on any of the S100 buss lines, thus indicating the state of that line. The P mode can be continuous or "pulse-catching" by switch selection. The board also has a screw link in each of the supply rails to allow current monitoring of the board on the extension. A nice set of stickon labels clearly identify each S100 bus line. All in all a very nice product (I got mine at Computerland).

Other hardware in my system includes an EME-2 video terminal which is my console device, a Philips 14 inch Pix monitor which has automatic switching to 525 or 625 line, an EME cassette interface, a 5X7 dot matrix 80 character impact printer and a Honeywell/Microswitch 103SD24 keyboard. I'll have more to say on some of these in my next report.

P.S. For all those of you who are envious of my Z80 CPU, consider the fact that none of my software to date came written in anything but 8080 code. There is very little Z80 coded software about, virtualy only the TDL/Xitan software plus a few assemblers and disassemblers. Hardly anything compared to the masses of 8080 coded software !!!!????!!!!.

Bill had two of his video monitors that he was displaying at the Home Computer Show stolen. If anyone has any information on two Sanyo VM9T video monitors serial numbers *1380112 & 1380135*, contact the Central Police Station immediately as the matter is in their hands.

CP/M--SOME NOTES.

by Bill Bolton

INTRODUCTION

These notes will look at the general principles involved in getting CP/M up and running together with a few hints and explanations on obscure features of CP/M which are not well covered in the documentation. While some general observations will be made on what CP/M is, this is not a full review of the software. A good review of CP/M can be found in Interface Age magazine's July and December 1978 issues.

WELL, WHAT IS CP/M?

Any disk based computer system need some "operating system" software to enable the user to access the disk system for program and data storage. This DOS (disk operating system) generally allows the user to create files (of data or programs), save files, delete files, load files, compact files on disk, test diskettes and whatever other disk based activities are required. CP/M a its very lowest level can be considered as a DOS, however it is much more than that. The standard CP/M as developed by Digital Research includes an assembler, dynamic debugger, text editor, peripheral interface transfer, hex dumper and loader, system regenerator and mover and a batch command processor, together with a few other utilities. It is much more of a program development environment than anything else. It isn't a language, so you don't write programs in CP/M but you do write programs in a computer language (assembler or higher level) to run USING CP/M to communicate with the disk system and system communication peripherals.

THE HARDWARE ENVIRONMENT

CP/M is written in Intel 8080 code and requires an 8080 family processor (i.e. an 8080, 8085, Z80) to operate. The software of the resident portion of CP/M (the part which is "booted" in at startup (and is needed to load programs) is divided into 3 parts, only one of which is system hardware dependant. This part is called the BIOS (Basic I/O System). Once a BIOS has been written to suit a particular 8080 family computer hardware environment (i.e. a North Star Horizon or a Processor Technology Sol or ... etc.) then any program developed to run under CP/M should be capable of operation in that computer provided that sufficient memory, the correct type of I/O peripheral and the correct media is available in that system. Unfortunately not all CP/M programs are fully transportable to all CP/M systems, but that is generally the fault of the program author, not the result of restrictions within CP/M itself. If you have a reasonable amount of memory (32K or more), a serial terminal (CRT or Teletype etc) and at least one 8 inch floppy disk drive you should be able to run most CP/M programs. You can run CP/M on minifloppies (I do) as well but a lot of commercially available software running under CP/M is only available on 8 inch diskette. For 8080 family computers it is the closest thing available to a "software buss".

There is one other aspect of this software which can be considered to be hardware dependent. Whilst it is possible to run a single disk CP/M system it is not the best way of doing things. CP/M is a very strongly disk oriented piece of software and supports up to 4 disk drives. A lotof the functions available work most efficiently on a disk to disk basis, so a minimum of two disk drives is necessary if you don't want to be eternally swapping disks in a single drive (as I am doing at the moment).

FDOS

The FDOS is the software section which interfaces CP/M to your particular system hardware. Digital Research wrote CP/M to exist in an Intel MDS hardware environment but there are now versions of CP/M available to support all of the popular disk systems (ie North Star, Micropolis, Discus, Tarbell, Versafloppy, etc) together with many of the popular I/O boards. Generally this means that you can buy the software with the BIOS section already configured for your disk system, however you may still have to write a BIOS user area to suit your I/O configuration. (Later I'll tell you how one CP/M supplier overcomes this.) Double density versions are now becoming available too, with their greatly increased storage capacity.

BIOS

The BIOS user area may only contain a simple routine which interfaces CP/M to a single console terminal or it can be complex software to handle up to 16 distinct I/O devices selectable under program control. the complexity of the BIOS is dependent on the I/O needs of the user.CP/M supports an "I/O Byte" which maps the 16 possible "physical" I/O devices to four "logicAl" I/O devices. These four logical devices can all access a single real I/O device if that is all your system supports.

A REAL CP/M SYSTEM

Now let's get down to the real nitty-gritty:-- how I got my CP/M up and running. I purchased my North Star Double Density CP/M from Melbourne's Byte Shop. It is a Lifeboat Associates Version 1.44 on a 5.25 inch Verbatim floppy diskette with the standard Digital Research documentation plus some extra notes on the North Star implementation. The specific software difference in this implementation was a requirement for 2K more system memory than the size of the CP/M system to be supported. The system as distributed needs 24K of memory which supports a 22K CP/M system in standard CP/M definition. The extra 2K of memory space appears to be taken up in the FDOS for the North Star Micro Disk System.

The first thing I did before trying to get CP/M running was to copy the entire diskette to a new diskette using the North Star DOS read and write commands. This meant that if I made a fatal mistake while activating CP/M I could simply make another copy of the original diskette and try again. After I'd got CP/M running I recopied the original diskette using the DDT program just to make sure I had good copies of all the programs on the original disk.

The additional notes on the North Star implementation gave comprehensive directions for bringing the system up using both North Star DOS and using a more generalised procedure not so dependent on the existing DOS. As I had the North Star DOS already running I elected to use that method. The distribution diskette had a North Star DOS file named CPM in place of the normal CP/M boot loader, so it was simply a matter of booting North Star DOS and then typing the DOS command GO CPM to bring a limited CP/M system into operation using the I/O routines resident in the N.S.DOS. As these routines have to be present for the N.S.DOS to function, this method guarantees that you have a limited CP/M system you can communicate with. CP/M loads from 0000H upwards while the N.S.DOS resides at 2000-2D00H, so the N.S.DOS is in the middle of the CP/M program area, which means that only small CP/M programs can be run -however the only reason for running this limited version is to make it easy to do the patching necessary to implement a full function CP/M system which is independent from the N.S.DOS!

Lifeboat Associates provide a number of extra programs over and above the normal Digital Research ones on their distribution diskette. These extra programs are mostly to make using CP/M in a North Star environment easy to implement and use. Two programs are specifically designed to make the BIOS configuration very easy.

The first program, called CONFIG, contains several standard BIOS configurations for specific computers such as North Star Horizon, Processor Technology SOL, Vector graphic, Imsai, Xitan etc. These configurations are all identified by a number, which, when inserted in a particular location in CONFIG will cause CONFIG to automatically insert the selected I/O software in the BIOS user area of the CP/M system in memory.

The second program is called SAVEUSER. It writes the BIOS USER AREA onto the CP/M system tracks on the diskette.

At this stage I had CP/M in memory communicating through N.S.DOS. I was able to use the CP/M DDT (Dynamic Debugging Tool) program to load CONFIG and insert the North Star Horizon I/O configuration number into location 120H, then Jump to the start of CONFIG. A message appeared on the VDU screen telling me that my system was configured and to use SAVEUSER to permanently save it on disk. I then ran the SAVEUSER program which gave directions on what to do. Once SAVEUSER put the Horizon BIOS USER AREA onto disk I did a cold boot to bring a completely fresh CP/M system into memory from disk as a test. The system came up without any problems and now I had a full CP/M system running.

NON-STANDARD CONFIGURATIONS

Configuring the Lifeboat North Star system for an I/O other than contained in the CONFIG program is more difficult and really needs at least a system monitor program to patch your I/O program into CONFIG, starting at 200H. Once you've got your I/O program into CONFIG then the method to get it into the system in memory and then onto disk is the same as previously described. Your I/O program need initially only be a basic console I/O only as once you've got that installed and running in CP/M you can use the full power of the CP/M assembler, text editor and

debugger to write your fully implemented I/O routines. The Lifeboat documentation covers this very well and provides assembly listings of several BIOS USER AREA programs as a guide to writing your own.

THE OTHER LIFEBOAT "SPECIAL" PROGRAMS

As well as CONFIG and SAVEUSER Lifeboat Associates also provide the following "extra" programs on the distribution diskette.--
LIST: gives a directory listing with record lengths in a more convenient and compact display than the normal DIR command.
FORMAT: will format a blank disk to either single or double density.
DENSITY: software selects a particular drive for either single or double density.
COPY: will do a complete disk to disk copy, or copy system, files, etc.

