7768 MICROCOMPUTER

The 7768 was designed to meet the need for a low cost, introductory, microcomputer which the beginner can easily build and use without having to buy expensive peripherals such as a teleprinter or VDU, but which can be expanded as the constructor's time and pocket permit to give a system comparable to any of the 8 bit microcomputer systems currently available.

In it's basic form, consisting only of a single board and a rudimentary control panel, the 7768 can be built for around £50 - or less if the constructor has a reasonably deep junk box, for care has been taken to use standard TTL packs wherever possible in preference to special, and hence expensive, MPU 'system' chips - and the only other piece of equipment needed to get the system up & running is a source of 5V DC at about 1 A. However, by using the popular 6800 MPU, and by providing for fully TTL buffered data and address busses, the system can be expanded without the constructor wishing he had started with a different CPU board, as can happen if the constructor starts with a very cheap MPU.
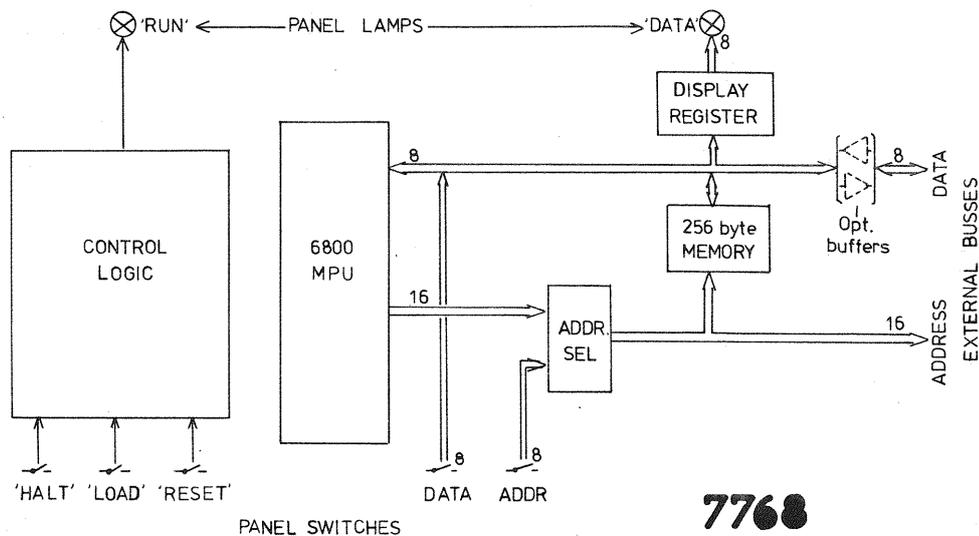
THE BASIC MACHINE

As shown in the block diagram above, the 7768 consists of;
- The 6800 MPU itself
- A 256 word 8 bit Random Access Memory used to hold data and instructions. Only the low order 8 address lines from the 6800 are recognised in the basic 7768, thus the high-order address byte of the 6800 instructions is ignored until extra memory is added to the system. 256 words correspond to (hexadecimal) addresses 00 to FF, however FF is reserved for the data switch/display registers.
- 8 bit data and address registers. If a MPU 'Write' operation references location FF, then the data is written into the register and displayed. Also, if a MPU 'Read' operation is performed from location FF, data is taken from the data switches rather than from the RAM.
- A 'Halt' switch that stops the MPU (after completion of the current instruction) then routes the 'Address' switches onto the address bus and holds the data register input gates open so the information on the data bus is displayed. Thus the contents of any memory location can be examined by setting the address switches to the appropriate pattern.
- A 'Load' switch. When operated, this loads the pattern set up on the data switches into the memory location determined by the address switches.
- A 'Reset' switch. Operation of this switch loads an internal 6800 register with the content of the data switches. This pattern is taken by the MPU as the address at which to start running when the 'Halt' condition is removed. If the 'Halt' condit-

ion was removed without the 'Reset' switch having been operated, the MPU would continue at the point in the program where it had been halted.
- Miscellaneous control logic.

Thus we can load a program and data into memory, run the program starting from a defined address, halt the MPU to examine or modify memory locations, and continue. While the program is running it can take in data from the switches and display the results.

CONSTRUCTION

This is straightforward, layout is not critical except that the Ø1,Ø2 drive circuitry should be kept close to the 6800. If the constructor can't get hold of 7428's, 7402's can usually be used for X5,6 (although spec limits would be slightly exceeded) if the 220 ohm res are changed to 1K.

Component and pin numbers are as the PC board available for the 7768 CPU. This is 8" square fibreglass with a gold-plated edge connector. To reduce cost a single-sided board has been designed so some additional wiring is necessary. £7.75 inc UK P&P from Mike Lord. This board is compatible with VERO VB/10725/1, 12681, 12682 & 13845/1 boards and 10859/4 77 way plus polarising key 0.1" socket.

X 15,16 are only needed for an expanded system.

X 11,12 could also be left off a basic system, but they are cheap and do protect the 6800 address op's against stray static charge.

The 22uF & 0.1uF capacitors should be distributed evenly across the board.

Mike Lord

7768   CPU

DATA LED DISPLAY

DATA SWITCHES

0V

+5V

'D7' 'D0'

←430

REGISTER OUTPUTS

from -X8,9 'Y' o/p

A0 — — A7

X7 &

256 SEL 5

+5V R4 430

R/W 4

X4

'LOAD' 0V 74

+5V R29 4k7

X10 7474 D Q C

E 6

'RUN' LED 71 +5V R31 270 X24

'HALT' 0V 75 +5V R30 4k7

X10 (7474) D Q C

CLK 10 X1

5MHz R1 430 R2 430 X1

X1 X1 X1

C12 33p

0V

C1-11

0V (GND) 5MHz HOLD

X14 DIN (X15 p1)

X19 SWSEL (X22 p10)

X19 X19

X3 X3 X3

MCE (X17 p13)

X19 X6 RSEL (X21 p13)

X14 X14

ADSEL (X9 p1)

X14

X14

X5 X6

X6 X6

X24 X4

X4

X5 X5

220 R16 220 R15 R18 22 φ2 37

R17 22 φ1 3

X5

R28 4k7 X24 R27 4k7 R13 430 R14 430

'RESET' 0V NMI IRQ

X13 (6800)

D7 26
D6 27
D5 28
D4 29
D3 30
D2 31
D1 32
D0 33

VCC +5V 8
VSS 0V 1,21

R/W 34
VMA 5
BA 7
HALT 2
DBE 36
φ2 37
φ1 3
TSC 39 0V
RESET 40
NMI 6
IRQ 4

A0 9
A1 10
A2 11
A3 12
A4 13
A5 14
A6 15
A7 16
A8 17
A9 18
A10 19
A11 20
A12 22
A13 23
A14 24
A15 25

R19 62 63 64 65 66 67 69 70

+5V

SWSEL

X22 (74125)

X23

R26 4k7

X21

Q1 Q1 Q2 Q2 Q3 Q3 Q4 Q4 (quad latch)
E D1 D2 D3 D4

Q1 Q1 Q2 Q2 Q3 Q3 Q4 Q4 (quad latch)
E D1 D2 D3 D4

X20 (74LS75)

RSEL

OPTIONAL DATA BUS BUFFERS

D7 38
D6 39
D5 40
D4 41
D3 42
D2 43
D1 44
D0 45

DATA BUS

MCE
MRW

R/W CE data (256×4 RAM) address X17 (2112)

R/W CE data (256×4 RAM) address X18 (2112)

+5V R5 4k7 R12

X9 (74157)

A1 B1 A2 B2 A3 B3 A4 B4 (quad 2 i/p sel) Y1 Y2 Y3 Y4
Sel G 0V

ADSEL

X8 (74157)

A1 B1 A2 B2 A3 B3 A4 B4 (quad 2 i/p sel) Y1 Y2 Y3 Y4
Sel G 0V

X12 (7432)

X11 (7432)

A0 21
A1 22
A2 23
A3 24
A4 25
A5 26
A6 27
A7 28
A8 29
A9 30
A10 31
A11 32
A12 33
A13 34
A14 35
A15 36

ADDRESS BUS

ADDRESS SWITCHES

'A7' 'A0'

0V

X2 (7495) SI C Qa Qb Qc Qd M A B C D

+5V R3 430

0V

2

DIN
MRW

X 15 (DM81LS97)   X 16

D0 — 46 D0
D1 — 45 D1
D2 — 44 D2
D3 — 43 D3

D4 — 42 D4
D5 — 41 D5
D6 — 40 D6
D7 — 39 D7

TO X13 (6800)   DATA BUS   TO X13 (6800)   DATA BUS

OPTIONAL DATA BUS BUFFERS

COMPONENT LIST - CPU BOARD

```
3 off 7400    ; X4,14,24
1 off 7402    ; X19
1 off 7404    ; X1
1 off 7410    ; X3
2 off 7428 *; X5,6
1 off 7430    ; X7
2 off 7432    ; X11,12
1 off 7474    ; X10
1 off 7495    ; X2
2 off 74125 ; X22,23
2 off 74157 ; X8,9
2 off 74 LS 75 ; X20,21
1 off 6800 CPU ; X13
2 off DM81LS97 ; X15,16
   (optional buffers)
2 off 2112 MOS RAM ; X17,18
1 off 5MHz crystal
20 off 4.7K resistors
6  off 430 ohm
1 off 270 ohm
2 off 220 ohm *
1 off 33pF capacitor
3 off 22uF 10V, 8 off 0.1uF
(* see text)
```

WB LIVES

'Bear Microcomputer Systems' is re-printing the Weeny Bitter designs (with, hopefully, the errors corrected and better looking circuit diagrams), a share of the profits (!) going to the ACC funds. £3.80 including P&P. Spare set of circuit diagrams also available (to drop solder & coffee on) 70p.

