AMATEUR COMPUTER CLUB NEWSLETTER

VOLUME 5 ISSUE 5 DEC 1977

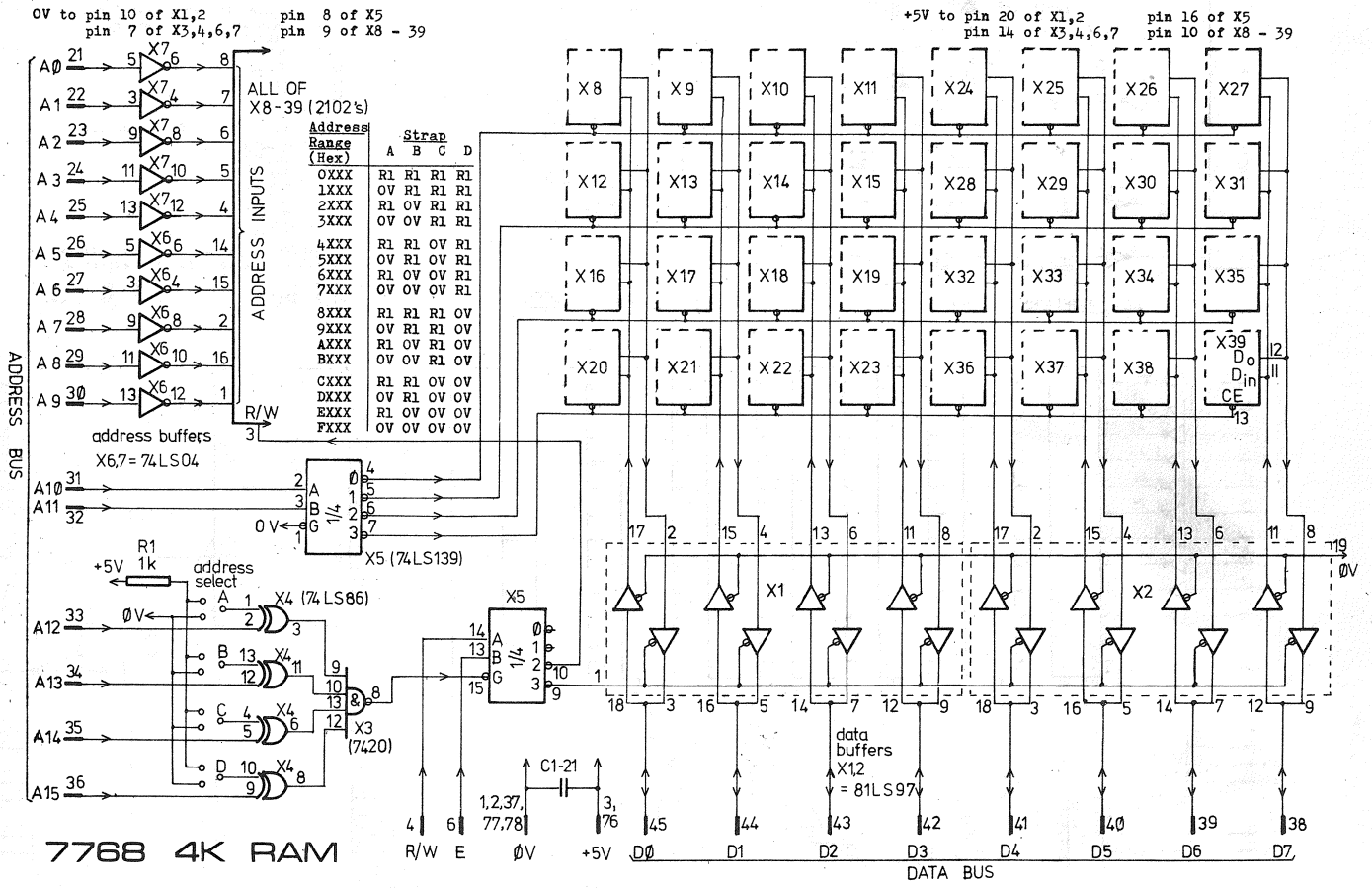
7768 MICROCOMPUTER

IN THIS ISSUE

As foretold in the previous issue of ACCN, we now have details of two extensions to the basic 77-68 CPU board;

7768 4k RAM

This is a straightforward 'cooking' design which should work with most current CPU's as well as the 7768 for which it was designed. Straps allow it to be set to any 4k block within the 64k byte address range. X8-39 are 2102's; the cheap low speed versions are OK for use with the original 1.6uS cycle time CPU board, otherwise choose a speed selection to suit the CPU timing.



7768 4K RAM

7768 MON 1

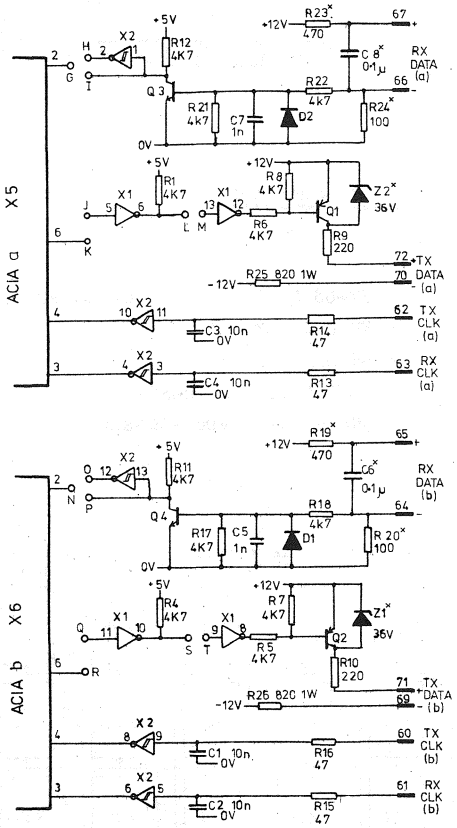
When the excitement of toggling programs into the basic 77-68 CPU board by hand has subsided a little, most people will wish to expand the system, not only by adding more memory, but also by linking up to some external device such as a cassette tape recorder or paper tape reader/punch combination so that long programs may be stored permanently. As well, anyone lucky enough to have a Teletype or VDU will want to be able to connect this, and to have some form of operating system, or monitor, to drive it. The MON 1 board has been produced to solve these problems.

It comprises;

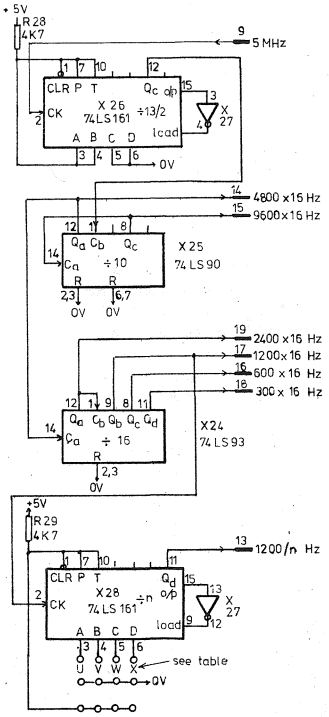
- Two UART's (one for initial use, one for later system expansion) with output buffers to interface with TTL or 20mA current loop or RS232 level devices.
- A divider chain, fed from the 5MHz CPU clock, to provide the correct frequencies for the UART's.
- 1024 bytes of memory, which is located at the top of memory address space, in the correct position for a system monitor. The address decoding circuits for this RAM also determine the address location of the 256 byte RAM on the basic 77-68 CPU board so the two do not conflict.
- Write protection circuitry for the 1k byte RAM, so that when it is used to hold an operating system it cannot be corrupted by malfunction of the user program. This feature is optional, strap selectable.
- Provision for a 32 or 64 byte 'Bootstrap' loader in easily programmed TTL 32 x 8 bit PROM. This is intended to be used to load a full system monitor (from cassette or paper tape) into the 1k RAM via one of the serial input ports.