While the above applies specifically to the North Star D.D. version, I gather that most of the Lifeboat Associates implementations provide similar facilities.

SOME NOTES ON THE CP/M TEXT EDITOR.

The documentation accompanying CP/M is excellent but there are a few oversights and omissions. The documentation on the ED text editor mentions the nZ command in a command table only and gives no real details. The Z command is used to halt a type out of the edit buffer every 24 lines for a brief period. A value of 10Z gives a delay of approximately one second. If you have a VDU with a 16line display rather than a 24 line display you can set up a macro command to give a pause every 16 lines.. the form is nM16T16L10Z.. this will give a 17 second pause every 16 lines and will repeat the macro n times.

Appendix A to the Ed documentation mentions the V command. It is well worthwhile reading this section of the documentation carefully and learning how to use this command as I feel it is the real key to making ED a useful tool. The V command puts line numbers on the text ONLY while it is in the edit buffer, --the line numbers are not put on the disk file. In V mode a command such as 10:10T will print ten lines from line 10 onwards. A command such as 10:5K will erase five lines from line 10 onwards and renumber all following lines. A command such as 45:FSTRING^z will search line 45 onwards for the first occurranceof the string "STRING" and place the command pointer after it. In other words the line number can be used as an absolute address to perform most ED commands. A -V command will switch off the line numbering.

CONCLUSION

I've been using CP/M for a couple of months now and its hard to remember how programming was possible without it. It is a powerful software development tool in itself but even more importantly it is the basis for supporting a lot of ther powerful software. All 8080, 8085, and Z80 users should have a good look at CP/M, I believe it is very good value for money.

"BABY" 2650 GROWS UP.

by Ray Biddle.

In common with many others I purchased a "Baby" Signetics 2650 kit and two hours later "CR", "LF" and * appeared on my display. The 256 bytes of memory and their limitations soon became apparent.

On the horizon appeared the new "Expandable" 2650 Mini-computer system. In comparing the circuits of both, the following differences were noted: 2114 memory chips instead of 2112's, an extra decoding IC (74LS138) and a few different connections, plus the provision for expansion up to 5K memory (1K ROM and 4K RAM).

So out came my trusty pocketknife. I warmed up the soldering iron, located a small decimal fraction of a meter of rainbow cable (namely 8" long) from a source known to all, a 74LS138, and 8 2114 RAMs, plus a few square inches of perforated Veroboard.

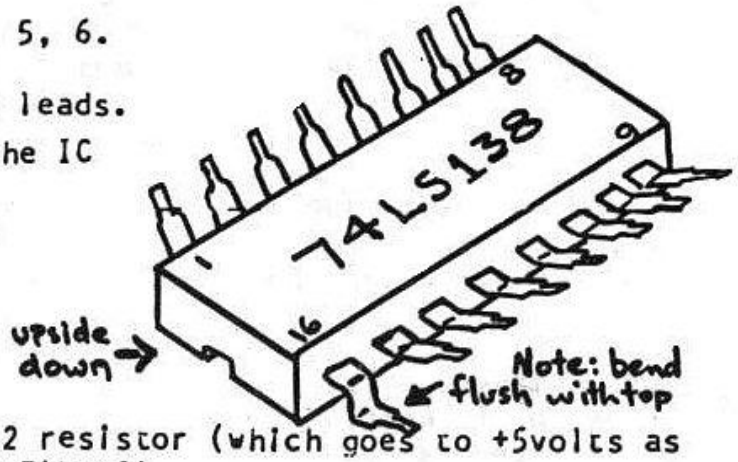
OPERATION PROCEDURE: Remove all chips, having marked some pin numbers on the back of the circuit board;- 2650 mark 1,20,21,40; 2608 mark 1,12; 74LS38 mark 1,7. NOTE that the pins are reversed on the back of the board!

Now take a penknife and cut the following tracks, as close to the pins as possible remembering that you will be using the pins again.

2650 cut track pin 4 on both sides.
2608 cut tracks 10 and 11.
74LS38 cut tracks 1, 2, 3, 4, 5, 6.

Now take 74LS138 and bend the pins OUTWARDS for easy attachment of the leads.

Bend pin 16 level with the top of the IC and solder it to TRIANGULAR + on Baby board oriented same as 74LS38.



WIRING UP.

74LS38

Pin 1 to pin 22 on 2650
pin 2 to pin 20 on 2650
pin 3 to pin 4 on 74LS138 to 1K2 resistor (which goes to +5volts as pullup for open collector output of 74LS38)
pins 4, 5, and 6 have no connection.

74LS138

pin 1 to pin 4 on 2650.
pin 2 to pin 3 on 2650.
pin 3 to pin 2 on 2650
pin 4 already done.
pin 5 to earth.
pin 6 to pin 24 on 2650.
pin 7 no connection.
pin 8 to earth.

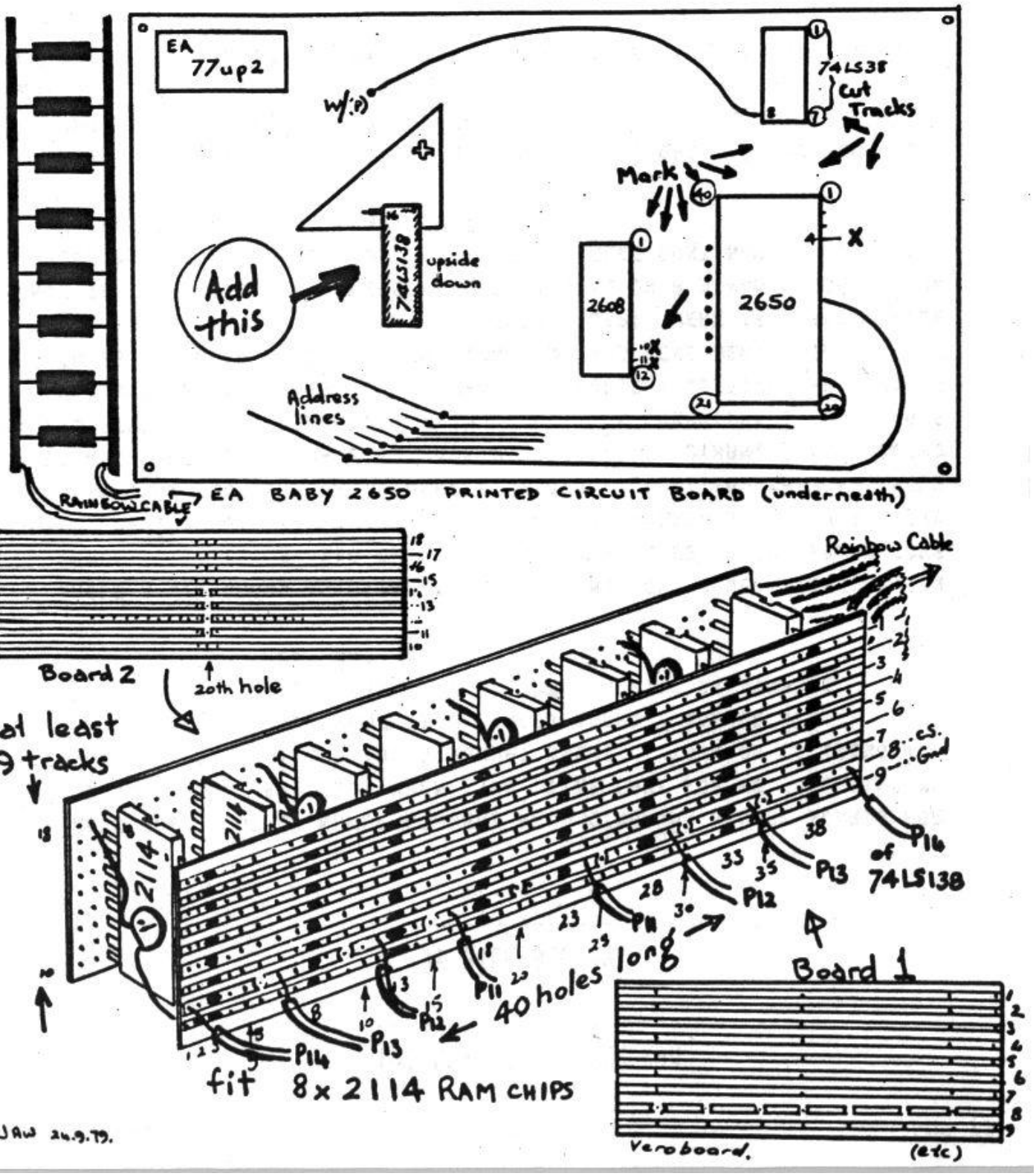
2608

pin 11 to pin 15 on 74LS138

2650 PINS

pins 5 to 14 for memory 0 to 9
pins 33 to 30 for data 0 to 3 & 29 to 26 for data 4 to 7.