Future publications planned include the 7768, with much more information than in this issue, and the IOTA system. SAE for details to BMS 24 College Rd., Maidenhead, Berks, SL6 6BN

ACC 6800 LIBRARY

At a recent Committee Meeting it was decided that the ACC would purchase 6800 system software for members' use. It was decided that £50 could be spared at this time, with an additional £25 if & when the Club membership grows to 800. Details from Tim Moore.

The ACC will consider supporting other User Groups in a similar way, provided that the software can legally be copied & distributed, and that the ACC membership records show a reasonable level of support for the hardware needed to run the software.

LETTERS

SWTPC 6800

Am now building a 6800 system from the SWTPC kit - very good kit, well packaged. I decided I needed something that would work, as all my peripherals would be rather experimental. I have made the main part of the system (CPU board, mother board & control (serial) interface). Before starting on memory boards I have to ensure correct o/p of the MIKBUG program - at present I get an output to R035 TTY but not what it should be. So I'll have to get that right. The 'Starter' kit now includes 4K memory at no extra charge (was advertised as 2K only).

W A Hughes

A USEFUL BIT

I have made a discovery .
I had been trying to make an unsoldering tool (rather like one described in Byte) but it never got hot enough to melt any solder. While looking through my bits & pieces I realised that the earth pin from a 13-amp plug fits exactly between the rows of pins of 14 or 16 pin DIL IC's. If you file off the shoulder on one side so that it lies flat on the board, the terminal hole fits over the bit of the iron and the screw holds it in place.
It is very simple but it worked atraight away (I was using a 30W element)

Bill Hughes

POINTS ARISING ?

As many of the members of the ACC will know, I advertised some items for sale in a previous ACC newsletter. The phone has only just stopped ringing and as always, first come first served, but after about 50 (FIFTY !) phone calls (some of them up to an hour long !) a number of points arose.

The magazine BYTE which I subscribe to was the one that cropped up again and again, and people either didn't know it at all or couldn't find the right person to send to for subscription. The person I sent to was;
       John Remizo
       142 Grove Lane
       Hale, Altrincham, Cheshire
The sub was £10 for 12 issues (I believe it has risen since then .. ed)

Another point was my choice of CPU chips, and here people were interested in the various aspects of different MPU's.
My views are now as follows; The 8008 and 8080 which I have both used, and the 6800 which I now use (discussed later) all have their merits, and some aspects of choice will relate to other MPU's.

The 8080 series is an excellent set and is an Industry standard. Further, the Z80 is software compatible with the added advantage that it has another 70 or so instructions (some very juicy!) The problem with the 8080 comes with the software. First, the monitor program for the 8080 is not an easily obtained item since 8080 users mostly design their own and have them 'burnt in' to their own ROM's. Should anyone require one there is a good listing in a Byte issue last year. This was designed by AMSAT for 8080 users and has a lot of good functions in it. The other problem with software is that if you require BASIC or EDITOR or assembler then you are going to pay a lot of money for them, and even in the USA hobby market the software either comes with your MPU kit or you pay! There is now one alternative to this and this is a book called SCELBAL. This is a USA publication and costs $49, but it is a 'cookbook' of listings and ideas to produce your own BASIC type compiler to your own designs.

The 6800 series is also a good standard and the

instruction set is quite similar with the added
advantage of some relative branching instructions
which make it easy to generate fully relocatable
code. The monitor for the 6800 is the well known
MIKBUG, and it is easy to obtain and very satis-
factory to use (there are rumours of a new version
becoming available). Software for the 6800 is
easily available and cheap. Computer Workshop has
3 versions of BASIC all less than £15 on either
paper tape or cassette. The 6800, as is evident from
our own collection of 6800 users, has become the
hobbyist's standard with the 8080,Z80 etc being a
good second for the purists writing their own
software.

Possibly as a guide to you potential 'Cybernuts';
Before you embark on your MPU's, decide what your
ultimate aims and desires are, and if you think that
once constructed you will become impatient to run
programs on your MPU, then consider a system with
at least a monitor available. The result if you don't
is 1 year's construction and 6 months' very frust-
rating programming! I know, I did it !
My system is now a 16K MP68 on which I run BASIC
and shortly I will have an EDITOR/ASSEMBLER. I use
a cassette interface that was published in MICROTREK
and needs no adjustment at all to run Kansas City
Standard tapes. I finished the ETI VDU but the mods
that appeared in ACCN didn't work (not on mine any-
way) so we are going to find another wallet to burn
a hole in and get a VDU. Anyone in the area that
wants to view my system, play Startrek, etc., is
welcome, but please ring me first.

Constructors note; A UART I have been using rec-
ently is the Intersil IM6402. This is pin compatible
with other UART's but is CMOS and is single power
supply 4 - 12V and consumption is less than 10mW.
Another version is 6403. This has a built in XTAL
oscillator as well. It is available from Rapid
Recall about £4.32 + VAT.

D V Goadby   2 Lupin Close, Hinckley,Leics
              tel;0455 35621 (evenings, weekends)

WB BITS & BOOKS

All those reports of working WB1's are making me
jealous, as impending exams prevent me from
finishing the final wiring up. However, I am recom-
pensed to some extent in being able to use a PDP10
(batch) and several PDP8's (assembler and machine
code).
I like the look of R H Stopford's expansion
logic. The lack of some larger memory was pushing
me to look at a uP system.
An article in EDN (Feb 5 '77, p 97) on the SC/MP
keyboard looks interesting and shows some detailed
circuitry which looks fairly easy to modify for
WB. With the addition of some ideas from some of
the OPUS1 (TM221) information, in diode ROM, and
an ETI VDU, a primitive interactive system should
be realisable.
A further modification that I would like to see
to the VDU, would be to only allow information to
be written into the bottom line and use 'rollover'.
I should think that some logic could be devised to
read (upon carriage return) the 2nd line, transfer
this line's information to the first, then read
line 3, transfer to line 2, and so on down the page.
Also, the top line's rightmost boxes could be
permanently wired to the PC of the computer, to
give an indication of where the PC is pointing.

2 interesting books;
'The Moscow Puzzles' Boris A Kordensky £1.25
Pelican. Some are suitable for using a computer to
solve. One of the classic mathematical/logic puzzle
books. About 300 pages including solutions.
'Theory and Design of Digital Computers' D Lewin
£5.50 Nelson. About 375 pages including some worked
examples. Midway between just-leaving-basic-logic
and so-complicated-that-you-can't-even-read-the-
title book.
A useful book for those for whom the initial 'Ah,
pretty lights!' syndrome is starting to fail and
for those who feel that they would like to get
deeper into the subject (none of us has to admit to
still liking pretty lights, do we ?)
                                          W G Beer

4

HELP

RO 35 MODS

I understand that a large number of members bought
the RO/35 printers that Chiltmead were selling as
ASCII coded. I have seen several of these units and
none of them had a code anything like ASCII.
I did, however, succeed in converting mine to a
fair approximation and feel that other members may
be interested.
It is a rather simple procedure but too verbose to
produce here, so if anyone is interested I can let
them have a look at my unit and give them a copy of
the notes I made.
Please ring me at home on 01 751 2872 (6 - 7pm)
                                          Bob Selby

MORE HOLES WANTED

I would like to get in touch with anyone in the
Essex area who could show me a Creed model 25 8-
hole paper tape punch; I am interested to discover
the **detail** differences between the 5/7-hole and the
8-hole versions so as to decide whether it is feas-
ible to modify.
P.C.Weedon, 19 Helmsdale Close, Rise Park,Romford,
Essex RM1 4RS

WOT'S TREK ?