Operation of the bootstrap facility deserves some explanation; both the PROM (X3,4) and RAM(X15-22) occupy the same address space at top of memory,

7768 MON 1



ACIA BUFFERS *=20mA only



ACIA CLOCK DIVIDERS

Straps;

D-E to write protect 1k RAM, otherwise D-F

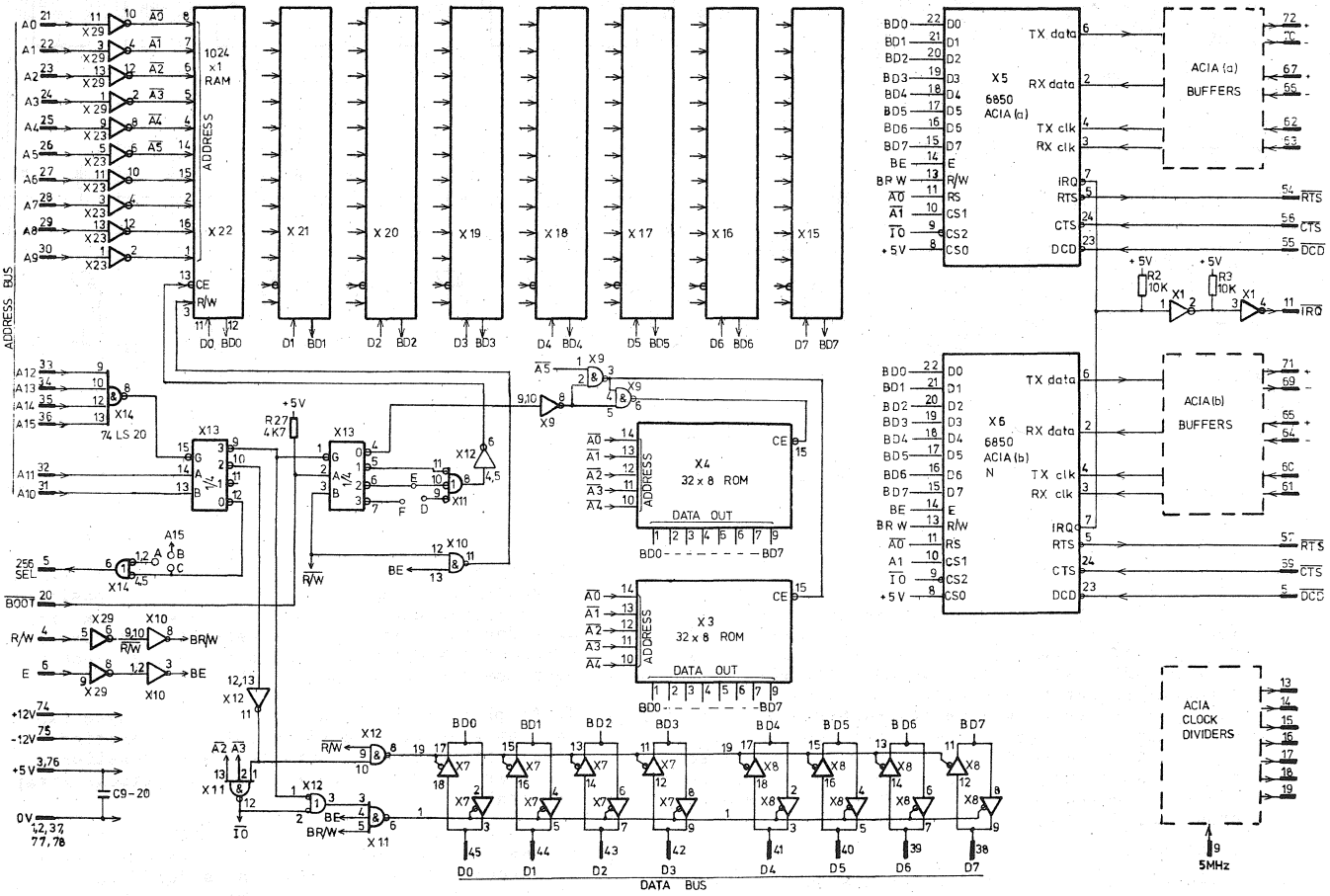
A-B for 256 word RAM on CPU to respond to low addresses, otherwise A-C

G-H & K-M for 20mA (ACIA a)
N-O & R-T for 20mA (ACIA b)
G-I,J-K,L-M for RS232 (ACIA a)
N-P,R-Q,S-T for RS232 (ACIA b)

X1	7406
X2	74LS14
X3,4	32 x 8 TTL tri-state PROM containing bootstrap prog.
X5,6	6850 ACIA's (X6 optional)
X7,8	81LS97
X9,10	74LS00
X11	74LS10
X12	74LS00
X13	74LS139
X14	74LS20
X15-22	2102 (2102-1 for 1.2uS CPU)
X23	74LS04
X24	74LS93
X25	74LS90
X26	74LS161
X27	74LS04
X28	74LS161
X29	74LS04

(although 'LS versions given, standard 74 TTL may be used for all except X4-7)

Q1,2	2N2907	Q3,4	BC107
D1,2	1N4148	Z1,2	36V 0.4W



and selection of the appropriate memory depends upon the setting of a 'Boot' switch (connected between card pin 20 and 0V), and whether the CPU is doing a Read or a Write.

When the CPU does a Write into the top 1k of memory, then;

-With the Boot switch open, data will be written into the 1k RAM if the write protection option has not been strapped.

-With the Boot switch closed, then data will be written into the RAM regardless of the setting of the write protection strap.

When the CPU does a Read from the top 1k of memory, then;

-With the Boot switch open, data will be read from the 1k RAM.

-With the Boot switch closed, data will be read from the PROM.(Because of the incomplete address decoding, the 64 bytes of data in the PROM's are 'echoed' across all of the top 1k of memory address space, however this is of no concern)

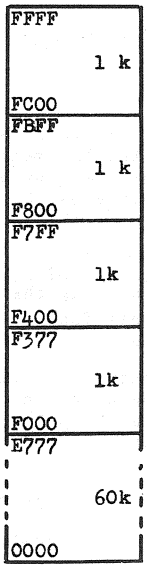
Thus, with the Boot switch closed, a momentary operation of the CPU Reset switch will cause the CPU to begin executing instructions from the PROM, and these instructions can cause data to be stored in the RAM - while when the Boot switch is opened, momentary operation of the Reset switch will start the CPU executing the program in RAM.

A suitable PROM bootstrap program is shown, this is a minimum version trimmed to occupy only 32 bytes (one PROM) and merely loads the first 1024 bytes of data to come in via ACIA (a) into the RAM without any form of checking - but it has proved sufficient in practice. A listing of a rudimentary 'Dump' program is also given; toggle this in, then run it to generate a tape containing itself, which can then be loaded with the Bootstrap program in PROM ! An operating system ?!

This approach, of using a 'Soft Monitor', has been adopted in preference to the more usual method of keeping the monitor in ROM because it is more flexible, allowing the user to modify his operating system to suit his particular system as it expands, and possibly to incorporate ideas gleaned from other monitors.

A word of warning for those who want to generate their own PROM's; if you build the board exactly as shown note that the address inputs to the PROM's are inverted and out of sequence relative to the connections assumed in the manufacturer's data.

The table shows the addresses fixed by the MON 1 card. A compromise has been reached between the amount of hardware needed to fully decode the address lines, and the practical use to which the memory address space saved by full decoding can be put. The decoding used allows 60k of memory space free for additional memory, as well as 12 addresses for additional I/O and 1k for a memory mapped VDU without encroaching on the 60k. This seems to be a reasonable solution.