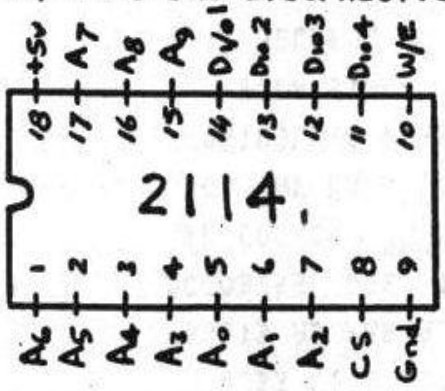
OTHER MEMORY CONTROLS
pin 8 of 74LS38 is W(P) line.
From 74LS138 pin 14 is 0400 to 07FF
pin 13 is 0800 to 0BFF
pin 12 is 0C00 to 0FFF
pin 11 is 1000 to 13FF.

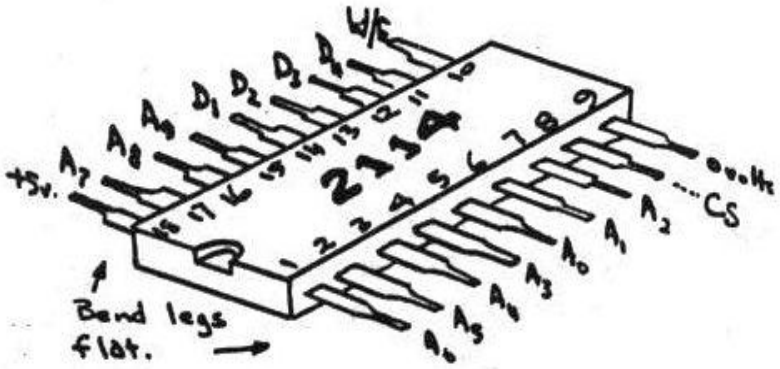


DESCRIPTION OF MEMORY BOARDS.

BOARD 1 is 9 tracks wide by 40 holes long. Track 8 is cut at holes 5, 10, 15, 20, 25, 30, and 35. WIRING:track 1 is A6, 2 is A5, 3 is A4, 4 is A3, 5 is A0, 6 is A1, 7 is A2. Track 8 holes 1 and 40 is 74LS138 pin 14, 6 and 34 is pin 13, 11 and 29 is pin 12, 16 and 24 is pin 11. These are the chip selects. Track 9 is ground.

BOARD 2 is also 9 tracks wide by 40 holes long. Tracks 5, 6, 7, and 8 are broken at hole 20. WIRING: Track 1 is +5 volts, 2 is A7, 3 is A8, 4 is A9, 5 hole 1 is 33 on 2650, 6 hole 1 is 32 on 2650, 7 hole 1 is 31 on 2650, 8 hole 1 is 30 on 2650, 5 hole 21 or 40 is 26 on 2650, 6 hole 21 or 40 is 27 on 2650, 7 hole 21 or 40 is 28 on 2650, 8 hole 21 or 40 is 29 on 2650, 9 w(p) (WE). NOTE that holes 40 may be more convenient than holes 21, thus the alternative.





Flatten the 2114 chips as you did for the 74LS138 (see diagram). The 2114 chips are placed in holes 3, 8, 13, 18, 23, 28, 33, and 38. Recheck their orientation. Add a few 0.1 bypass capacitors from +5 volts rail to earth rail near each RAM.

The memory of 4K is made up of two pieces of Veroboard with some tracks broken as detailed. The boards are aligned in parallel with the copper tracks outside, and the memory chips between them soldered to the copper strips.

Tin all tracks and place earthed short across ends of tracks while soldering. Earth soldering iron, etc, taking all the usual precautions for handling static sensitive MOS memory devices. If unfamiliar with this ask someone at a Club meeting. Solder all 2114 memories, pin 9 (ground) first to track 9 Board 1 and then pin 18 to track 1 on board 2. Proceed to solder rest of pins, using a low heat (current recommendation is circa 320 degrees CF) and a good resincore solder. Remove one short and attach to Baby 2650 board, with rainbow cable, then remove another and attach, and so on.

Replace 2650, 2608, 74LS38 and 74123 in Baby board, BUT NOT the 2112 memories.

Arrange the memory board so that air can circulate through it.

Good luck and bon computing!

THINKER TOYS SUPER RAM

BY BILL BOLTON

THERE ARE A LOT OF MANUFACTURERS PRODUCING RAM MEMORY BOARDS FOR S-100 BUSS COMPUTERS; SO WHEN THE TIME CAME FOR ME TO EXTEND MY SYSTEM MEMORY TO 48K I HAD THE SOMEWHAT DAUNTING TASK OF DECIDING ;

A. WHAT SIZE MEMORY BOARD - 16K OR 32K
B. WHAT TYPE OF RAM - STATIC OR DYNAMIC
C. WHAT BRAND

WHILE A 32K BOARD IS ATTRACTIVE IN TERMS OF POWER CONSUPMTION AND SPACE; IT IS A BIG CHUNK OF MEMORY TO LOSE IF THE BOARD GOES FAULTY. AS I'M NOT LIKELY TO RUN OUT OF MOTHERBOARD SLOTS IN A HURRY I DECIDED THAT TWO 16K BOARDS WERE A BETTER PROPOSITION THAN A SINGLE 32K BOARD. ALTHOUGH I HAVE A NORTH STAR 16K DYNAMIC RAM BOARD WORKING PERFECTLY SATISFACTORILY IN MY HORIZON COMPUTER; I STILL HAVE AN INSTINCTIVE MISTRUST OF DYNAMIC MEMORIES; DESPITE THE OBVIOUS ADVANTAGES OF LOWER POWER CONSUMPTION AND THUS LESS HEAT OUTPUT. IF A FAULT DEVELOPS ON A STATIC MEMORY BOARD THERE IS A REASONABLE CHANCE OF ISOLATING IT; HOWEVER FAULT FINDING ON A DYNAMIC BOARD CAN BE MUCH MORE COMPLEX DUE TO THE REFRESH CYCLES CONTINUALLY OCCURING. SO THE CHOICE FOR ME WAS STATIC RAM.

THE CHOICE OF BRAND WAS THE MOST DIFFICULT ONE. MY SYSTEM RUNS AT 4MHZ SO I NEEDED A BOARD WITH 250 NS ACCESS TIME FOR GUARANTEED OPERATION AT FULL CPU SPEED. PRICE WAS ALSO AN IMPORTANT CONSIDERATION; PRICES VARY WIDELY FOR 16K BOARDS; I DIDNT WANT TO SACRIFICE QUALITY FOR PRICE.

I FINALLY DECIDED ON THE THINKER TOYS 16K SUPER RAM BOARD. THE BOARD IS DOUBLE SIDED; PLATED THROUGH; SOLDER MASKED AND COMPONENT OVERLAYED. SOCKETS ARE PROVIDED FOR ALL ICS AND THERE ARE A MINIMUM OF NON-IC COMPONENTS ON THE BOARD. THE DOCUMENTATION IS VERY GOOD WITH FULL STEP BY STEP ASSEMBLY INSTRUCTIONS; A MEMORY TEST PROGRAM AND A GLOSSY PHOTOGRAPH TO SHOW YOU WHAT THE FINISHED BOARD SHOULD LOOK LIKE.

THE 16K MEMORY IS ARRANGED AS 4 x 4K BLOCKS; EACH ADDRESSABLE AT ANY 4K MEMORY BOUNDARY. MEMORY ADDRESS DECODING IS DONE BY A PROM AND ADDRESSING IS SET BY DIP SWITCHES AT THE TOP OF THE BOARD. IT IS POSSIBLE TO MAKE TWO OR MORE 4K BLOCKS DISAPPEAR AS FAR AS THE SYSTEM IS CONCERNED BY SELECTING THE SAME ADDRESS FOR THOSE BLOCKS. EACH 4K BLOCK IS ALSO WRITE PROTECTABLE; AGAIN DIP SWITCHS ARE SELECTABLE AT THE TOP OF THE BOARD. SWITCHING IS ALSO PROVIDED TO COPE WITH THOSE S-100 COMPUTERS WHICH DON'T GENERATE MWRITE ON THE CPU BOARD; (I.E. ALTAIR AND IMSAI). OVERALL THE BOARD IS EXCELLENT BOTH IN TERMS OF COMPONENT QUALITY AND DOCUMENTATION.