There has been a lot of talk in ACCN (and other
magazines) about Star Trek, but nobody ever defines
the rules or even the basic concepts. Would it be
possible to have a suitable article in ACCN ?
                                          K Horton
*** Please , ed ***

EL MUCHO CHEAPO ??

I am trying to get together a computer system but,
due to a restricted budget, cannot afford all the
parts I require. If anyone has any **cheap** bits and
pieces that they do not require could they please
let me know. Anything considered, I would collect
if necessary.
M McLean 19 Haldane Ave, Haddington, E Lothian.

WILL HELP

I would welcome the opportunity to program an
existing micro computer or to help somebody on the
programming side in exchange for tips and help on
the constructional side. My current knowledge on
practical electronics is virtually nil - and only
slightly more on the theory ! But I have been prog-
ramming on & off since the late fifties. Micros
are not that dissimilar to the first generation
mainframes.
I would like to meet anybody who might be inter-
ested in this arrangement.
T F Hobson 44 Milton Ave., Eaton Ford,Huntingdon,
Cambs PE19 3LE tel; Huntingdon 74090

WHO HAS THE KEY ?

I have recently purchased a keyboard with 84 keys
from a VDU. The key switches are open type with a
bar which shorts out two contacts on a PCB below.
The coding on the PCB is ICL 83607712 Class No.
5068 D6668 and I should be grateful if anyone
could let me have any information on this PCB as I
cannot trace it out without stripping off the pro-
tective covering.
I should also be very interested to hear if anyone
has tried to use the matrix on this PCB for the
first stage of ASCCC coding.
Keith Roberts 56 Victoria Ave.,Grappenhall,
Warrington WA4 2PD

MPU - TYPEWRITER LINK ?

A few of us are interested in the possibility of
hooking an electric typewriter (any sort) to a MPU
(any sort). Can anyone direct us to anything publ-
ished, has anyone actually managed this, could you
persuade someone to write it up in the ACCN ?
D B Johnson 9 Uplands Rd., Bromborough, Wirral,
Merseyside L62 2BY

FAILED FRIDEN

I have recently come across a 7 bit Friden Flexowriter but for some reason I cannot find out how to trigger the Read, Write and Sense circuitry (even the light does not go on !). I presume that as there is no local/line facility there must be some connection somewhere to acuate the whole system. If anyone has any suggestions or ideas about where I could get information I would very much like to hear from them.
Simon Garth c/o The Babbage Society,Bedford School, Bedford.

CORE

Do you know of any member I can get in touch with for information about Core memories ? Poly Paks of PO Box 942 South Lynnfield, Mass USA are advertising Burroughs 2600 bit core planes at $8 a piece. I am interested in following this up for memory, but lack detailed information on how to mate the core plane up to a computer memory & address bus etc. If any of the more electrically oriented members could help I would be very grateful.
J B Huntley 77 Emolweni Rd, Kloof, Natal 3600, S.A.

CALCULATOR HEX

Texas inform me that the SR-22 has been discontinued. As far as I know it was the only Hex calculator (also Octal & Dec.) available, and as I have been given the go-ahead to find out costs with a view to buying one for the office, it is a chance I don't intend to miss if I can avoid it. I used to use a SR-22 when I worked for Texas, & it was a good thing! So, does anybody have one for sale, or know of another Hex calculator on the market ?
G Hayes 42 North View Cresc.,Epsom Downs, Surrey

COMPUTER COMICS

Following the note about Byte subscriptions in the last issue, the general opinion seems to be that the cheapest & simplest way is to subscribe directly using Barclaycard or Access (or a bank draft, which is slightly more complicated). Be prepared to wait 6 - 8 weeks for any response, and accept that your copies are going to arrive 4 - 6 weeks late.

KILO BAUD (a fairly new magazine similar in content to Byte) is available via J Coote 56 Dinsdale Ave., Wallsend, Tyne & Wear NE28 9JD for £10 for a year's subscription or £16.50 for 2 years, which seems good value at today's exchange rates. Mr.J Coote also handles subscriptions to 73 magazine, for the benefit of radio enthusiasts.

PERSONAL COMPUTING is now being distributed by L.P. Enterprises, as you will see from the enclosed form, at the ridiculously low price of £5.75 per year. This magazine, now including MIKRO-TREK, is aimed at using computers, rather than making them, and the issue I have seen is very imaginative. L.P.Enterprises are also going to import some other US magazines, possibly Dr. Dobbs Journal, details later.

mike lord

WB PLUS

Mr R.H.Stopford has offered to co-ordinate the design of a WB-2, although because of other commitments he will not be able to start in earnest until after June. Contact him at;
14 Uplands Rd., Flixton, Manchester

𝔖𝔥𝔬𝔭

St Alban's Computer Market

Rumour has it that there is a stall in St Albans Saturday market with lots of ex computer bits, boards, tape drives, PSU's etc.

FOR SALE

M6800 family 128 byte, 575 nSec static RAM's MC6810L1. I have 30 of these, surplus to my requirements. Completely unused never removed from packing. £2.50 each.
Alan Hill Top Flat, 19 Brook St, Bath,Avon

FOR SALE

Ampex TM4 tape deck. ½" tape, 19" rack mounting. With most of the control electronics and power supplies. Buyer collects £ 25 ono.
ICL card verifier mechanism. Works OK. Buyer collects £5 ono.
Oil filled transformer, suitable core store PSU, 35 - 0 - 35 V @ approx 8A. Buyer collects £5.
Ring; R Selby 01 751 2872 between 6pm & 7pm only.

WANTED URGENTLY

Fortran IV for PDP8, to run under OS/8 V3 with a minimum of 8k of core.
R Selby 01 751 2872 (6 - 7pm)

FOR SALE

Ampex 7 track TM-7 tape deck with all read/write electronics and ICL 1900 interface, in very good condition. ½" tape. 36"/Sec. 200/556 BPI.
Ian Leslie
4 Hampstead Court,Grovelands Rd.,Reading,Berks

FOR SALE

Eleven core matrix planes. Mullard type MA74794 Each plane 32 x 32 + inhibit & sense wires. No technical data. £10 + P&P
R A Munt
51 Beechwood Drive,Feniscowles,Blackburn,Lancs.

FOR SALE

One GT6144 graphics board, built and tested but now surplus to requirements. Gives 6K addressable spots on TV screen, using single 8 bit port.£65.

Set of ETI VDU boards built & tested as per ETI with 2 extra memory chips for 16 lines (not fitted) £25

Two NCR key to tape encoders (similar to Honywell units being sold by Chiltmead). One complete, one in major units (taken apart to fit in garage) but complete. These have excellent tape decks and keyboards, there is even a small core store inside. The pair, delivered by arrangement, £70

D V Goadby Hinckley (0455) 35621 evenings / w'ends

I have large quantities of the following parts at low prices for ACC members.All prices are VAT inclusive, just add 30p for postage.

Motorola MC6800L Ceramic......................£18.75
Signetics 2102L-1 500nS RAM....................£1.75
Fairchild 9368 Latch/hex decoder/LED driver....£2.50
Intel 1702A 256x8 EPROM (Blank)................£5.75
Standard Microsystems 2502 UART................£6.75
National 81LS97 Octal Tristate Buffers.........£1.15
National 81LS98 Octal Tristate Buffers.........£1.15

Fairchild 1AMP regulators,set by two resistors,both in TO202 package:
78G (5v to 30v),79G (-3v to -30v).......each...£1.05

LoProfile Sockets: 8,14,16pin...................11p
                   24,28,40pin..................35p

741C OpAmp (8dip)...............................17p

74367 Hex TriState Buffer.......................65p

74LS20,74LS21,74LS27,74LS30,74LS32 all at.......20p
74LS74,74LS86.............................at.....30p
74LS164...90p

All parts are prime quality, money back if not satisfied.A basic pinout datasheet is available for each device at 10p.

John Hawthorne,23 Iver Lane,Cowley,Middx,UB8 2JD
Phone Uxbridge 36428 evenings only.

MINE

IN PRAISE OF THE 2650

Having followed with interest the activities of ACC members over the last year or so, and having seen the emphasis move from WBL to microprocessors, I have been surprised by the lack of comment on the Signetics / Mullard 2650.

I would have thought that many members' requirements were similar to my own, i.e. to get as much home computing power as possible for the smallest financial outlay.

To this end speed is of secondary importance to simplicity of interfacing and single supply operation on the hardware side.

On the software side, it helps if the micro has multiple accumulators and powerful instructions particularly with indexing and indirect operations. This allows any given program function to run in the minimum of memory space. This is surely of great importance to an ACC member because after the initial splurge of getting a given micro hooked up with power, peripheral chips and minimum memory, the member will be getting his first programs going. After a few weeks, as programming ability improves, the programs will be getting more sophisticated and the first 1k bytes (say) will be overflowing. Very few members are likely to be able to afford the full complement of memory that the micro allows. Even so the cost of memory is likely to be the most expensive single section of a home computer.