1k RAM for monitor plus 32/64 byte bootstrap ROM (reflected through all 1k address space)

Reserved for memory mapped VDU

I/O ; see other figure

256 word RAM on CPU card (reflected 4 times)

Space for additional memory

77-68 Memory Map for use with MON 1

X14 pin 6 output enables the 256 word RAM on the CPU board for addresses in the range F000 to F3FF (256 words echoed across 1k address space) and it is envisaged that a full system (with extra memory) would use this 256 words as a scratchpad for use by the system monitor. However for those without any additional memory, strapping A to B causes the 256 word memory to be reflected across the entire lower half of memory address space as well to take advantage of the short (Direct) addressing mode that the 6800 can use to reference the lowest 256 words of memory space.

I/O address allocations;

F400	ACIA (a)	data registers
F401	" "	control/status registers
F402	ACIA (b)	data registers
F403	" "	control/status registers
F404 - F40F		Not used by MON 1 card
F410 - F7FF		Reflections of F400 - F40F

X28 strapping

U	V	W	X	n	Useful output frequency
0V	0V	0V	0V	16	75 x 16 Hz
0V	0V	R29	0V	12	100 x 16 Hz
R29	0V	R29	0V	11	110 x 16 Hz
0V	0V	0V	R29	8	150 x 16 Hz

The cassette/paper tape interface has not been included on this board, mainly because several good cassette interface designs are in the ACC general library, or kits can readily be purchased from several suppliers; the choice is left up to the user.

Although this board was primarily designed to be part of a large system with terminal(s), it can nevertheless be used with a 'bare bones' switch + LED system. But, the user should note;

-Since we now have more than 256 words of memory, care must be taken to use the full 16 bit addresses where required, especially when loading the Reset vector into FFFE & FFFF to establish the program start point.

-The CPU card switch and data register address is no longer top of memory, but F0FF (also F1FF,F2FF, F3FF and, when A has been strapped to B, 00FF, 01FF . . . EFFF). Thus a Reset no longer picks up the program start address from the switch register, but from the 1k RAM locations FFFE & FFFF, which must have been loaded accordingly.

-The Load logic on the CPU card still loads into the top 256 words of memory, which is now part of the 1k RAM, rather than the 256 word RAM on the CPU board.

Data and Address busses, and control bus lines (E,R/W,256SEL,5MHz) on the MON 1 and 4k RAM boards connect directly to the correspondingly numbered pins on the CPU card socket.

IC numbering and pin connections shown for the MON 1 and 4k RAM are as the 8.0" x 8.0" single-sided PCB's available from; Newbear Computing Store,7 Bone Lane, Newbury,Berks tel (0635) 46898

BOOT - 32 byte bootstrap program for PROM X4, reads 1024 bytes from ACIA(a) into top of memory.

```
FFE0  86 03      START: LDA A #3     reset ACIA(a).
  E2  B7 F4 01          STA A F401
  E5  86 11             LDA A #11    set up ACIA(a).
  E7  B7 F4 01          STA A F401
  EA  CE FC 00          LDX  FC00    point to start of
                                     top 1k of mem.
  ED  35                TXS          stack must be put
                                     somewhere.
  EE  B6 F4 01  LOOP:   LDA A F401   ACIA(a) full ?
  F1  85 01             BIT A #1
  F3  27 F9             BEQ LOOP     hang about.
  F5  B6 F4 00          LDA A F400   get data byte.
  F8  A7 00             STA A 0,X    store it.
  FA  08                INX          point @ next loc.
  FB  26 F1             BNE LOOP     passed loc FFFF ?
  FD  3E                WAI          kill 'run' lamp.
  FE  FF E0             START        reset vector.
```

DUMP - transmits top 1024 bytes of memory via ACIA (a) as 8 bit bytes .

```
FFDE  8E FF D0  START: LDS   FFD0   point stack
                                    somewhere.
  E1  86 03            LDA A  3     reset ACIA (a)
  E3  B7 F4 01         STA A F401
  E6  86 11            LDA A  11    set ACIA(a)
  E8  B7 F4 01         STA A F401   control.
  EB  CE FC 00         LDX  FC00    point index reg
                                    to start of 1k.
  EE  B6 F4 01  LOOP:  LDA A F401   ACIA busy ?
  F1  85 02            BIT A  2
  F3  27 F9            BEQ LOOP     hang about
  F5  A6 00            LDA A 0,X    get byte
  F7  B7 F4 00         STA A F400   send it
  FA  08               INX          point @ next
  FB  26 F1            BNE LOOP     finished ?
  FD  3E               WAI          halt CPU
  FE  FF DE            START        reset vector
```

SHOP

6500, KIM SUPPLIER CHANGE

The KIM 1 microcomputer, and IC's from the 6500 MPU family, are now available from G.R.Electronics Ltd.,80 Church Rd.,Newport,Gwent NPT 7EH, and are no longer available from Aardalect Electronics.

UK PERSONAL COMPUTING MAGAZINE

50p per issue, six times a year, 'Personal Computing World' is Britain's first hobby magazine, and will be available from LP Enterprises at 313 Kingston Rd.,Ilford, Essex from January 1978.

FOR SALE

IBM 3982 model 1 and model 2 terminals with maintenance agreement available. Remote or local (state which). £350.00 or £100 down and rest over 2 years.

Cheap time on an ICL 1903 for ACC members. 32k, tapes,discs,plotter,cards & paper tape, printers, plotter, bells & whistles. Usual languages etc. Batch only. Some hardware is available for sale; EDS disc drives, various peripherals & odd bits.

Galdor,52 Brighton Rd.,Surbiton,Surrey
01 399 1300

COMPUTER BOOKS

L.P.Enterprises, of 313 Kingston Rd.,Ilford,Essex now distribute a useful range of computer books, as well as magazines. Among them are the famous and highly recommended 'Introduction to Microcomputers' series by Adam Osborne; Vol 0 'Beginner's Book' and Vol 1 'Basic Concepts' @ £5.95 each, Vol 2 'Some Real Products' @ £11.95. Send an SAE for a full list.

WANTED

We have bought four Rank ENM ½" tape cartridge transports at the Electronic Hobbies computer sale, but have only 4 cartridges. Somebody bought a whole box of cartridges before we got there, and we would very much like to negotiate to buy some of them.

Simon Peyton Jones
Fairlawn,Riverwoods,Marlow,Bucks.
tel Marlow 4404

ANOTHER COMP SHOP

I have located a computer shop in Canterbury. Has ICL ASCII keyboards and many ICL readers,punches, VDU etc. Address is Cursons Industrial, 78 Northgate,Canterbury,Kent tel 0227 65442

FOR SALE

One working Creed 7E teleprinter. £25 o.n.o. No cover.
T.Liles 8 Turnstones Court,105 Westgate Bay Ave., Westgate,Kent CT8 8NX tel;(0843) 55920

LETTERS

SUPER SC/MP VDU PLUSSES

My VDU design in October ACCN has undergone further improvement, and with little more than a rearrangement of a few wires, the visible display is now 24 lines, and the underlining effect removed. The addition of a further monostable has eliminated the white flecks on the screen caused by computer access. I have redrawn the diagram to show the precise details of the address multiplexer together with the new mods for the benefit of another ACC member, and if anyone else is interested I will gladly send them a copy on receipt of an SAE. I should mention that on the published diagram X10 pin 2 goes to the preset pin 2 of X9. The number has got missed off the drawing.
By the way, has anyone got a listing of Nationals NIBL BASIC interpreter for the SC/MP ?