I'D LIKE TO SAY THAT THE TWO BOARDS THAT I PUT TOGETHER WORKED FIRST TIME BUT UNFORTUNATELY THAT WASN'T THE CASE. WHEN I PUT THE BOARDS INTO MY HORIZON COMPUTER; THE SYSTEM PROMPTLY CRASHED ! TO CUT A LONG STORY SHORT THIS TURNED OUT TO BE A LATENT DESIGN FAULT IN THE HORIZON RATHER THAN A SUPER RAM DESIGN OR ASSEMBLY FAULT. THE NEW NORTH STAR DOUBLE DENSITY DISK CONTROLLER BOARD TRIES TO DRIVE THE S-100 PHANTOM BUSS LINE WITH AN OPEN COLLECTOR TTL GATE; WITH UNPREDICTABLE RESULTS AS FAR AS LOGIC LEVELS GO. THE SUPER RAM BOARD PROM ADDRESS DECODERS USE THE PHANTOM LINE AS PART OF THE ADDRESS DECODING TO ENSURE CORRECT PHANTOM OPERATION AND THEY DIDN'T LIKE THE UNDEFINED LEVELS ON THE PHANTOM LINE. A PULL UP RESISTOR ON THE NORTH STAR DISK CONTROLLER BOARD FIXED EVERYTHING AND THE BOARDS HAVE WORKED PERFECTLY EVER SINCE ! ROD WHITWORTH OF AUTOMATION STATHAM; BANKSTOWN; WAS VERY HELPFUL AND EVEN CALLED THINKER TOYS IN THE U.S. TO GET THE SOLUTION TO THE PROBLEM.

I AM VERY HAPPY WITH THE THINKER TOYS 16K SUPER RAM MEMORY BOARDS. NOT ONLY ARE THEY EXCELLENT QUALITY BUT THEY ARE ALSO ABOUT THE CHEAPEST S-100 16K RAM BOARD AVAILABLE AT THE MOMENT. WHAT IS MORE I KNOW THAT THE LOCAL AGENT; AUTOMATION STATHAM; DOESN'T LOSE INTEREST IN THEIR CUSTOMERS AFTER THEY HAVE MADE THE SALE AND ARE PREPARED TO OFFER REAL AFTER SALES SERVICE; SOMETHING ALL TOO RARE IN THE MICROCOMPUTER INDUSTRY !!!!!

THIS ARTICLE WAS PREPARED USING THE CP/M CONTEXT EDITOR PROGRAM ED.COM; FORMAT PROCESSED WITH THE CROMEMCO PROGRAM FORMAT.COM AND PRINTED ON AN EXTEL AF-11R DOT MATRIX IMPACT PRINTER.

MICROCOMPUTERS--1979 AND THE EIGHTIES

Notes of a talk given to the August meeting of MEGS by Lionel Stewart.

Once upon a time there were three different types of computers.

There were microcomputers, defined as 8-bit computers built on a chip.
There were minicomputers, defined as 16-bit computers usually requiring a board or two to contain all the computer logic.
And there were the 32, 48 or 64-bit architecture computers generally filling a large cabinet.

Today the distinction is a little confused but will once again become clear in a couple of years when we will have:

MICROCOMPUTERS which will be distinguished by the fact that they consist of one microprocessor.
MINICOMPUTERS which will be identified by the fact that they are made up of two or more microprocessors, and
MAINFRAME COMPUTERS which will be two or more 32-bit microprocessors.

In other words that humble little microprocessor chip that many of us like to take into our garages and poke and prod and try to get games running on. That little $35 piece of silicon is the forerunner of the future business and scientific computers.

The only thing that puzzles me is how mainframe suppliers will be able to justify a cost of hundreds of thousands of dollars for a computer consisting of maybe six microprocessors each costing about $100. It is probably no coincidence that they are starting to charge separately for software. We could reach the stage where the operating system and the software costs a couple of hundred thousand dollars and the CPU is thrown in free.

But I am getting ahead of myself.

I have spent the last year looking into the use of microcomputers currently in use in business application and have come up with some interesting observations.

One application is in cash registers and the related devices where you sometimes run into the situation where the salesman says to you- "here is our lovely machine and you can have it for $9000. Give us a day or two and we will alter the program in it to meet your needs." When pushed he will finally reply "OK, I give in! You can have it for $4000. The CPU is an Intel 8080 and you can change the program yourself."

If you can bundle the product, that is, include programs and hardware in the price and call it an intelligent device then you can double, triple or quadruple the price.

The most interesting device I came across was a cash register with a CPU and 250K of memory which sold for $8000. To my way of thinking that is not a bad way to buy a quarter megabyte computer. Just strip off the cash

register and rehouse it in a computer-like cabinet and you have a good buy.

Word processors are another sample of bundled software hardware. Separately you can buy a TRS80, a Selectric typewriter and a word processing program for say $4000. If you buy a similar configuration repackaged as a word processor you pay $8000 - $12000. A word processor is nothing special. It is simply a microCPU with high quality printer and a program.

I would like now to look at the hardware side of development that has taken place in the last few years and tie together various developments that have been and then project the results forward.

In 1977 computer costs were relatively high. For example a typical mainframe computer (IBM 145) cost $1 million for the CPU alone. Today, two years later, you can purchase a more powerful CPU (the 4341) for $200,000. The cost of memory has dropped from $250,000 (1976) to $80,000 (late 1977) to $12,500 now for one megabyte. Prices have gone through the floor in two years.

On the other hand we have recently heard announcements that switching times have been reduced to 10 picoseconds (ie. 10 one millionths of a millionth of a second. The speed is so great that a propagation delay has been introduced, that is, at these speeds electricity travelling at the speed of light takes 15 picoseconds to move from one component to another.

In my experience it takes 5 years from the announcement of a discovery to implementation and 5 years from now is 1984 which is very convenient because 1984 is a popular year for predictions. "SUPERFAST MICROS IN 1984!"?

On the other hand a survey taken in America found that 256K memory chips are planned for next year and that one manufacturer is bypassing the 256K chips and aiming to release 1Mb chips in 1982.

Now when I consider the prospect of superfast low cost micro computers together with super low cost memory and the effect this will have on man must be farreaching.

Those incredible developments seen in the film "Now the Chips Are Down" (seen at MEGS recently) are really only the tip of the iceberg. They had to be cost justified, or almost cost justified at least, and they were implemented on computers and electronics which was five times as costly as today and many many times more costly than in the near future.

So then, we have today 8 and 16 bit microcomputers which benchmarking has shown, perform as well as the present day minicomputers and outperform some of the mainframe CPU's of the previous decade.

If we take these microprocessors and speed them up from 200nanoseconds to 200 picoseconds (that is 1000 times faster) then we have a single chip CPU that will outperform practically all modern day computers.

The high speed chips will cost say $100 rather than today's price of $35 however this beats the $1 million price tag on the more powerful CPU's available in 1977.

I have made statements about the power of microprocessors.For example I have said in effect the little TRS80 toy or the Apple or whatever are driven by CPUs that outperform some minicomputers. The demonstrations [which followed this talk - Editor] will justify my statements [they did - Ed.]

[At the same time the effects of programming languages on the subject were discussed. An effective demonstration using the TRS80 showed the vast increase in speed of movement of a "spaceship" in a TV game program achieved by using some machine code sub-routines within the BASIC language program. The differences between interpreters, compilers and compiler/interpreters was described. - Ed]

BACKUP is vital for computer users. Stop every 15 minutes or so and ensure that you have duplicated your work. Copy your updated software and make sure that if the power failed or your media erased you would have only a limited disaster.

DISTRIBUTED PROCESSING BY DEFAULT

Bill Bolton

Recently I've acquired a number a new peripherals for my trusty S-100 system. Now I've always regarded the Z-80 in my North Star as "The Processor" but lately I've come to realise that it isn't that simple! I was aware many new peripherals had processors in them performing dedicated tasks but hadn't realised that the processing power in them can be a significant part of total system capabilities !