Returning to the 2650, this micro costs approx £18 in one-offs and £13 each for 25 off. It has 7 accumulators (as different to registers) consequentally partial results may be computed in one accumulator, and left there, while another partial result may be computed in another accumulator.

Hence up to seven partial results may be computed without the time and space requirements that other common micro's (8080,6800,6502,SC/MP etc.)require.

The 2650 is fully TTL/CMOS compatible, even for its single phase clock. It also requires a single 5V supply.

In terms of instruction execution speed the 2650 is near the top, being much faster than SC/MP, similar to the 6502,6800 but slower than an 8080. The instructions, taken as a set, are more powerful than some minicomputer sets.

The 2650, unlike most other micros, does not appear to have been designed as a 'family' chip. This, at first sight, may appear as a disadvantage but because of this all signals are fully timed 'on chip'. This makes the design of memory and peripheral circuits very straightforward with almost any manufacturer's chips. For example chips from the Intel 8080 family and the Motorola 6800 family may be mixed into a complete system along with, say, memory or other chips from any Tom, Dick or Harry.

To quote a particular example of a timing relationship, on the 6800 it is a requirement that an interrupt request does not occur at a certain point in the clock cycle. This is quite easy to get round by adding chips, but the 2650 has no such 'hang-ups!

Signetics/Mullard are, however, to add some 2650 family chips shortly. The first two are ;
- a three parallel port / timer / 8 bit serial in-out port.
- a synchronous serial chip, with many options for use both for serial communications and floppy disk/cassette tape use.

Although I have yet to finish my home computer system, I have used the 2650 and can thoroughly recommend it after reading the literature for just about every 8 and 16 bit microcomputer on the market.

R A Munt

LANCASTER UNIVERSITY AMATEUR RADIO SOCIETY
NORTH WEST AMATEUR RADIO CONVENTION 1977
        17 - 18 September 1977
Details from J.R.Morris, Dept of Physics,
Lancaster University, Lancaster.

LOCAL GROUPS

SOUTHAMPTON UNIVERSITY AMATEUR COMPUTER CLUB

SUACC gained its status as a club in Autumn 1976. Its membership is now approximately 40, mostly 'engineers' and 'mathematicians' with an apparent wide range of interests.

We do not have regular meetings, as yet, but over the term we have held several events, mostly talks.

Our first major talk was given by Ron Mount, of the ACC, on his 'Weeny Bitter'. A demonstration of the machine was given, using a fairly standard 'flashing light' routine - a pseudo random sequence generator. The talk was livened up by various humorous comments such as, on the subject of a hefty power supply and the requirements of TTL "What's a few amps between friends?".

Other events were a trip to an IEE microprocessor lecture at Brighton, and a talk/demonstration by one of our members, Tom Gardener, of his home built 6800 machine. This consists of several large PC's and a large front panel/console with LED's and switches for I/O, but it works and is reasonably cheap, both of which are crucial factors in an amateur's design philosophy.

The club has some time on the University 1907 computer, and it is hoped that we will be able to arrange a club programm & project.

P D Maddison

CAMBRIDGE UNIVERSITY PROCESSOR GROUP

is a club for those interested in building, programming & learning about small computers. Members have micros & mini systems at various stages of completion. Weekly informal meetings plus lectures by guest speakers & a termly magazine. Non-University welcome.

Contact Emrys Williams (Cavendish Lab, Downing College) Chris Daykin (Sidney Sussex College) or come to a meeting (posters in college)

VDU THOUGHTS

IDEAS FOR VDU   R Selby

In the article by Mr Howland, he gave a circuit for control of the cursor. It ocurred to me that it would be fairly easy to arrange for the screen to scroll upwards on receiving a LF character.
    This can be done as explained below;

1) We want the cursor to always be on the last line (line 7).

2) The display has line 0 at the top of the screen. Therefore if we take the current scan address 'A' and add to it the 1's complement of the write address 'B' produced by Mr Howland's circuit, and use this as a read address for the display, the data will appear to shift up one line per LF character. e.g. if 'A' = 0 (top line of display) and if 'B' = 0, then the answer = 7, so the incoming data will be written to line 0 in the display but displayed on line 7 on the screen.

Similarly if 'A' = 7 (bottom line of the display) and 'B' = 0 then data will still be written to line 0 in the RAM, but line 7 in the RAM will be seen on line 0 of the display.

If 'A' = 0 and 'B' = 1 then new data will be written to line 1 in RAM, displayed on line 7, while the old contents of 7 are displayed on 6.

An idea for the circuit is shown as Fig 1, but I have not tried it so don't take it as gospel !

A minor problem exists in that on receiving a LF character the text on the top line appears on the bottom line of the screen. This can be overcome if it proves embarrassing by writing space or delete from the current cursor to end of line, an idea for a circuit is given as Fig 2.

scan address 'A'

ETI
IC 19

To IC's 22,
23,24,25 on
ETI cct

write address 'B'
(goes to IC's
20-21 on ETC cct)

74193 on Mr Howland's
circuit
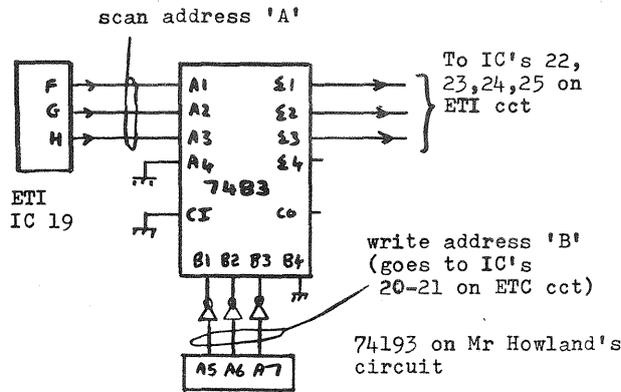
Fig 1   n.b. the links between IC19 and IC's 22,23,
24,25 should be cut before this cct is
added

set write data = null

$\overline{WRITE}$ to RAM's
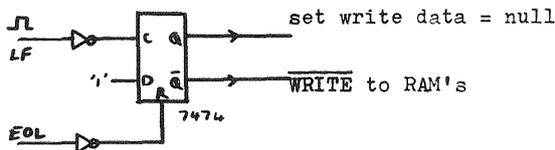
Fig 2

MORE DIVISION   A Russell

The accompanying flow-chart is based on the method of binary long division expounded in O F Morgan's article in ACCN Vol 4 Iss 5 p 4

The method, applying to unsigned numbers, is compact and simple and, moreover, appears to be readily expandable to multiple word operations.

Two points need to be noted;

i.   The term 'overflow' in step 4 refers to CARRY; the OVERFLOW condition is irrelevant (and could only occur on the first execution of step 3 anyway).

ii. It may not be clear from the text of step 5 that this step also is included in the iterative loop, although the exit point is after step 4 when, immediately prior to re-entry in step 3 of the loop, the divisor has been shifted back to its original position.
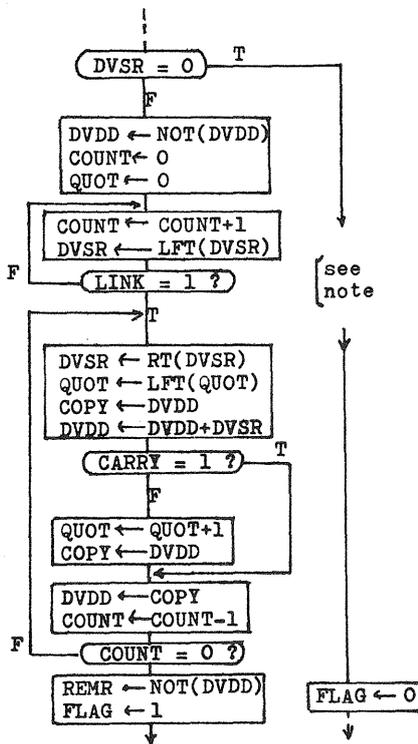
GLOSSARY

DVSR : DIVISOR
DVDD : DIVIDEND
COUNT: SHIFT CTR
QUOT : QUOTIENT
COPY : WORK SPACE
REMR : REMAINDER
FLAG : ROUTE IND-
       ICATOR FOR
       REFERENCE

FUNCTIONS

NOT : 1's comp
LFT : Shift left
      one place
RT : Shift right
      one place

Note; LFT must shift into a link bit & pick up 0's from the right. RT must shift the same link bit into bit 7 (or other MS bit according to word size)

There are further points to be noted about the WB code given;

It might be thought that, since the method is applicable only to unsigned numbers, it is legitimate to use the entire register length for the numerical value of, in particular, the divisor. The alignment routine will effectively discard the MS digit and, if this happens to be the only digit, the program will wander off into orbit.