Bill Marshall
64 Pine Ave, Gravesend, Kent DA12 1QZ

MINE S-100 SYSTEM

Thought you'd like to know how my system is progressing. My micro consists of the S.D.Sales Z80, TDL system monitor board with 2 serial ports, parallel port, and fast (8k in just over 1 minute) cassette interface, 2 GODBOUT 4k RAM boards, and Poly chassis.

Start up was delayed because of faults on the CPU board. These consisted of two microscopic bridges between parallel tracks beneath the solder mask. An inverter had to be added external to the CPU board to create one of the S100 signals which SD Sales left off.

The Zapple monitor itself is fascinating and one needs to spend several hours with it before becoming familiar with the 26 commands.

I now have two 8080 interpreters operating; ProTech 5k BASIC and the Pilot interpreter as published in Dr. Dobbs (ever typed in 4k of Hex characters from a teletype ?) The BASIC seems a bit heavy on memory considering the command set. PILOT (Programmed Instruction Learning Or Teaching) provides for conversational interaction between user and machine and is of particular interest to educators. In fact when initially loaded my tapes enter a pilot program for teaching the user how to program in Pilot !

At this moment I am typing in 'CASUAL', another interpreter from Dr. Dobbs for the 8080.

I have TDL's 8k Z80 BASIC but will not have memory to use it until after Christmas (I have never stopped believing in Santa Claus).

Anyone interested in Pilot (or the others) please contact me. I do need to add a line editor to Pilot. In its present form the only way to change a program is to re-enter the whole thing (or use Zapple to locate the string in the program space, and to type ASCII in place of the original string, only problem is when the length of the new line is longer than the old). I also need to complete the patches to enable the program save and load commands to access Zapple.
Kem Talbot 33 Easter Livilands, St.Ninians,Stirling, FK7 0BQ tel Stirling 70126

6100 PROBLEMS SOLVED

My computer system is nearing the end of its first stage of development. It is based on the Intersil/ Harris 6100 12 bit micro. I have found some rather fundamental problems with this chip which may be of interest to other ACCN readers;

1) MPU chips date coded pre-week 46 of 1977 generate an LXMAR pulse during the 3rd phase of an IOT instruction. At this time the MPU is NOT driving the DX (data/address) bus, hence device interfaces, memory etc. latch some undefined address.

2) As a result of (1)the IM6508 or HM6508 CMOS RAM will corrupt the contents of stored data if its chip-select has a -ve edge when the data input line is between logic levels (e.g. floating). Chip select is generated from LXMAR.

The cure is to use 10k pull down termination resistors.

P J Kingsland

FREEDOM AT LAST !

With the exception of the switch and lamp processors most 6800 micro users have an inbuilt monitor program, or 'Bug'. The purpose of the Bug is essentially to get the machine going, and to provide useful routines as well. The monitor can be in the form of a PROM e.g. 6830, or can be a bootstrap program that will enable the monitor to be loaded into RAM. The monitor in RAM is then executed. The latter method has the advantage of enabling modifications to the monitor very easily. It does need loading however and someone has to generate it on another machine in the first place, or you switch load it into your own (5 finger exercises !). The PROM method is of course instant and examples are MIKBUG,SWTBUG,RT68 etc. For further information on these read elsewhere. The PROM method is convenient in the first place, but most have shortcomings since two systems are seldom alike and having tried most I decided to go it alone.
The question of a new monitor was followed by another, how ?. The answer was 2708's, now quite cheap (£12 from USA) and the obvious choice. This in itself raised more problems. The first was programming and the second was compatibility. I built a very simple programmer (*** am trying to persuade Dave to let us have details for the next ANNC ed) and with some simple software was able to program easily, erasure was also easy since I have found a source of tubes for £6. Erasure with health lamps is also OK, but you will also get a tan if you do too many !

Armed with programmed 2708's, the little question of pin compatibility with 6830's had to be solved. I have designed a connector ('PROMVERTER') which takes 2708's and plugs into 6830 sockets. If your 2708 is not 5V type then a fly lead takes 12V and -5V onto the board. As the 2708 only has one chip enable, extra decoding has to be done with an on board 7410. I will be making the PROMVERTER available after January for less than a fiver (Hope!) also a programming service will be available at moderate costs so you can have your own version of ***BUG or whatever programmed for you.

We are now free so what to do with our new found versatility ? The early ideas will be modified versions of the standard Bugs, but things are afoot for a little revolution. Having modified routines for your IO devices and created a few extra ideas of your own there is still a more disturbing problem to be solved, that of compatibility. Most software available assumes certain routines are resident in the monitor, and altering things here may mean altering a lot of programs to suit, so an idea I had (since discovered not to be original) was to use a standard point of reference to software which wouldn't make any difference no matter what the monitor was, or where it was for that matter. If you havan't spotted it yet it is the 3F instruction; SWI.

Maybe you are puzzled, so an explanation of the proposed mechanism ; The 3F instruction is very special since it causes the machine to reference the SWI vector at FFFA,FFFB. The value here is the address of the SWI routine and the machine executes instructions from that point. Since this is an interrupt the machine pushes A,B,X etc onto the stack and this will be useful as you will see later. There will be a single byte following the /F (two or more bytes if you wish),this value is read by the SWI routine and the monitor will act accordingly e.g. 3F01; A reg to TTY, 3F02 A reg to cassette etc. You have 255 calls possible, enough for now ! Since the 3F instruction is universal, then providing we standardise on the calls a standard set of monitor calls could be established and this of course is what IBM do, only they call them 'Supervisor Calls'. What is good enough for IBM is good enough for us. The mechanics of the call would be; after the SWI the software reads the following byte then using lookup tables or using '2 x Call Value' as an index into a jump table, the monitor will execute as required. The stack contains all the register values at entry and since the exit from the monitor is via RTI the stack values can be used to pass parameters both to and from the monitor. The address on the stack is of course modified to ensure program continuation after the SWI call. The calls being 2 bytes would save a little space as well.

The advantage of common routines would, judging by the response at the 'London Group' meeting when monitors were mentioned, be well received.

So there it is, revolutionary or evolutionary ,say what you like but it is a different idea. I am actually writing a monitor using the '3F' principle and will modify all my software to match. If anyone tries it, patches for CO.RES,BASIC etc will be available. If anyone is interested in my ideas please write with your ideas so common ground can be established early on in the project. The debate starts, but I hope it doesn't end here !

Dave Goadby 2 Lupin Close,Hinckley,Leics LE10 2UJ
tel 0455 35621

KEEP IT UP

Thanks to Mr. Frank Cato's A.C.C. Library, I have recently had the chance to browse through all the back issues of A.C.C.N. and would now like to make a few comments.

It appears that the A.C.C. or possibly just A.C.C.N lacks some continuity with projects, articles etc. For instance, what happened to part 2 of 'Don't Throw Away That DTL' (A.C.C.N. October '74). I could not find it. Also it appears that WB2 was a definate development idea for publication but it seems to have died as such (killed by an MPU perhaps). However there must have been a lot of people waiting for the follow up article. Th WB, although not as powerful as a similarly priced MPU system, still has a place as a demonstration instrument as as a very good exercise in using TTL.

There was a suggestion some time ago that an A.C.C. project could be a V.D.U. However this was suppressed in view of articles appearing in national publications, and eventually the 7768 partly occurred. Having looked at most of the articles published in U.K. magazines I feel that there is still room for another design and will throw in the following suggestions in case any member has enough time and inclination to complete a design.