Here is the current count of processors in my system:

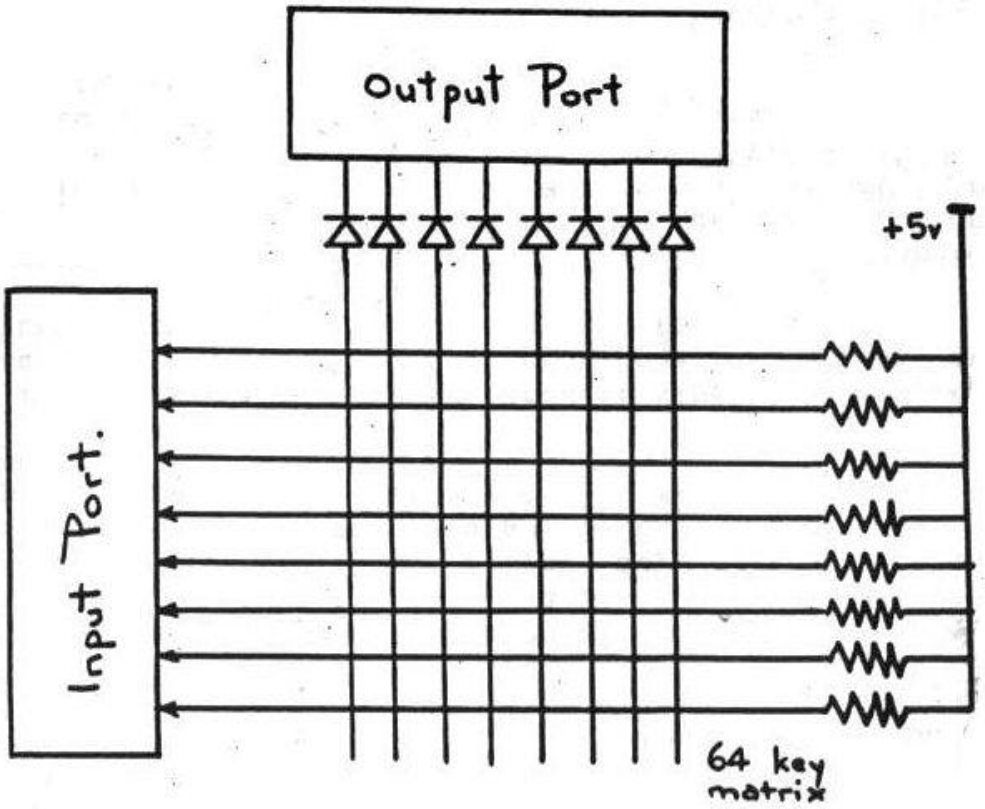
1 x Z-80	in the "Computer"
2 x 8080	in a daisywheel printer
1 x 8080	in a programable interface
1 x 8741	in a keyboard
1 x 8008	in a dot matrix printer
1 x 6800	in a video terminal
1 x F8	in a high speed printer

That makes a total of eight processors not to mention a CRT controller and Disk controller, both of which approach the complexity of a processor chip. It seems that with the falling cost of processors and memory that more and more peripherals will include at least one processor chip. Sooner than we think it may be impossible to point at one box and say that "there is my computer"!!!!!!

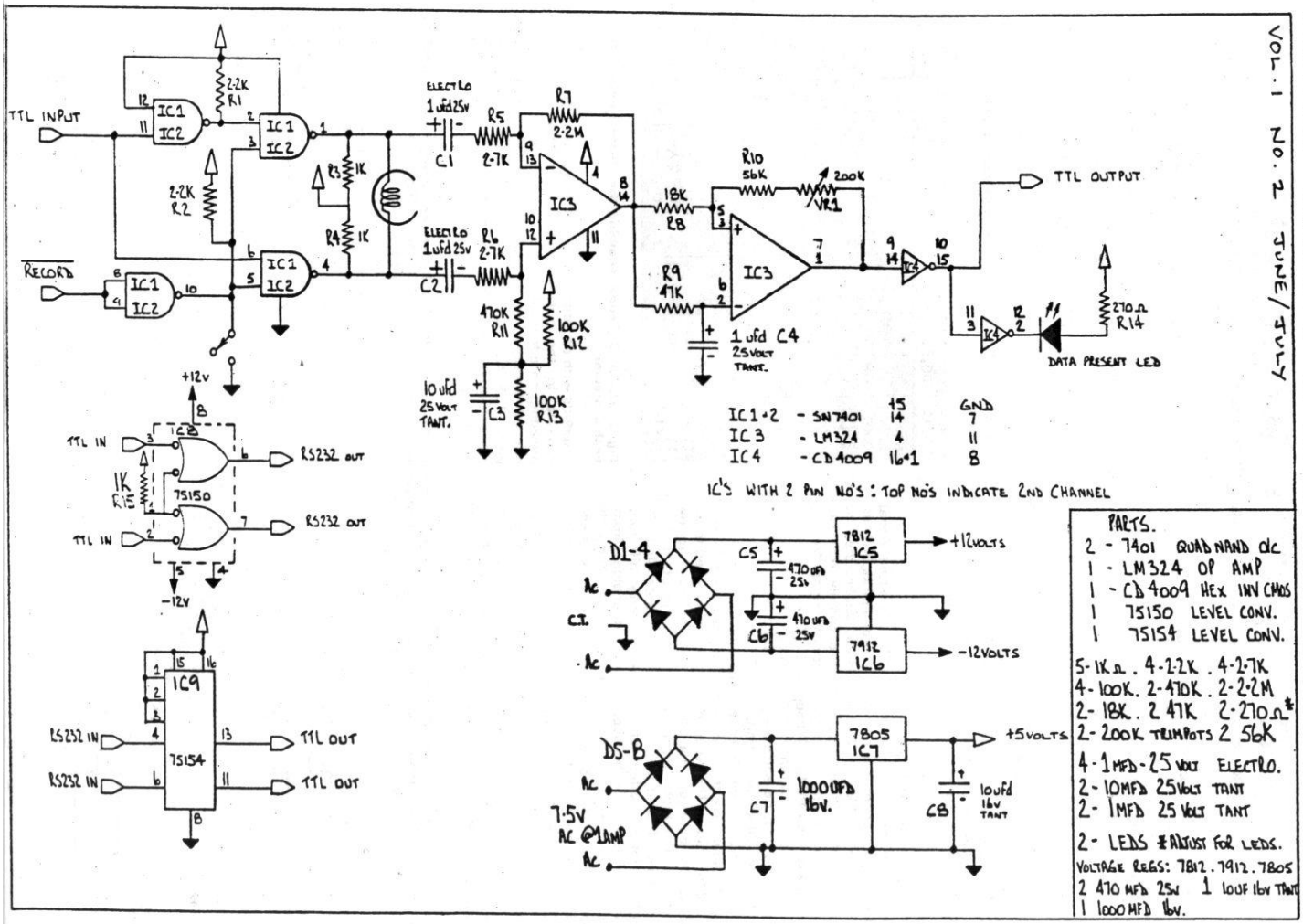
ADD A SOFTWARE-DRIVEN KEYBOARD TO YOUR 6502 SYSTEM.

The main features of a software driven keyboard are as follows:
(a) Should use a minimum of hardware and incorporate the required functions into software.
(b) While waiting for input from a keyboard, most microprocessors are idle. A software controlled keyboard therefore is much more efficient from a system utilisation aspect.
(c) Dynamic key debounce can be used. There is no fixed debounce period (eg. 5 ms). The software should continuously monitor the hash on key open and close and adjust the debounce period required accordingly. A variety of keyboards can therefore be used without giving trouble and in addition the system will accomodate deterioration of key contacts with age.
(d) Since upper and lower case characters are stored in memory, allocation of characters to keys is at the discretion of the user. Control characters may also be stored in memory if required.
(e) Special keys can be used -- eg to clear VDU screen or turn tape recorder on or off.
(f) Since the last character sent is stored in RAM the repeat function can be implemented by using a single key. Juggling up to three fingers in the right sequence is eliminated.
(g) 2-key rollover, ie the ability to press the next key before releasing the last, can be easily achieved in the software design.
(h) Programs can check for individual key closures and check response times, duration etc. This is useful in realtime applications such as games.
(i) An "upper case" key can be used to provide upper case only.

On the hardware side all that is needed is a keyboard with singlepole single throw contacts and a PIA (peripheral interface adaptor chip). The popular 63 key Datanetics keyboard is ideal since it has good quality keys, is reasonably priced, and can be easilyarranged into the required 8X8 matrix. The PIA could be a 6520, 6522, 6820, 6821, etc.



MODIFIED CASSETTE INTERFACE (NRZ) SEE NEWSLETTER



PARTS.

2 - 7401 QUAD NAND clc
1 - LM324 OP AMP
1 - CD4009 HEX INV CMOS
1 75150 LEVEL CONV.
1 75154 LEVEL CONV.

5 - 1KΩ . 4 - 2.2K . 4 - 2.7K
4 - 100K . 2 - 470K . 2 - 2.2M
2 - 18K . 2 47K 2 - 270Ω
2 - 200K TRIMPOTS 2 56K
4 - 1 MFD - 25 VOLT ELECTRO.
2 - 10MFD 25VOLT TANT
2 - 1MFD 25 VOLT TANT
2 - LEDS # ADJUST FOR LEDS.
VOLTAGE REGS: 7812 . 7912 . 7805
2 470 MFD 25V 1 10UF 16V TANT
1 1000 MFD 16V.

ACOUSTIC MODEM

SETTING UP THE FSK TRANSMITTER

(a) Switch the Modem to SEND.
(b) Apply a short circuit to the data input.
(c) Connect a frequency meter to TP.
(d) Adjust VR3 until 1000Hz is shown on frequency meter.
(e) Now remove the short circuit from the input and apply +5V with respect to the common connection.
(f) Adjust VR2 until the new frequency becomes 1200Hz i.e. 200Hz *change* between "0" and "1".
(g) An audible indication will verify that all is well as these two frequencies are trimmed.