I appreciate, of course, that the program is undoubtedly intended for use with unsigned values derived from a signed system and that all the MS digits will therefore be zeroes. Those who may not have realised this are now warned.

Further to this, the count control is faulty. Lines 15,16 of the WB code together contrive that, when DENOM is at last returned to its original position, the count is zeroed. Line 17 then prevents DENOM from being 'tried' in this position.
Since QUOT will already have been shifted left, its rightmost digit will always be zero at termination.
Thus, say, 112 / 32 will be returned as 2, remainder 48. (The numbers are chosen to make a desk check fairly short). Similar results will be obtained whenever the correct quotient would be an odd number.
A simple solution to the problem is to substitute GPL for GNZ in line 17.

Now for points about the flow-chart;

It is left open at entry and exit for adaptation to any of subroutine (in which case the two branches can enjoy a common RETURN), routine or stand-alone program.
The constraints placed on the shift functions are pretty usual and most processors can implement them immediately.
The flag indicator is not strictly necessary, since the status of the result can always be afterwards ascertained from the zero status of the divisor. However, as coded for the M6800, this flag costs nothing, sheer spin-off, you might say, and its use saves time and space for a separate testing of the Divisor. It has, after all, been tested once, so why do it again ?
Whether such a manouver is available in WB code, I don't know, as I am not familiar with it.

I am intrigued by the reference to speed of operation in the article. Certainly, long division is much faster than successive subtraction, but the program is comparatively very much longer.
The question we have to answer is, as always, what do we do with what we save ?
In terms of program bytes occupying valuable floor space, the answer is pretty obvious to micro-system users. In terms of real-time, however, the answer is often that the processor spends that much more time idling.
There seems to be little point in saving idle time at the expense of core-space, especially if the time saved can be measured in milliseconds per routine usage.
In the worst case, successive subtraction in 8 bit operation takes 255 iterative loops plus any prelim and close-out. In the M6800 coded program, this amounts to 3338 machine cycles. The long division, in the worst instance, takes 452.
At 1MHz, we gain 2.89 milliseconds with each use of the routine or one second for every 346 long divisions (supposing that each one would have occupied the maximum time on successive subtraction).
In 16 bit operation, the case is slightly altered. This is because, whereas in long division, doubling the length of the word doubles the number of loops (just about), in successive approximation, the number of loops is squared (at maximum). The 452 becomes something less than 904 but the 3338 becomes about 850 K. The saving is now a second a go. Even so, in many applications, this may not save any non idle time. The process may only use the routine on occasion and have plenty of idle time to mop up.

In 32 bit operation (and why not, pray ? The WB-32 may be just around the corner, or, at any rate, around the long slow bend), the maximum length successive subtraction will take something like

7

$15\frac{1}{4}$ hours and, naturally, long division takes on a profitable glow, at around 2mS.

Methods of performing extended arithmetic by software have been known to contain 'funnies' and total reliance on unsupported theory is hazardous (I am using the term in its loose sense. Theory is support, of course). I therefore append a reasonable validation of the Morgan method.

LONG DIVISION

			effect on CARRY	program bytes Min	Max	note
	TST	DVSR	0 *	3	3	
	BEQ	RTS	-	2	2	(ii)
	LDA B	DVDD	-	2	3	
	COM B		1	1	1	
	CLR	COUNT	0	3	3	
	CLR	QUOT	0	3	3	
INC	INC	COUNT	-	3	3	
	ASL	DVSR	Bit 7*	3	3	
	BCC	INC	-	2	2	(i)
ROR	ROR	DVSR	Bit 0*	3	3	
	ASL	QUOT	Bit 7*	3	3	
	TBA		-	1	1	
	ADD A	DVSR	C *	2	3	
	BCS	CLC	-	2	2	
	INC	QUOT	-	3	3	
	TAB		-	1	1	
CLC	CLC		0 *	1	1	
	DEC	COUNT	-	3	3	
	BNE	ROR	-	2	2	
	COM A		1 *	1	1	(ii)
RTS	RTS			1	1	
				45	47	(iii)

Effect on CARRY

0 : Execution clears CARRY
1 : " sets "
C : " " " if the sum is larger than the register, clears CARRY otherwise.
Bit n: Execution shifts Bit n of the register into CARRY.
- : Execution maintains CARRY in its previous state.
* : Crucial manipulations of CARRY in this application.

Notes;

(i) The MP6800 MPU uses the CARRY as a LINK bit.
(ii) The state of CARRY at entry point RTS is designed to conform to the flag indicator requirements. No seperate flag is necessary.
(iii) The number of program bytes will depend on the number of variables allocated locations in the range 0 - 255, since these are addressable in Direct mode. In this application the advantage is marginal.

SUCCESSIVE SUBTRACTION

			effect on CARRY	program bytes Min	Max	note
	TST	DVSR	0	3	3	
	BEQ	RTS	-	2	2	
	LDA A	DVDD	-	2	3	
	LDA B	#$FF	-	2	2	
INC	INC B		-	1	1	
	SUB A	DVSR	C	2	3	(i)
	BCC	INC	-	2	2	
	ADD A	DVSR	C	2	3	(i)
RTS	RTS			1	1	
				17	20	(ii)

Effect on CARRY;

as before

Notes;

(i) When ADD A DVSR is accessed as an instruction, SUB A DVSR has necessarily set the CARRY, in which case ADD A DVSR (given no intermediate interference with the values in either A or DVSR) will also necessarily set the CARRY. (A proof of this is appended).
Once again CARRY will act as a flag indicator as to whether a division has been carried out or whether not, by reason of the Divisor being zero.
(ii) See note (iii) of the previous notes.

Proof;

Where $X7 =$ bit 7 of the ACCUMULATOR contents at start.
$M7 =$ " " " " MEM LOC " " "
$R7 =$ " " " " ACCUMULATOR " " end.

In executing SUB
$$CARRY = \overline{X7}\,M7 + M7\,R7 + R7\,\overline{X7} \quad \text{(Boolean)}$$

In executing ADD
$$CARRY = X7\,M7 + M7\,\overline{R7} + \overline{R7}\,X7 \quad (\text{ " })$$

Given the conditions referred to in note (i) above, the X in ADD is the R in SUB, and the R in ADD is the X in SUB. If the substitutions are made in the expression appropriate to ADD, it becomes
$$CARRY = R7\,M7 + M7\,\overline{X7} + \overline{X7}\,R7 \qquad QED$$

VALIDATION OF MORGAN'S METHOD

Beginning with the obvious; X cannot be a fractional multiple of the LS digit of the register in which it is held.

Continuing with the obvious;
$$NOT(X) + X = 2^r - 1,$$ where $NOT(X)$ is the complement of X in a register of length r.
$$NOT(X) = 2^r - X - 1 \quad \text{(unsigned value)}$$

Staying for a moment longer with the obvious;
$$\Rightarrow X = M.D + R ,$$ where M is the integer quotient of $X \div M$ and R the remainder.

By substitution, then
$$NOT(X) = 2^r - M.D - R - 1$$

Forbearing with the obvious yet again;
D, shifted left by S places $= D.2^S$

Adding this to NOT(X) gives
$$2^r + (2^S - M).D - R - 1$$

If and only if, this sum exceeds $2^r - 1$, the CARRY will be set.
$$2^r + (2^S - M).D - R - 1 > 2^r - 1$$
$$\Leftrightarrow (2^S - M).D - R > 0$$
$$\Rightarrow (2^S - M).D > 0 \quad (R_{min} = 0)$$
$$\Rightarrow 2^S - M > 0 \quad \text{provided that D is +ve}$$
$$\Rightarrow 2^S > M ,$$ in which case 2^S is not one of the digits of the quotient M.

On the other hand
$$2^r + (2^S - M).D - R - 1 \leqslant 2^r - 1$$
$$\Leftrightarrow (2^S - M).D - R \leqslant 0$$
$$\Rightarrow (2^S - M).D \leqslant R < D \quad (R \text{ is the remainder})$$
$$\Rightarrow (2^S - M) < 1 \quad \text{provided that D is +ve}$$
$$\Leftrightarrow M - 2^S > -1$$
$$\Leftrightarrow M - 2^S \geqslant 0 \quad$$ since $M - 2^S$ cannot take a value between -1 and 0.
$$\Rightarrow M \geqslant 2^S \quad$$ so that, if S is the greatest value for which the forgoing is true, 2^S is the leading digit of M.

At this point in the algorithm, the residual is substituted for the original complement. This has now to be further dealt with.