A problem with all the published designs is that they use page mode, but, to replace a Teletype, scroll mode is much better, perhaps with page mode as an option. Scrolling can be achieved easily, but expensiveley, with shift registers, but more cheaply with greater decoding difficulty using RAMs. RAMs (e.g. 2102s) however will allow line length options. To obtain a switchable display the decoding could best be done using a small MPU e.g. SC/MP II. This would also allow certain other functions to be easily incorporated such as not printing a word unless there is room on the line for it, generating graphics, off line activity etc. The A.C.C. could possibly market control programs in a PROM (e.g. 82S114). Perhaps it could also decode Teletext ? Also why not upper and lower case now that suitable ROMs are available (74S262,X887,2513/CM2141 + 2513/CM3021). U/C only for the low cost version. The full version can also have facilities for 5 level (Murray/Baudot) I/O as well as 8 level ASCII.

Now the wants. Is there anyone in this area who subscribes to Personal Computing, or similar magazines, who would be prepared to lend it to me regularly. I can offer on an exchange basis Byte, National Geographic, E.T.I. etc. (Why is it that the foreign magazines are so much better than the U.K. ones ?) Also, now that MIK BUG II is available is there anyone who wishes to dispose of their MIK BUG I cheaply ?.

I should be prepared to discuss any of the above on Hugh Wycombe 21150 (day) or 31314 (evening).

M.I.Connell G8HDL 38 White Close,High Wycombe.

COMPUTER-LESS PROGRAMMING ERROR

Ref the item on pages 6/7 of V5 I4 ACCN.

ST 1: IF 1=2:3:3:

should have been

ST 1: IF 1=2:2:3:3:

apologies ed.

BUSSES *Part 5* Useful Buffers

Turning to the medium power tri-state buffers, of most use to the average microcomputer builder, the most popular is National Semiconductor's DM81LS95-8 series . These contain eight tri-state buffers in a long thin (twenty pins on 0.3" spacing) package and as can be seen from the diagrams below are available as inverting or non-inverting buffers either controlled as two groups of four buffers or as a single group of eight. DM71LS95-8 are similar devices but specified for a military temperature range. In all cases the outputs are placed in the high impedance state by applying a high logic level to the enable pins.
Each output can sink 16mA (at 0.5V) and source 5mA (at 2.4V out) when on, and has a maximum leakage current of 20uA in the high impedance state. Each input takes a maximum of 0.36mA (low input) or 20uA (input high), corresponding to normal Low Power Schottky input loadings. Propogation delays are typically 10ns for the inverting types, and 13nS for the non-inverting LS95,97.

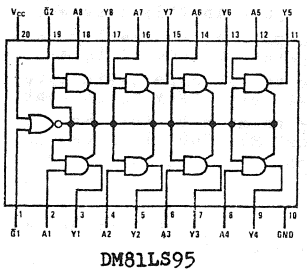
The Intel 8212 (or Texas 74S412) was designed to support the 8008, 8080 CPUs (and so provides 3.65V high level outputs rather than the 2.4V normally guaranteed for TTL) but is generally useful. The Q outputs of the eight latches follow their data inputs while their clock inputs (C) are high, and will latch when C returns low. Taking $\overline{CLR}$ low clears all eight latches. The Q output of each data latch is connected to a non inverting tri-state buffer. When activated ($\overline{EN}$ high) they can provide 15mA (0.45V 'low' output) or 1mA (output 3.65V). When EN is low, output leakage current is 100uA max. The data inputs to the latches take 0.25mA max for a 'low' input, 10uA max for a high level input.Thus both data inputs and outputs are compatible with a tri-state bus and the 8212 can be used either as an eight bit output port with latches (Connect MD input permanently high, and STB permanently low to keep the data output buffers enabled, then data will be latched in when the select inputs $\overline{DS1}$ & DS2 are respectively low and high), or as a latched input port (Connect MD low, then data will be latched from the input device by pulsing STB high, and put onto the system bus when $\overline{DS1}$ & DS2 are 0 & 1).

Intel also suggest that the 8212 can be used as a simple octal tri-state buffer (MD low and STB high to permanently enable the data latches) but this seems to be a bit of an over kill, especially as the typical chip power consumption is 90mA from 5V!

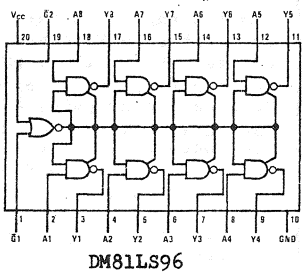
The main reason for this amount of current being taken is that the 8212 has been designed to be reasonably fast, propogation delays being typically 20 - 30nS.

DM81LS95,LS96,LS97,LS98

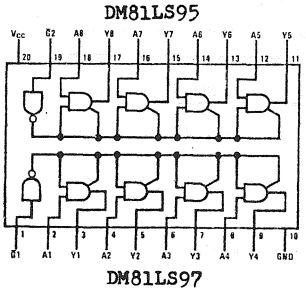
TRI-STATE Octal Buffers



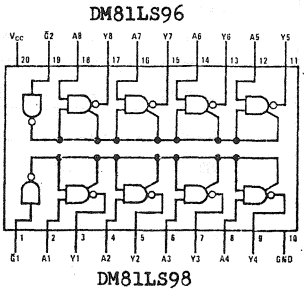
DM81LS95



DM81LS96



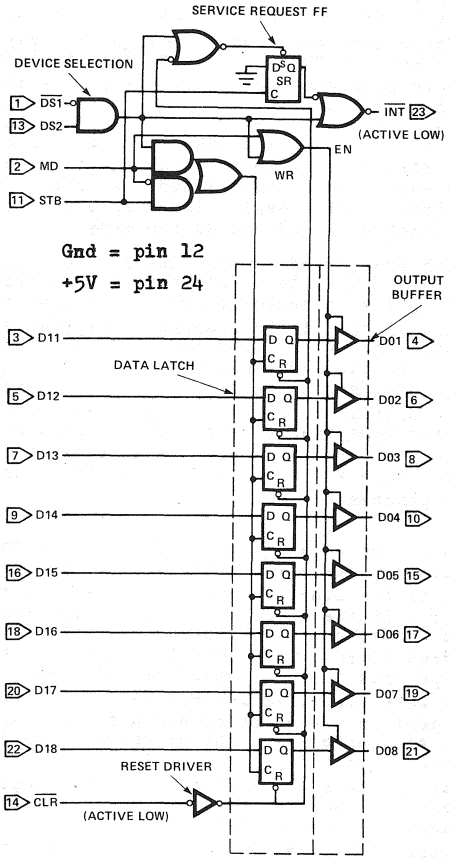
DM81LS97



DM81LS98

The Service Request (SR) flip-flop is used to generate and control interrupts. It is asynchronously set by the $\overline{CLR}$ input (active low).When set this flip-flop is in the non-interrupting state. The flip-flop is reset (Q output to 0) by a high to low level transition on its C input (e.g. the end of a STB pulse), and this causes the $\overline{INT}$ output to go low. By suitable selection of the MD and STB inputs, the 8212 can thus be used for a latched, interrupting input port or as an output latched port with handshaking via interrupts.

INTEL 8212 EIGHT BIT INPUT-OUTPUT PORT
(Also known as the SN74S412)



EDS BIT

In his letter (page 5 of this issue) Mr. Connell makes the valid point that the ACCN appears to lack continuity, with articles and projects stopping in mid flow, and I have to admit that he is right. Unfortunately, it is difficult to know quite what to do about it, as the causes seem to lie in the tremendously rapid changes that have been taking place in the hobby computer scene during the last few years - so that what is 'state of the art' now may well be out of date by the time a couple of newsletters have come and gone, and possibly also the fact that as the ACC is strictly an amateur, spare time, organisation, contributors aren't wholly committed to finishing what they started.

Talking of contributors; I'd welcome feedback on what kind of article you would like to see in the ACCN, and I would be most pleased to receive any articles which deal with the real fundamentals of hardware and software e.g. 'On Why A Byte Usually Has 8 Bits - And What A Bit Is Anyway', or 'What Is BASIC, Basically'. No, I'm not kidding - from

various conversations I've had with members, any articles along these lines would be gratefully received.