When this has been carried out it is worth applying a low frequency square wave to the data input of the transmitter to ensure that the frequency shift operation follows the logic levels being applied. A circuit suitable to carry out this test is shown in Fig. 4.

THE RECEIVER SECTION

Let us assume we are receiving the transmitted signal from a distant modem. The modem at our end picks up from the telephone handset the faint signals and applies them to an audio amplifier (IC3). A gain of approximately 100 was found to be sufficient to provide a working signal for the phase locked loop (IC4). C2 couples the FSK (Frequency-Shift Keying) signal to pin 2 of IC4. Diodes D5, D6, across R16 serve to provide some degree of limiting, as well as protection to IC4's input. Limiting by D5, D6 prevents amplitude modulation interference. The timing components VR1, C3, determine the frequency that the phase locked loop will free run at. This frequency should be adjusted to be 1100Hz, i.e. between the two FSK frequencies to be received. When a logic "0" is received (1000Hz) at pin 2 of IC4, the phase locked loop will lock onto this causing the voltage controlled oscillator of the PLL to suddenly shift from 1100Hz to 1000Hz. As this happens, the output signal from the phase comparator (pin 7 of IC4) becomes negative with respect to the reference potential at pin 6 of IC4. This potential difference will force the d.c. comparator IC5 to swing negative and this output signal will be caught at approximately −0·5V by D7 when the logic "0" is received. If now we receive a logic "1" signal (1200Hz) at pin 2 of IC4, the PLL will lock with its VCO frequency at 1200Hz, resulting in the phase comparator output becoming greater than the d.c. reference on pin 6 of IC4. Thus the voltage comparator IC5

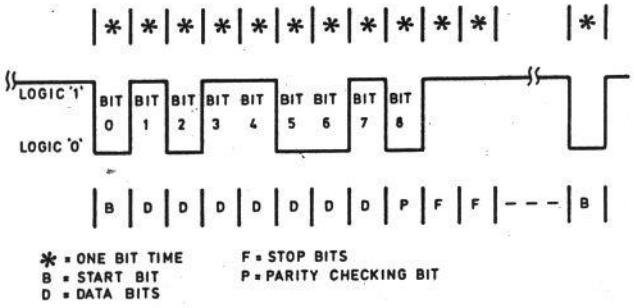


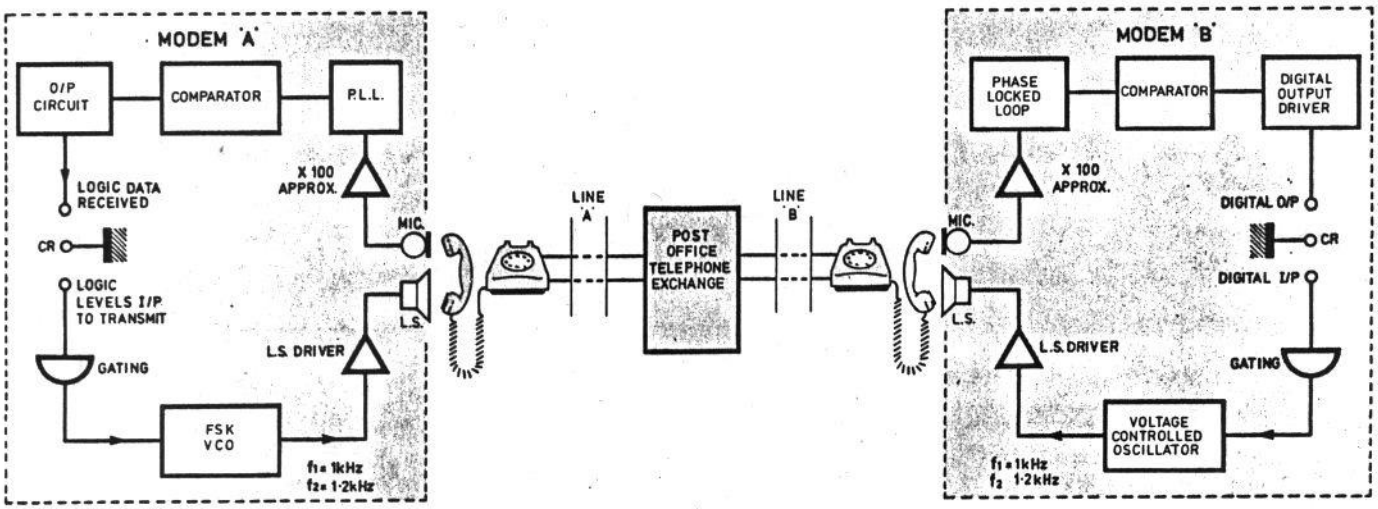
Fig. 1. ASCII code format, in this case representing the character "M"

b7→b5	000	001	010	011	100	101	110	111
b4→b1								
0000	NUL	DLE	SP	0	@	P		p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	—	=	M	]	m	}
1110	SO	RS	period	>	N	^	n	~
1111	SI	US	/	?	O	—	o	DELete

Fig. 2. The ASCII code, showing the binary equivalents of each character

responds with an output voltage positive going. It will in fact saturate in the positive direction at about 4·4 volts, providing an adequate logic level to become our Data output. C5 is chosen for determining the lock range of PLL, while R19, C6, R20, C7, R21, C9 form a filter network preventing the possibility of the VCO frequency, or harmonics, upsetting the output comparator.

Fig. 3. Block schematic of the Modem link-up. The system is half-duplex. Serial pulse repetition rates should be no higher than the VCO Tx frequency. Input levels are TTL. Applications might include micro' link-ups at 110 Baud, VDU communication using a 20mA loop, facsimile transfer etc. Trials with the Modem included a link-up with a teletype from Bebington (Merseyside) to Luton



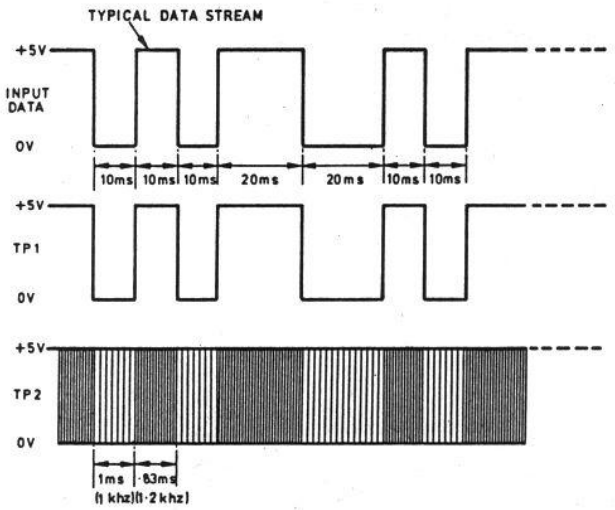


Fig. 12. Transmitter Test Point waveforms. When no input is applied and the Modem is switched to Transmit, the 1Ms square wave will appear continuously at Test Point 2

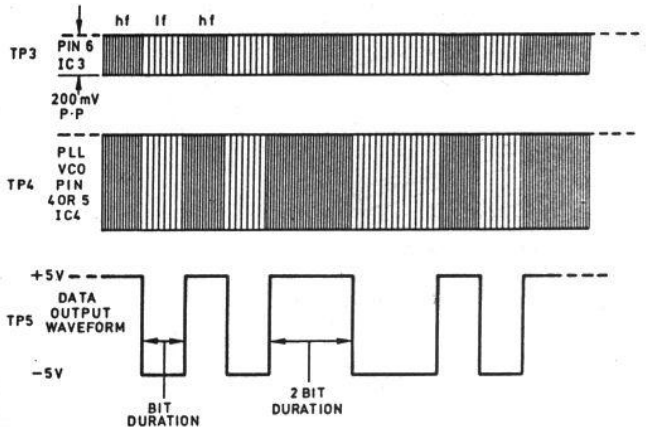


Fig. 13. Receiver waveforms. Example shown is the ASCII character "M". The l.f. pulses, i.e. logic 0, are a burst of 1ms pulses for the duration of the bit. The h.f. burst (logic 1) contains bit blocks 0·83ms wide. For Teletype operation the output bit duration is about 10ms

RECEIVER CHECK

(a) Look at signal on TP3 to ensure it is at least 250mV peak to peak when receiving from distant modem.
(b) If not, the gain of IC3 may have to be raised by increasing R15 (100k).
(c) With no signal being received, adjust VR1 until the frequency meter placed on TP4 reads 1100Hz.
(d) With Data being received observe TP5 and refer to receiver waveforms information.

DATA INDICATOR AND OUTPUT OPTIONS

Signals from both the transmitter and receiver are fed to l.e.d.s D3 and D4. This allows us to monitor the data both when we are sending and also receiving. It should be made clear that the modem will not be doing both at once, being *half*-duplex.