Re-writing $2^r + (2^S - M).D - R - 1$
as $2^r - N.D - R - 1$ (where $N = M - 2^S$) the maximum value of t is found such that
$$N \geqslant 2^t$$
so that 2^t is the next digit of M.

And so on.

Each time a component digit of M is identified by reference to the CARRY condition, a 1 is added to the - originally zero - quotient. By the time the divisor has been de-shifted to its original position the 1's will have been shifted to their correct binary position in M.

This continues until the residual is necessarily
$$2^r - 0.D - R - 1 \quad (= 2^r - R - 1),$$ the complement of which is
$$2^r - (2^r - R - 1) - 1 \quad = R,$$ the remainder

A Russell

PASCAL

The Language PASCAL   R H Kirkby

Intro:
       I am an undergraduate at Southampton Univ.,
where last year I took a unit course in advanced
programming. The working language was PASCAL, the
best language I have met yet, although new to me at
the time. I would like to tell you something of it.

History:
       Prof. Nicklaus Wirth has worked on high level
languages and their compilers for some time. Although
originally involved with Algol 60, he in 1965 left
the mainstream of thought on Algol's future - which
was leading to Algol 68, Algol X and Algol Y.   He
instead produced Algol W, which has anjoyed some
popularity. Around 1972 he published definition
PASCAL and a compiler for it. Among his latest work
is a parallel-processing version of it (called
"MODULA").

The Language:
       Designed to give source code that is natural
to write, reliable once written, clear in meaning
("transparent") and easy to compile to efficient
object code, PASCAL superficially resembles Algol
60 but with emaciated syntax. The most significant
difference from all other languages is the facility
to structure data (much as a program is structured)
and so I will confine myself to this.
   Here is a hypothetical payroll program's declar-
ation part, designed to work upon a small firm's
records of one week.
(*THIS IS A PASCAL COMMENT*)

CONST (* for 'constant'*)
      NOOFEMPLOYEES = 147
       (* defines the size of the firm's workforce,
          can be adjusted in future by altering
          this one statement *)
TYPE (* declarations of structural types follow *)
      DAYSOFWEEK = (MON,TUE,WED,THUR,FRI,SAT,SUN);
       (* so any scalar variable of type daysofweek
          takes one of the seven (ordered) values 'mon'
          to 'sun'. Note that arithmetic upon such
          values is meaningless, despite the existence
          of ordering *)
      WORKSNUMBERS = 1..NOOFEMPLOYEES;
       (* variables of this type take integer values
          only between 1 and 147 inclusive. So trying
          to give such a variable values 0, 243 or
          'tue' are all mistakes *)
      ATIME = RECORD HRS : 0..23;
                      MINS : 0..59 END;
       (* variables of this type cannot take values
          which couldn't be on a 24 hr clock *)

      DAYSWORK = RECORD BEGINMORNING, ENDMORNING,
          BEGINAFTERNOON, ENDAFTERNOON : ATIME END;
      ACLOCKCARD = ARRAY [DAYSOFWEEK] OF DAYSWORK1
      AWEEKSWORK = ARRAY [WORKNUMBERS] OF ACLOCKCARD;
VAR (*having described the data structure shapes
        needed, the actual variables are declared *)
   THISWEEKSWORK : AWEEKSWORK;
   CARD   : ACLOCKCARD;
   THEDAYSHOURS : DAYSWORK; TIME : ATIME;
   EMPLOYEE   : WORKSNUMBERS;
   TODAY   : DAYSOFWEEK;
   (* these six are the structured data variables,
     the main variable being 'thisweekswork' and all
     the substructure variables being provided to
     manipulate it's contents more easily *)

   Now consider the assignment statement

   TIME := THEWEEKSWORK [89][TUE].ENDAFTERNOON;

   somewhere inside the program body. This puts the
time that employee number 89 knocked off on tuesday
afternoon into the variable 'time'. Analysing the
structure of the statement;
   The data structure 'theweekswork' contains the
information of every clock card punching that occ-
urred in this particular week. It is an array. It's
element 'theweekswork [89] ' contains the data from
employee number 89's clock card this week. It too is
an array.

It's element 'theweekswork [89] [tue]' contains the
coding of this card's Tuesday column. It is a record.
It's field 'theweekswork [89][tue].endafternoon'
contains the data from the fourth entry upon this
column of the card. It is a record of the type 'atime'
as is the variable 'time'. The statement is acceptable
to a compiler because they are of the same type.
   Although the example is seen to be complicated, it
is seen to be easily analysed (due to the simple
syntax of the language).

Pro's & Cons
   Good features:

1) The structure of the language mirrors the way one
   thinks (unless one thinks in FORTRAN) and with it's
   simple syntax is easy to learn and produces code
   of stunning clarity - further facilitated by jud-
   icious use of mnemonic names and concise statements.
2) Every new identifier has to be declared. With the
   data structures available this produces code which
   is highly reliable, since a semantic error is also
   likely to be a syntactical error, so the compiler
   can trap it.
3) Fast compilation.
4) Efficient object code (usually better that FORTRAN,
   so if you want better, try machine code. If you are
   good you might get it)

   Bad features:

1) It is a bind thinking up new yet apt names for all
   the items in a big program.
2) The language is a bit new, so it is not yet well
   known, and some implementations are a trifle crude.
   Others are brilliant.

Booklist:

       The above was meant to tickle your interest
just enough to read further;

'Algorithms + Data Structures = Programs" by N Wirth.
A good book to read if you are interested in good
programming. Doesn't star PASCAL, but makes the eight
queens problem look trivial.

'Pascal Report & User Guide' by Jensen & Wirth. The
Pascaler's bible, and very good.

'A Primer in Pascal' Absolute rubbish

Anything by Wirth is wirth reading ...

Now What ?

   I shall persist in writing algorithms in PASCAL
(even if I cannot run them) just as I used to use
flowcharts.
   PASCAL has what most other languages have not -
Charisma. It will continue it's mushroom growth
until something better comes along.
   There has been a lot of talk of a common uP code,
mainly along the lines of assembler type codes. I
think PASCAL or some subset of it would be better,
or even MODULA for whole uP systems. Any assembler
could not do justice to all the families of uP,
yet hand compilation of PASCAL is so simple that
even if small machine compilation is impractical
then at least algorithms would be portable ; they
would also be transparent, and hence easily used
by others.

TERMINAL
SELECTRIC I/O I/O

   I believe that a number of ACC members may have
purchased IBM Selectric I/O typrwriters for use in
personal computer systems. Unfortunately these
machines are not simple to interface (especially
where ASCII coded I/O is required) requiring a fair
number of TTL chips for timing, signal gating etc.
   If any member is experiencing difficulty in des-
igning a suitable interface, if he (or she) contac-
ts me I would be pleased to supply information,
having successfully interfaced two machines to an
Intel 8080 system.
   At £100 each, an extra £15 to £20 is well worth
spending for an interface to allow the use of a
truly superb I/O device.
John F Anderson 72 Sneinton Dale, Nottingham

HANOI

```
'BEGIN'
    'INTEGER' N;
    N:=READ;
    'BEGIN'
        'INTEGER' ELEMENT, TEMP, WORK;
        'INTEGER' 'ARRAY' ROD[1:3,1:N], TOP[1:3];

        'PROCEDURE' STACKPRINT;
        'BEGIN'
            'FOR' TEMP:=1 'STEP' 1 'UNTIL' 3 'DO'
            'BEGIN'
                WRITETEXT('(''('C20S')'ROD%')')';
                'IF' TEMP=1
                'THEN' WRITETEXT('('A%')')
                'ELSE'
                    'IF' TEMP=2
                    'THEN' WRITETEXT('('B%')')
                    'ELSE' WRITETEXT('('C%')');
                'IF' TOP[TEMP]=0
                'THEN' WRITETEXT('('IS%EMPTY')')
                'ELSE'
                    'BEGIN'
                        WRITETEXT('('CONTAINS%THE%DISCS%')');
                        'FOR' WORK:=1 'STEP' 1 'UNTIL' TOP[TEMP] 'DO'
                            PRINT(ROD[TEMP,WORK],3,0)
                    'END'
            'END' OF TEMP;
            NEWLINE(1)
        'END' OF STACKPRINT;

        'PROCEDURE' HANOI(N,A,B,C);
        'VALUE' N,A,B,C;
        'INTEGER' N,A,B,C;
        'IF' N=1
        'THEN' MOVE(A,B)
        'ELSE'
            'BEGIN'
                HANOI(N-1,A,C,B);
                MOVE(A,B);
                HANOI(N-1,C,B,A)
            'END' OF HANOI;


        'PROCEDURE' MOVE(X,Y);
        'VALUE' X,Y;
        'INTEGER' X,Y;
        'BEGIN'
            WORK:=TOP[X];
            TEMP:=ROD[X,WORK];
            TOP[X]:=WORK-1;
            WORK:=TOP[Y];
            TOP[Y]:=WORK+1;
            ROD[Y,WORK+1]:=TEMP;
            STACKPRINT
        'END' OF MOVE;

        'FOR' ELEMENT:=1 'STEP' 1 'UNTIL' N 'DO'
            ROD[1,N-ELEMENT+1]:=ELEMENT;
        TOP[1]:=N;
        TOP[2]:=TOP[3]:=0;
        PAPERTHROW;
        STACKPRINT;
        HANOI(N,1,2,3);
        WRITETEXT('(''('2C20S')'***%ALL%DISCS%ARE%NOW%ON%ROD%B%***')')';
        PAPERTHROW
    'END' INNER BLOCK
'END'
```

This Algol 60 program was written as an exercise in recursive programming. It calcuates and prints out the solution to the Towers of Hanoi puzzle (V4 I6) for a given No of discs, which is read in as data.