Usually I retype articles for the newsletter (poorly at the moment I'm afraid - must get a new typewriter), which is no great problem, but for those contributors willing to send finished artwork ready for publication, some guidelines;
The original artwork for this newsletter is pasted up onto A3 (420 x 297 mm) sheets which are then photographically reduced to A4 (297 x 210mm) giving a reduction of 2:1 (area) or $\sqrt{2}$:1 (linear). So, on the original artwork, the width of a column is;

120 mm

Each page of the original is made up by sticking together the various pieces of text and drawings. For decent reproduction the originals must be high contrast black on white, which means using a new typewriter ribbon, and black ink (or fibre tip) for the drawings. Photo copies can only be used if they are good quality with dense black lines. Reproducing computer print-out is a problem as you've usually got no control over the quality, and those green or grey rulings can be a nuisance. If possible send an ASCII paper tape & we'll print it under controlled conditions. Large print outs or drawings can be reduced further than the normal 2:1 if required.

Contributions for the next newsletter are required by 10th February please.

Finally, may I wish all ACC members a happy 1978, and I can assure you that it will be a fascinating year for all computer enthusiasts.

Mike Lord

ACC 6800 LIBRARY

This has flourished in 1977 thanks to support from all members. It holds 173 items of hardware information, 43 interface items, and 66 software.
On a directly practical front, designs have been slowly hammered out for two popular card frame systems; the first being a 43 way $4\frac{1}{2}$" x $8\frac{1}{2}$" card for the 3U rack - popular because it is a stock item of R S Components - this is supported with a PCB for a 2k RAM using 2102's. The second is the Crowe-Howland E77 double-Eurocard bus which mow has both a 6800 and a Z80 implementation. It is supported with a 4k RAM PCB.
The library is a free service to ACC members, for an up to date index please send 32p stamps and an A4 SAE to Tim Moore, 15 College Rd, Maidenhead,Berks (Note change of house number) tel 0628 29073

TRURO BIT BASHERS UNITE

Anyone living near Truro, take note of the Personal Computing Club, c/o The Micro-B Computer Store, 22 Lemon St., Truro, Cornwall, tel Truro 78487

HELP

Has anyone a circuit diagram for Ferranti VDU type CDU11A.
Colin Rowley Grassendale Cottage,Hanley Swan,Worcs.

BEAR MICROCOMPUTER SYSTEMS

Bear and Newbury Laboratories have joined forces to form the Newbear Computing Store. This is a retail shop where visitors are welcome Monday - Saturday inclusive. There are three sections; Hardware components, Literature & Software, and Systems.

All previous Bear wares are stocked, plus a great deal more. 7768 will be fully supported and all ACC discounts will be honoured. For more details write, phone or visit the NBCS 7 Bone Lane, Newbury,Berks tel (0635) 46898

Discount prices for ACC members;

7768 manual £4.50)
WB-1 manual £3.80) 30p P&P

7768 PCB's, all @ £7.75 each)
3U/2K RAM PCB, price to be announced.) +8%VAT and
Double Eurocard 4k RAM PCB £5.75) 30p P&P

MORE FOR SALE

IBM 1130 4k system running on Fortran, complete with tape reader, punch, communications adaptor, and full set of manuals. £750 (Built-in keyboard, golfball printer etc.)

Olivetti 8-bit terminal. ASR fully teletype compatible 110 baud upper and lower case. RS232 in/out £250.

Ferranti VDU,looks modern but no gen so only £100.

Litton ASCII terminal - 30 ch/sec printer & keyboard and all electronics. With stand,very modern £250.

Litton 1231 processor - very compact, plug-in TTL IC's.With magnetic drum store.(2048 x 24 bit) £120

Sagem 5-unit RO teletypes, very quiet and small with electronic decoding. £75 with lots of spares. 50 or 75 baud selectable.

Litton tape reader/punch. Made by Roytron,30 ch/s 8 level with all electronics. £80

Plessey 4k core stores £5 each

IBM card reader, no details,maybe working £10

Honywell 201 processor. Enough discrete components to last you a lifetime. Take it away for £15. No PSU, but full details.

Nigel Dunn, 21 Campion Rd.,Widmer End,High Wycombe, Bucks

FUTURISTIC ZILOG

Zilog are now talking about their next generation MPU chips; the Z8, a single chip processor with four I/O ports, 2k bytes of ROM and 97 bytes of RAM all on the same chip. Towards the end of 1978 watch out for the Z 8000, a 16 bit chip with hardware multiply and divide, and the ability to directly address up to EIGHT MILLION bytes of memory.

INTEL BITS

100 up prices for Intel EPROM are now £27.40 for the 2k x 8 bit 2716, £12.90 for the 1k x 8 2708, and £13.70 for the 5V 1k x 8 bit 2758.
Keep an eye open in mid 1978 for Intel's new MPU chip, the 8086. This will be a 16 bit device, although it is claimed to be software compatible with the 8080 but with additional instructions. Operating at 5MHz, Intel claim that the 8086 will have a processing capability similar to a PDP 11/45; about 10-15 times faster than a 8080, and it will be able to address ONE MILLION bytes.

MEETS

MIDLANDS MEETING SUNDAY 13th NOV.

The 7768's absent from the last meeting made their debut this time. Two examples were shown, the first was made by John Diamond, and had the refinement of a Hexadecimal keyboard. This was loaded with the music program published in the 7768 manual from BMS, and various different tunes were played. Other programs tried out included; a chiming clock which struck the appropriate number of times each hour, and a reaction timer which measured the speed of reaction to a counter display. The various programs shown will be submitted to the 7768 program library.

The other 7768 was shown by Nick Wright. He had interfaced a 'Meccano' XY plotter to his computer, and had it writing messages and drawing patterns. This marriage of Microprocessor and Meccano is an absolute 'Natural' and we look forward to a fascinating progeny.Nick finished by re-programming his computer to play the game Run Rabbit Run . The output LED's are sequentially switched on giving the impression of a running light - that is the 'Rabbit'. The idea is to depress one of the data switches when the 'rabbit' was on the corresponding output bit. This would 'kill' the rabbit and the light would be extinguished. But if you missed, an extra 'rabbit' would appear so that you could end up with several rabbits dashing across, and you would have to continue banging away until you had killed them all !

From a minimum computer to a maxi; Geof Cutler brought along his new Teletype - a very impressive dot matrix job running at 300 baud - together with a borrowed modem. So we rigged it up to the telephone, dialled up a bureau, and got down to some serious computing, such as a picture of 'Snoopy', a game of StarTrek etc.

Many thanks to those who brought equipment along.

Next meeting Sunday 5th Feb. Contact; Roy Diamond,
27 Loweswater Rd.,Coventry CV3 2HJ
tel Coventry 454061

BIGGER & BETTER D I Y

Remember May's DIY Computer Conference ? Now look forward to a three day show scheduled for June 22-24 at the West Centre Hotel, London.

THE VAUXHALL COMMUNITY COMPUTING CENTRE

Details from 132 South Lambert Rd.,London SW8

BOOKS

MICROPROCESSORS - TODAY & TOMORROW

A Book List by W.Old

It is almost impossible to prepare a recommended book list that will suit everyone, but the following titles should be considered as being potentially useful.