TR3 switches on when the output of gate (IC1) goes high (Transmit Mode) or when Pin 6 of IC5 goes positive (Receive Mode). This will result in (D3) turning on and indicating a logic "1" level being present. As TR3 is turned on, R12 is virtually connected to ground, thus turning off TR5. The transistors TR3 and TR5 form a "see-saw" stage. When one is off the other is on, so that when no base current flows in TR3 it is a sure sign that a logic "0" is being received or transmitted, thus TR5 will be turned on and the "0" l.e.d. will be illuminated.

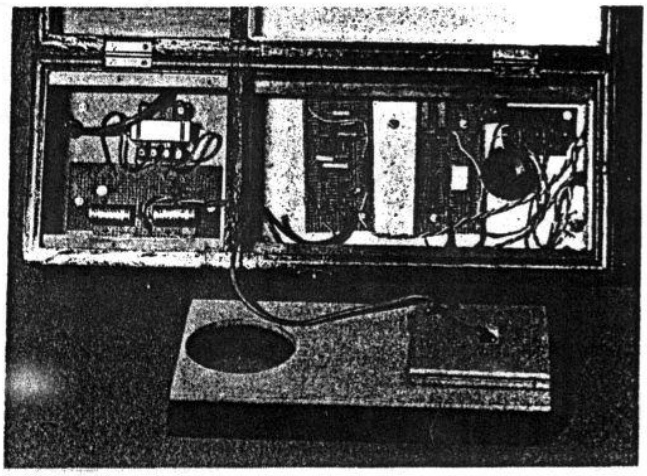
TP5 would normally be the receiver output point. It might be that a 180° signal is required to drive the device coupled to the modem. If this is so, connect the output of the modem to the open collector of TR4. i.e. TR4 will be turned on when "0" is being received. This is suitable for coupling to a teletypewriter.

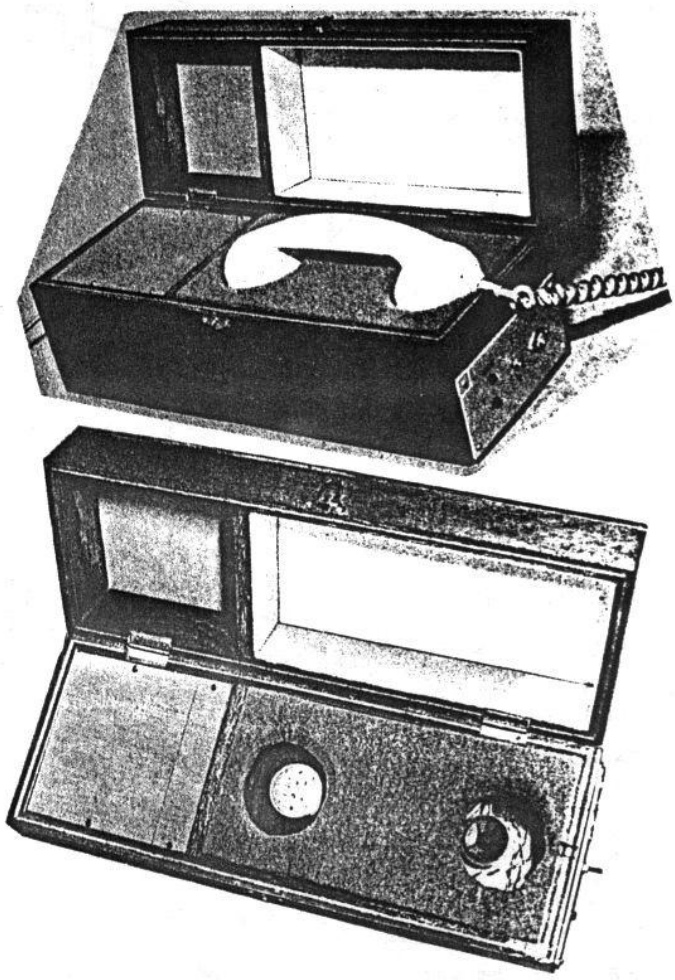
COMPONENTS . . .

Resistors

R1	18k
R2	5k6
R3	47k
R4	27k
R5	1M
R6, R8–R10, R19–R21	10k (7 off)
R7	82
R11, R12	270 (2 off)
R13	100k
R14, R16, R17	1k (3 off)
R15	100k nom. (subject to trimming)
R18, R22	2k2 (2 off)

All resistors 5% ¼W unless otherwise stated





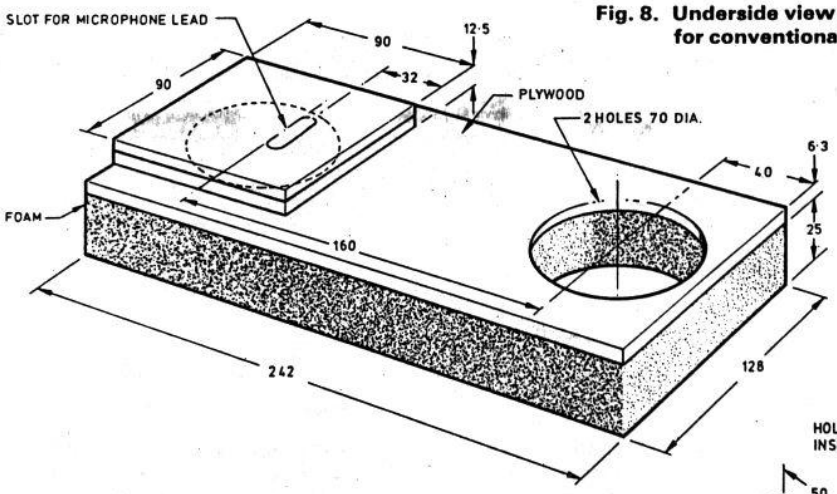


Fig. 8. Underside view of handset mount for conventional telephone

DIMENSIONS IN mm.

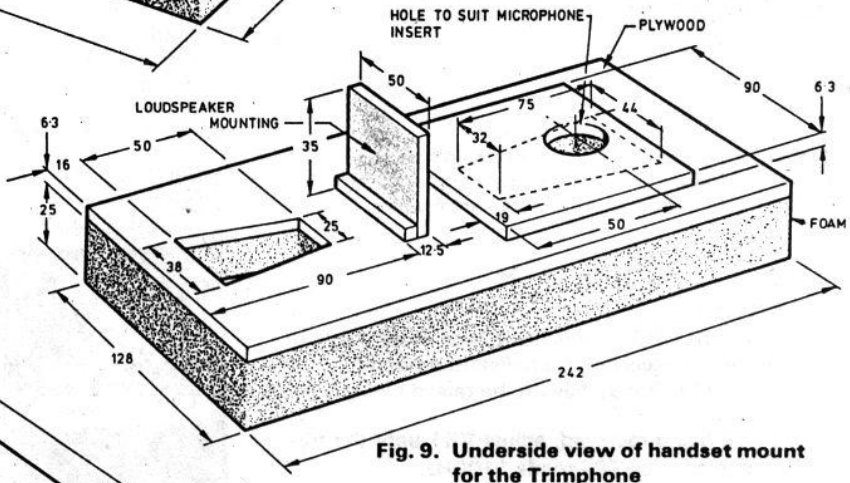


Fig. 9. Underside view of handset mount for the Trimphone

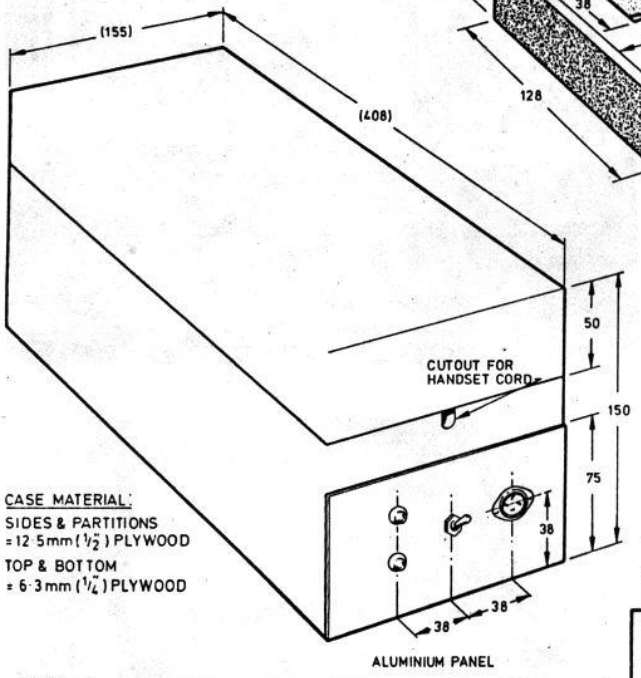
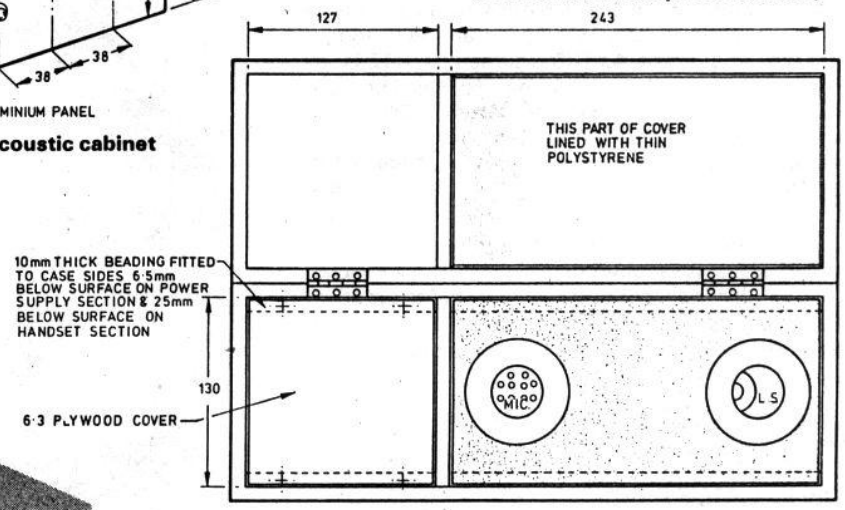