Philip Rutherford

ROD			8	7	6	5	4	3	2	1
ROD A	CONTAINS THE DISCS		8	7	6	5	4	3	2	1
ROD B	IS EMPTY									
ROD C	IS EMPTY									
ROD A	CONTAINS THE DISCS		8	7	6	5	4	3	2	
ROD B	IS EMPTY									
ROD C	CONTAINS THE DISCS		1							
ROD A	CONTAINS THE DISCS		8	7	6	5	4	3		
ROD B	CONTAINS THE DISCS		2							
ROD C	CONTAINS THE DISCS		1							
ROD A	CONTAINS THE DISCS		8	7	6	5	4	3		
ROD B	CONTAINS THE DISCS		2	1						
ROD C	IS EMPTY									
ROD A	CONTAINS THE DISCS		8	7	6	5	4			
ROD B	CONTAINS THE DISCS		2	1						
ROD C	CONTAINS THE DISCS		3							
ROD A	CONTAINS THE DISCS		8	7	6	5	4	1		
ROD B	CONTAINS THE DISCS		2							
ROD C	CONTAINS THE DISCS		3							
ROD A	CONTAINS THE DISCS		8	7	6	5	4	1		
ROD B	IS EMPTY									
ROD C	CONTAINS THE DISCS		3	2						
ROD A	CONTAINS THE DISCS		8	7	6	5	4			
ROD B	IS EMPTY									
ROD C	CONTAINS THE DISCS		3	2	1					

6800 LIBRARY

First, the opposition !
I hold a limited amount of information on the following microprocessors; SC/MP,MICPROC,F-8,6100,9090,2650,5710, 1770,N601,9080,1600 & general inf.
Also the 6500, which is distributed by Aardalect Electronics Ltd., Suite E,Georgian House,Trinity St., Dorchester,Dorset 0305 67141
Aardelect have donated a complete set of manuals, price list, data etc. The CPU chip is of the order of £13.
For an Index of the above items please send an SAE etc. I'm not trying to monopolise the library business so if anybody wants to take over any of these items please feel free...

Now for 6800 lovers
Once again will all requests please enclose an A4 envelope and 32p stamps, any surplus will be returned.

Ed's note; We are unable to print the additions to the 6800 library in this ACCN because of lack of space. Those not already in touch should get in touch with Tim Moore, enclosing a SAE.
Tim Moore, 24 College Rd.,Maidenhead, Berks   (0628) 29073

8080/Z80 LIBRARY

***Ed's note; again, I've been forced by lack of space to leave the latest library list out of this ACCN. Your contact is, of course, Neil Harrison,15 Hill Rd.,Watchfield, Swindon,Wilts, SN6 8SB SAE PLEASE

EDS BIT

Note to new members; This issue is a bit larger than normal!
My apologies to all those who sent in excellent articles which I've not been able to include in this issue, but then it means that I can confidentally promise readers an interesting Iss 2 (& 3 !)
Looking forward to seeing many ACC members at the DIY COMPUTER expo on May 14.   Mike Lord

---

WB USER GROUP

This consists of a register of all interested parties and a library of software and hardware articles. The register and indexes will be circulated every 3 or 4 months to the participants. If you are about to, in the middle of, or have completed WB-1 please contact me.
Tim Moore
24 College Rd.,Maidenhead,Berks

FOR LOAN

Teletype KSR35 Manual (Vol 2 - adjustments) with type 28 typing unit & keyboard description & principles of operation. Send 80p to cover P&P.
R Kellock, New Mills, Tisbury, Wilts

TRADE

I would really like to correspond or trade with some of your members. I've being trying to collect runs of various English electronics magazines as well as receive current issues. Subscriptions are not possible at all for Practical Wireless, Practical Electronics, Television & Elementary Electronics. Electronics Today International cut me off since they now have a Canadian edition (very poor imitation is is). Wireless World arrives occasionally, but not often enough for me to renew my sub. Elektor comes in with great regularity, but in pieces. They use a very cheap mailing label and the PO was never known for gentleness.

So what I'd like to find is somebody to buy me copies on the news stand every month, put them to-hether in an envelope or parcel and post them to me. I can pay in sterling (I maintain an account in London) or trade for American magazines. I have access to virtually everything electronic published here and also have stacks of back issues. I know there are boxes in the garage containing back issues of Electronics, QST, 73, Computer Design, Electronic Design, Datamation etc.

Since I have good collections of all these, as well as most IEEE publications, I'm willing to photo-copy particular articles if somebody needs only one article.

I'm also willing to help anybody who wants to trade parts or who is having trouble getting US parts. I have sent parts to England in the past in return for books or magazines published there. Don't have much at home right now, a few 8080's and some 1702's are about all that's worthwhile. But I'm building the ETI synthesiser and want to start on their organ soon. I'm sure we could work out some sort of trade. If nothing else you can't buy Meccano in the US and their parts are always in demand around my house.

Billy H Pettit
PO Box 580   Bethany, Oklahoma, 73008   USA

INTERFACE STANDARDS

V24, RS232

THE ELECTRICAL SPECIFICATIONS OF DATA TERMINAL EQUIPMENT INTERFACES

Introduction

Many of us, in trying to get microprocessor kits to work, or to interface Teletypes, modems, or similar, have probably come across expressions like 'V24', 'RS232' or '20 mA current loop'. These are all standard methods of providing an interface for transferring data to allow interchangability of consoles, modems etc. They are electrical specifications of circuits for transferring a two condition code, or in other words a binary '0' or '1', from one piece of equipment to another. This code may be used for serial data transmission or simply turning functions on and off.

I have attempted to give the substance of the definitions for those interested, and followed these by practical circuits which are tested and proven, so they can be used without fear.

The specifications covered are;
    CCITT V24
    EIA RS-232-C   (RS-232, issue C)
    '20 mA Current Loop'

To all intents and purposes the first two are identical, which leaves us with just 2 types of interchange circuit in common use. These will be dealt with in turn below. As far as most of us are concerned, Data Terminal Equipment is just a smart way of saying 'console'.

V24 (RS-232-C)
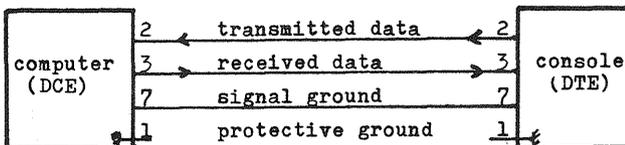
pin 1 is connected to the metalwork of the pieces of equipment to allow connection of external earths if required.

Although widely referred to as 'V24', the electrical characteristics of the V24 interface are actually defined in CCITT Recommendation V28. V24 defines a 'shopping list' of different interchange wires which may be required at a Data Terminal Equipment / Data Communications Equipment interface.

At the V24 interface there will be one common return wire (signal ground) and a set of other wires carrying the signals. For the simplest computer console there are 2 signal wires, one for serial data in each direction. For more advanced applications further signals will be required to provide 'ready' lines etc. The definition of these is beyond the scope of this article, and anyway depends upon the application. Those interested should refer to V24.

Each signal wire is unidirectional and signals carried by that wire are defined in terms of the voltage at the interface.
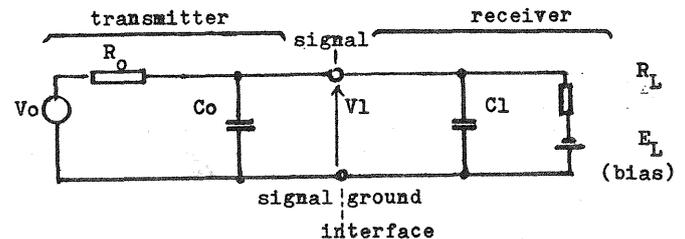
INTERCHANGE EQUIVALENT CIRCUIT

In the above diagram, if $V1$ is more negative than $-3V$ there is said to be a 'marking' condition, and if $V1$ is more positive than $+3V$ there is a 'spacing' condition. Voltages in between these are undefined.