Also, it should be noted that few semiconductor manufacturers give away data books, and most insist that both industrial and private users obtain this sort of information from their authorised distributors. With some distributors this can be quite a marathon, and it is likely that the required data, or text book might be on the shelves of;

The Modern Book Co. Ltd.,19-21 Pread St.,Paddington, London W2 1NP tel 01-723-2926 or 4185

Texts on Hardware - may also deal with software

MINICOMPUTERS & MICROPROCESSORS by Martin Healy
Hodder & Stroughton

MICROCOMPUTERS/MICROPROCESSORS by Hilburn & Julich
Prentice Hall

M6800 MICROPROCESSOR APPLICATIONS MANUAL
Motorola

AN INTRODUCTION TO MICROCOMPUTERS Vols 1 & 2
by Adam Osborne

MICROPROCESSORS - NEW DIRECTIONS FOR DESIGNERS
by E A Torrero Hayden

Texts on Software - could have a hardware content

MICROPROCESSOR/MICROPROGRAMMING HANDBOOK by B Ward
Foulsham-Tab

SOFTWARE DESIGN FOR MICROPROCESSORS by Wester and Simpson Texas Instruments

M6800 MICROPROCESSOR PROGRAMMING MANUAL Motorola

8080 PROGRAMMING FOR LOGICAL DESIGN Adam Osborne

6800 PROGRAMMING FOR LOGICAL DESIGN Adam Asborne

Texts on Computer Games - These give listings etc.

101 BASIC COMPUTER GAMES by David H Ahl
Digital Equipment Corp /Creative Computing

WHAT TO DO AFTER YOU HIT RETURN
Hewlett Packard

Useful Text on Digital Circuitry

PRACTICAL DIGITAL DESIGN USING IC's by J D Greenfield
Wiley

CALCULATORS AND COMPUTERS J Hamilton
Part 1

INTRODUCTION

Is there any difference between a programmable electronic calculator and a computer? Yes,there is. In fact there are many differences. Although most of them are differences of degree in that computers and calculators perform similar operations which differ in complexity, there are also differences of function and architecture. The essential difference is architectural.

On the face of it, computers seem to be complex calculators. In practice, while computers can do everything that a calculator can do, the reverse is not true, and the reason is partly based on the difference between computation and calculation.

In this first article, I shall try to describe what a computer is and how it works. In the second article I shall then try to show how a programmable calculator differs. I shall give an example of a calculator program, and an equivalent computer program.

WHAT IS A COMPUTER

1.Computation and Data

Computation may be defined as the transformation of data. I shall define data as symbols embodying defined meaning. Words, numbers, punctuation symbols and mathematical and printers' symbols are all examples of data. Symbols are composed of characters, these being alphabetic;

	eg)	A	M	O	Z
numeric	eg)	0	1	5	9
alphanumeric	eg)	A	3	;	+
or graphic	eg)	D	♣	½	♀

Data transformation may also be classified according to the type of transformation. Some examples are arithmetic,logical,structural and ordering transformations. Arithmetic and ordering transformations are

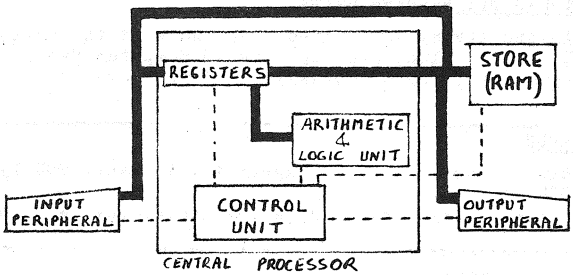


FIG. 1 A BASIC COMPUTER

those best known to the layman.

eg) 3 + 1 = 4

is an arithmetic transformation of 3 and 1 into 4.

GDACBFHE → ABCDEFGH

is an ordering transformation commonly known as 'sorting'.

Computers transform data. The description of a data transformation is called an algorithm, and languages developed to write algorithms are called algorithmic languages. In particular, an algorithmic language used to describe data transformations to be performed be a computer is called a computer programming language.

2. Basic Parts of a Computer

Early computers merely transformed data. Modern computers store data before transforming it (although a transformation occurs in the process of storing), and then display the results. Therefore, a computer comprises at least a store, a data processor, and two devices one of which accepts data (input) and the other displays it (output).

In what form is data stored in a computer? Analogue computers store the data as voltage levels. This article is not concerned with analogue computers which are radically different from electronic calculators and digital computers because data is stored and transformed continuously rather than in discrete steps. Digital computers store data as patterns of voltages which represent two states; 0 and 1 called binary digits or 'bits'. Computer stores are arrays of devices having two stable states; bistables. Early stores consisted of magnetic bistables made from ferrite rings. Some were as small as 2mm in diameter. Modern stores consist of electronic bistables in chip form. Arrays of four thousand bistables are quite common. Whichever bistable is used, one state represents 0, the other 1. By combining several bistables in parallel, several bits may be associated. A group of associated bits is commonly called a word. In particular, eight associated bits are called a byte .

Fig 2 shows an array of 1024 bistables whose associated logic enables tham to act as a store for 1024 bits. Eight such stores would be needed to store 1024 bytes.

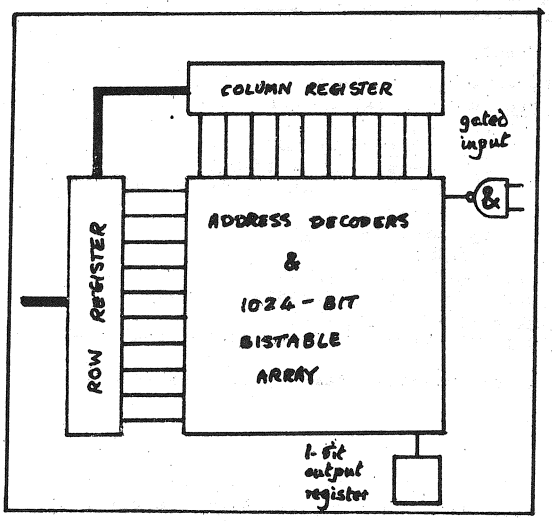
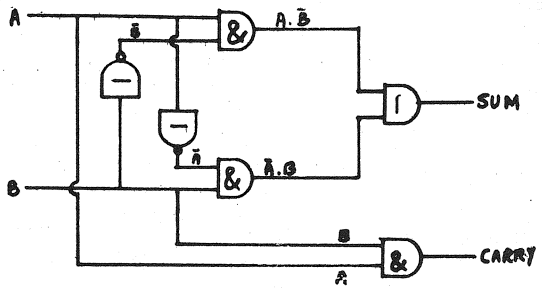


Figure 2. Block diagram of bistable store.

The data processor includes several registers for the temporary storage of data. Besides these registers, the processor contains the data transformer, usually called an 'Arithmetic and Logic Unit' or ALU, and a control unit. All transformations of data are, in fact, logical. All other transformations are either composed of logic transformations, or simulated by logic. Fig 3 shows the logic diagram and associated truth table of a half-adder. Two half-adders make a whole adder (almost). Thus, a computer cannot even add 1 and 1! It simulates arithmetic.



A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 3. Logic schema and associated truth-table of a half-adder.

Logic functions such as AND, OR, NOR and shift are performed by other combinations of logic circuits. The point here is that computers (and calculators) are really logic machines which seem to perform arithmetic. Mechanical calculating machines add and subtract shaft rotations, but cannot perform logic operations other than crude indications like positive or negative shaft rotations. It is the power of logic which gives computers their versatility.

The control unit contains logic circuitry which ensure that bit patterns are transformed and transported at the right moment. For example, when adding two numbers (represented by two bit patterns in the store) the numbers must be transmitted from the store to a register in the processor, not at the same moment, but successively. Control circuitry comprises many logic circuits called gates, and one or more clocks which provide timing pulses. In Fig 4, registers A and B can each store four bits. The transfer of the bits in register A to register B is controlled by four AND gates, the gates being open when the clock line is at logic 1 (usually 2.5V for TTL), and closed when the line is at logic 0. The carry column in Fig 3 shows the truth table for an AND gate. Note that the output is always 0 if input 2 is 0, and the same as input 1 when input 2 is 1. When the gates are open, register B is said to be enabled.