Fig. 10. Overall dimensions of the acoustic cabinet

Fig. 11. Top view of cabinet with lid fully open (Conventional telephone version)



Miscellaneous

Crystal microphone insert
Miniature loudspeaker 8Ω 100mW
Transformer 0–12V: 0–12V: 3VA
I/O d.i.n. socket (3-pin)
Double-pole, double-throw switch (send/receive)

Capacitors

C1, C3	47n WIMA FKC-3 temp. stable or equiv. (2 off)
C2	100n
C4, C9	1000p (2 off)
C5	220n
C6–C8	22n (3 off)
C10, C11	1000µ/25V (2 off)
C12, C13	470n (2 off)

Ratings and types not critical ... specified

Integrated Circuits

IC1	CD4011
IC2	4046
IC3, IC5	741
IC4	565
IC6, IC7	5V 1A regulator (plastic)

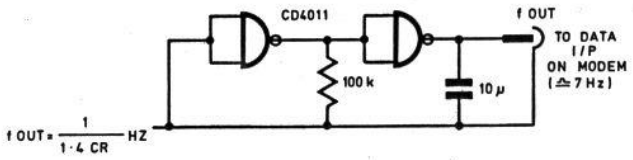


Fig. 4. Test oscillator for FSK transmitter

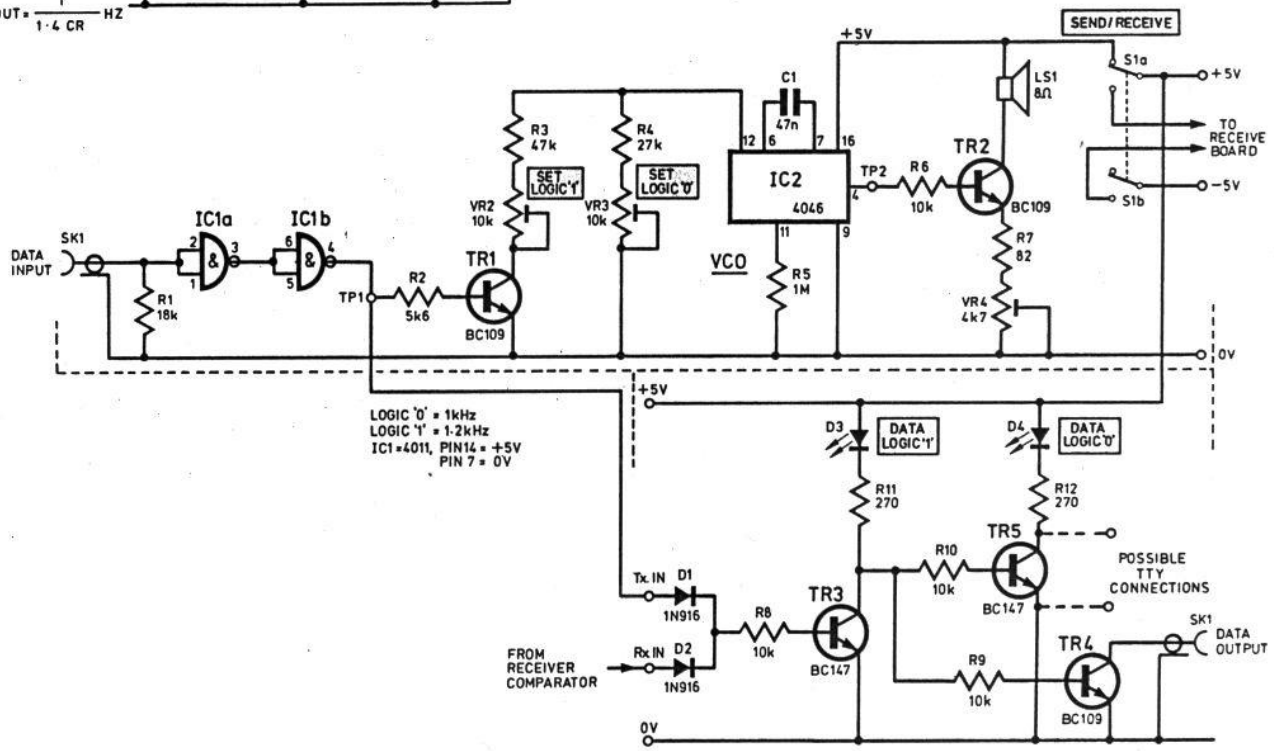


Fig. 5. FSK transmitter of Acoustic Modem

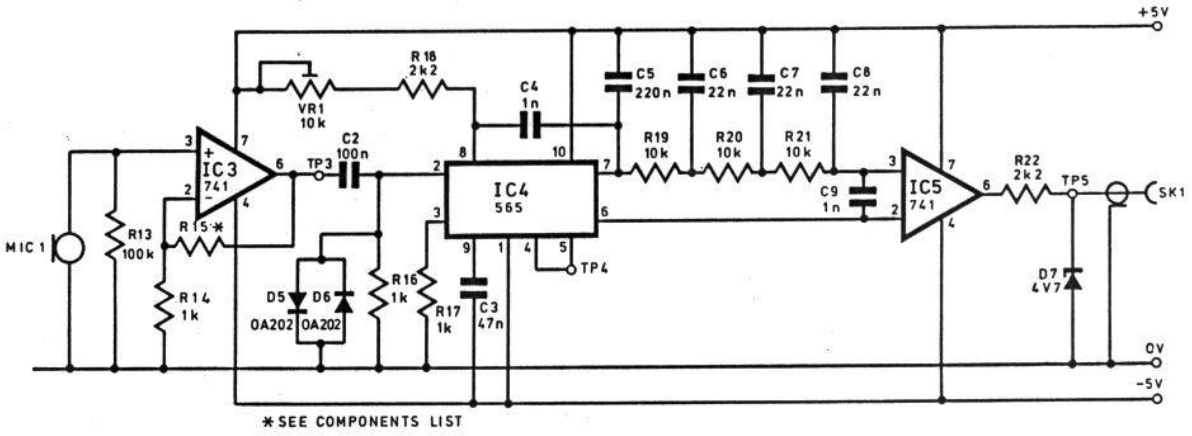
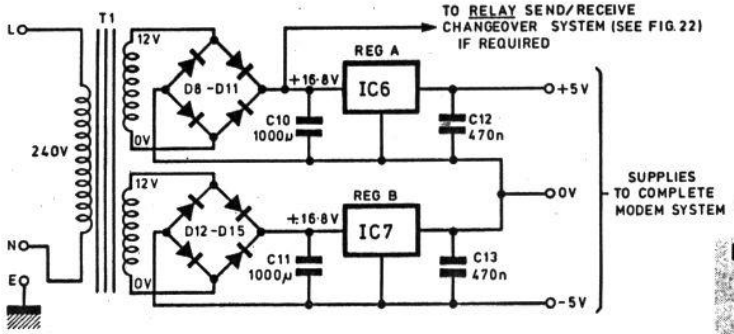


Fig. 6. FSK receiver section



Potentiometers

VR1, VR2, VR3	10k preset (3 off)
VR4	4k7 preset

Transistors and Diodes

TR1, TR2, TR4	BC109 (3 off)
TR3, TR5	BC147 (2 off)
D1, D2	1N916 (2 off)
D3	l.e.d. (red)
D4	l.e.d. (green)
D5, D6	OA202 (2 off)
D7	4V7 Zener (BZY 88)
D8–D11, D12–D15	1A 50V rectifier bridge (2 off)