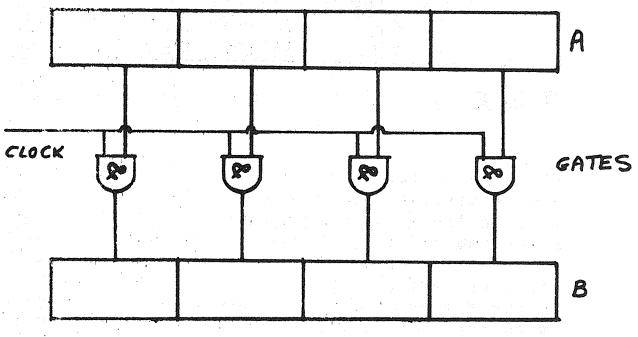


Figure 4. Example of register enable using AND gates.

Table 1
Binary equivalents of 0-9

Decimal	BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

3 Functions of a Computer

Computers are characterised by their ability to manipulate binary digits. Storage, transformation, transport and control are the four essential functions. Fig 1 shows a processor embodying transformation and control functions. The transport of binary digits from one sub-assembly to another (like the processor and the store) is accomplished by logic gates and a data highway. If the bits are transported one after the other, or serially, the highway consists of only one line. If the bits are transported one after the other, or in parallel, the highway consists of as many lines as there are bits. Modern computers use parallel highways.

Bit patterns have no intrinsic meaning - they are not data! It is only when a pattern is interpreted that it becomes meaningful. If eight bits, or a byte, are interpreted as a number, then this number may range from 0 to 255. A byte could be interpreted as a number which, instead of being stored in binary as in the previous example, could be stored in binary coded decimal. Only four bits are required here (for numbers 0 to 9) so two binary coded decimal numbers can be packed into one byte. Table 1 shows the equivalent of decimal numbers 0 to 9 in binary coded decimal.

Excess-3 code, which has certain advantages over straight binary coded decimal, is derived from the latter by adding three. Many electronic calculators store decimal digits using excess-3 code. The last column in Table 1 shows the excess-3 code equivalents of the numbers 0 to 9.

Instead of a number, a bit pattern may be interpreted as a character, classified as previously. Many different codes are in current use. Table 2 gives the ASCII code for 7-bit groups (the eigth bit could be used as a parity check).

b_7	b_6	b_5	b_4	b_3	b_2	b_1		0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
								0	1	2	3	4	5	6	7
0	0	0	0	0	0	0		(TC0) NUL	(TC7) DLE		0	@	P	`	p
0	0	0	0	1	1	1		(TC1) SOH	(DC1)	!	1	A	Q	a	q
0	0	0	1	0	2	2		(TC2) STX	(DC2)	"	2	B	R	b	r
0	0	0	1	1	3	3		(TC3) ETX	(DC3)	£	3	C	S	c	s
0	0	1	0	0	4	4		(TC4) EOT	(DC4) STO	$	4	D	T	d	t
0	0	1	0	1	5	5		(TC5) ENQ	(TC8) NAK	%	5	E	U	e	u
0	0	1	1	0	6	6		(TC6) ACK	(TC9) SYN	&	6	F	V	f	v
0	0	1	1	1	7	7		BEL	(TC10) ETB	'	7	G	W	g	w
0	1	0	0	0	8	8		(FE0) BS	CAN	(	8	H	X	h	x
0	1	0	0	1	9	9		(FE1) HT	EM	)	9	I	Y	i	y
0	1	0	1	0	A	10		(FE2) LF	SUB	*	:	J	Z	j	z
0	1	0	1	1	B	11		(FE3) VT	ESC	+	;	K	[	k	{
0	1	1	0	0	C	12		(FE4) FF	(IS4) FS	,	<	L	\	l	|
0	1	1	0	1	D	13		(FE5) CR	(IS3) GS	-	=	M	]	m	}
0	1	1	1	0	E	14		SO	(IS2) RS	.	>	N	↑	n	~
0	1	1	1	1	F	15		SI	(IS1) US	/	?	O	←	o	del

Table II. USASCII 7-bit communication code.

Fig 2 shows daigrammatically the logic associated with a bistable storage array. Notice that a bistable in the array is accessed by sending pulses along one column wire and one row wire. In other words, the column and row wires would have a 1 on one wire, and a 0 on the others. The bit patterns in the row and column registers could be regarded as numbers, which when combined by simple juxtaposition, form a number called the address of the accessed bistable. In practice, one register will act as a buffer for the address, and then storage address logic will decode the address into row and column addresses. Thus bit patterns can be interpreted as a number which 'names' the location of another bit pattern (by using groups of arrays with connected column and connected row wires). The addressing capability of a computer is one of its most important functions. When several bistable arrays are connected in parallel, access to a group of bits is called parallel access. Parallel data highways coupled with parallel access results in fast bit processing. In particular, the 6800 can access one byte from a bistable store in less than 350nS. Large computers have fast registers with access times of less than 40 nano-seconds. (It is easy to forget what these times mean; in one nano-second, light will travel about one foot in a vacuum). Fig 2 shows that ten bits are sufficient to access 1024 bits.

Since an address is a number, it may be combined arithmetically with another address. This is particularly useful in indirect or indexed addressing. Both computers and calculators use indirect addressing.

Two more interpretations commonly occur. Numbers and addresses can be combined into one word giving a reference to a group of words. For example,4/37 might refer to 4 words starting at location 37.

Lastly, a bit pattern may be interpreted as an instruction which, when decoded by the control unit, enables and/or disables various gates in an order determined by the processor design, such that bits are transported or transformed. An instruction, therefore, is the backbone of a computer program.

It should be realised that the interpretations discussed are determined by the computer program rather than by the control unit. The control unit can distinguish different bit patterns, but which interpretation is valid depends on the current instruction. Thus a computer program determines the overall functions of a computer. It can be said that the architecture of a computer is designed to be neutral, not having a bias toward any particular mode of operation. This is most important, it being the main difference between a computer and a programmable calculator.

Computer instructions can be manipulated in the same way as numbers because they are numbers. In particular, computers can transform groups of characters into instructions (both being sets of bit patterns). That is, the instructions are bit patterns which may be decoded directly by the control unit. For this reason, such instructions are called machine code. The process of transforming characters into machine code is known as assembly or compilation and is representative of an important part of computation which calculators cannot perform. The subject of program development and compilation is studied in its own right, several algorithmic languages, designed to facilitate program development, have been evolved and applied to computers. Fig 5 shows a program written in Algol (Algorithmic language) which would compute the overshoot of an underdamped shock-absorber.

```
overshoot
(real astart,astep,aend;
   proc peak = (real a) real:
      (real xt,xpeak:=0;
        for t from 0 to 100
        while real loop = t/100;
           xt:= 1-exp(-a*loop)*cos(loop);
           xpeak := xt
        do xpeak := xt od;
        xpeak);
   read ((astart,astep,aend));
   astart :=1; astep /:=astart; aend/:=astart;
   for a by astep to aend
   do print ((newline,"overshoot=",peal(a*astart))))
```

Fig 5 Second-order damping problem in Algol

Note; Thanks to the British Amateur Electronics Club for permission to republish this article from their newsletter. Details of the BAEC may be obtained from Mr. C Bogod 'Dickens',26 Forrest Rd.,Penarth,Glam.

AMATEUR COMPUTER CLUB NEWSLETTER
Vol 5 Iss 5 December 1977
Editor; Mike Lord
7 Dordells, Basildon, Essex
tel ; 0268 411125