The following table gives the logical equivalents of the two states. By referring to it you should be able to get your data the right way up the _first_ time;

NOTATION	INTERCHANGE VOLTAGE V1		COMMENTS
	NEGATIVE	POSITIVE	
Binary state	1	0	signal polarity for UART's
Signal condition	Marking	Spacing	
Signal condition	Z	A	teleprinters
Function	OFF	ON	V24 functions like 'Ready to Send'
Amplitude modulatin	Tone On	Tone Off	
Frequency modulatin	Low Freq	High Freq	Modems
Perforations	Perforation	-	Paper tape
Start - Stop Code *	Stop element	Start element	

* In serial data transmission using Start-Stop code the quiescent line condition (i.e. while not sending characters) is Mark.

The Rest Of The Specification (V28)

In order to ensure that the transfer of data across the interface is trouble free, and that $V1$ reaches $\pm 3V$ under a wide range of conditions, a number of rules are laid down concerning the trans-

mitter and receiver circuits. These are summarised below, for the full spec see V28, or RS232-C.
(The following is fairly dry and can be skipped if you are not going to design your own circuits but use the ones given later)

The power-off source impedance of the transmitter must not be less than 300 ohms.

The transmitter shall withstand open circuit or short circuit to any other wire in the interface cable without damaging anything or allowing more than $\frac{1}{2}$A to flow.

Transmitter design shall be such that, with R_L im the range 3000 - 7000 ohms, and $E_L = 0$, V_1 shall be in the range 5 to 15V.

The receiver shall withstand $\pm$ 25V, and the open circuit transmitter voltage shall not exceed $\pm$ 25V.

R_L must be between 3000 and 7000 ohms.

C_L must not exceed 2500pF.

The reactive component of the load impedance must not be inductive.

The open circuit terminator voltage must not exceed + 2V.

Transitions in signals at the interface must pass through the transition region (-3 to +3V) without changing direction, in less than 1mSec or less than 4% of a signal element, whichever is shorter. However it must not change faster than 30V/uSec.

Practical Circuits for RS232-C/V28

In practice the standard is very easy to implement. The easiest and most compact (and expensive) method is to buy an integrated circuit specially designed to convert between TTL and RS232 levels. Examples are Texas 75150 (dual driver) and 75154 (quad receiver), and Motorola MC1488 (quad driver) and MC1489 (quad receiver).

A typical circuit using the Texas IC is shown below;
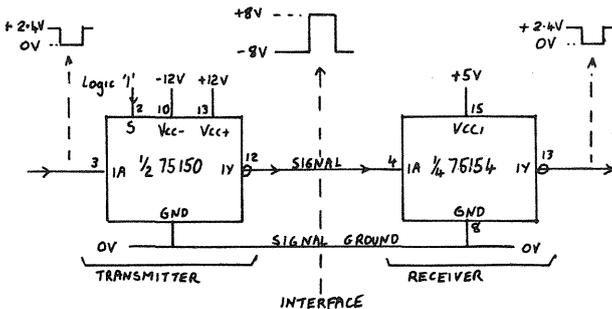
Refer to data sheets for more information. Unfortunately these IC cost about £3 each.

Simple discrete circuits are possible, as shown
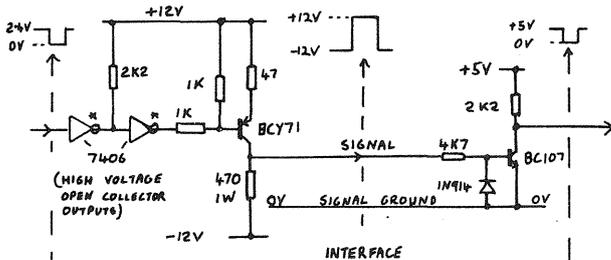
| Data is true (non-inverted) at this stage. If required, the first inverter and the 2k2 pull-up resistor may be omitted to allow an inverted signal to be applied.	Data is true (non-inverted) binary at this stage. It should pass through at least one gate to speed it up if it is required to provide a clocking edge. Otherwise it may pass directly to (e.g.) the UART without inversion.

'20 mA Current Loop'

Unfortunately this is not a standard, as such. It is in fact the interface which Teletype have used for a long time and owes a lot to relay circuitry. However, companies selling 'Teletype replacements' have been obliged to simulate the interface so it has become a sort of de-facto standard, even when solid state circuitry is employed.

In essence the current loop works as follows; The receiver contains a current detector and is connected by two wires to the transmitter which contains a switch which can be either open or closed. Thus current either flows round the loop (a marking condition) or doesn't flow (spacing) (Those BEA 35RO's recently on the market have an active 20 mA receiver but current flowing represents a spacing condition. i.e. non-standard) A slight complication occurs in that the EMF required to drive the current round the loop may reside either in the transmitter or the receiver.

A transmitter or receiver containing the EMF is said to be 'active' and a transmitter or receiver without the EMF is said to be 'passive'. Each transmitter to receiver connection must contain one active and one passive element.

A Teletype ASR33 for example is completely passive, this means that the receiver is simply a relay coil whilst the transmitter is a simple open/closed contact.

A theoretical representation of the interface to a passive console is shown below
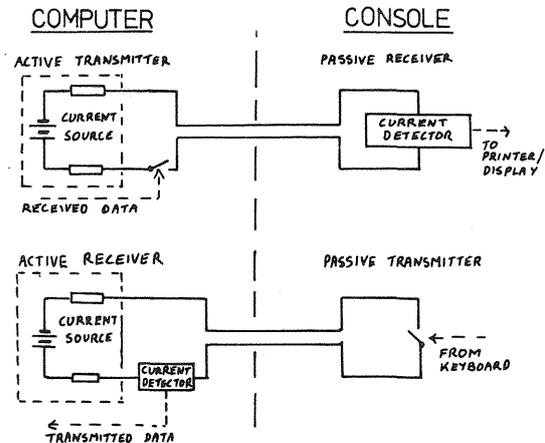
Typical 'Specification'
Receiver Passive, isolated

	MIN	MAX
Voltage drop, marking	1.2V	2.7V
Spacing current	0	3mA
Marking current	15mA	80mA

Transmitter Passive, isolated

	MIN	MAX
Open circuit voltage (of circuit being driven)	5V	40V
Spacing current	0	2mA
Marking current	20mA	80mA

Practical Circuits (20mA)

Suggested circuitry is given below showing both ends of the links in the case of a passive console. However if you have to build both ends of the link yourself it would be worth considering using RS232 instead. 20 mA loop should really only be used where this is necessary to interface existing devices such as a Teletype you happen to have. The reasons are that RS232 is;
- Cheaper (using discrete components)
- Faster (20 mA may not be reliable above 110 baud)
- A real international standard

COMPUTER | CONSOLE

ACTIVE TRANSMITTER | PASSIVE REC., ISOLATED

ACTIVE RECEIVER | PASSIVE TRANS, ISOLATED

TRUE DATA | INTERFACE | TRUE DATA
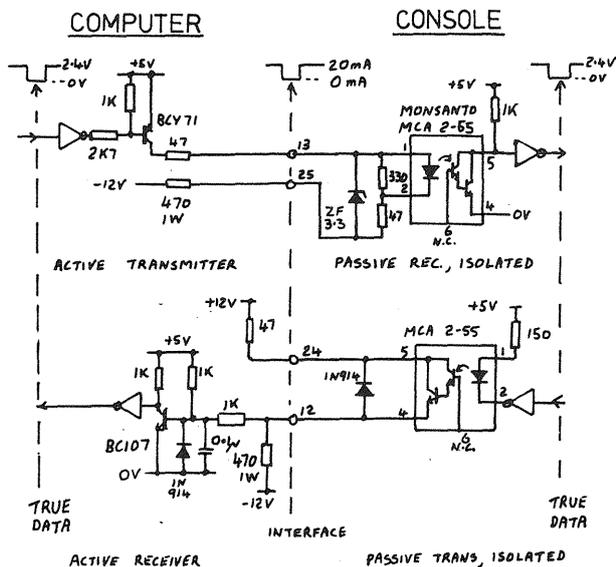
Notice that each link has an isolation element (the opto-coupler). This should only be necessary at one end of the link (preferably the passive end otherwise a separate floating supply would be required). There is no reason why both ends should not be isolated (apart from cost). For example the current loop interface on the 6800 evaluation kit MEK6800D1 is an active, isolated, interface. It depends upon having a floating ± 12V supply, to maintain the isolation. It isn't necessary to worry too much about this as the console end of the link will almost certainly be isolated, especially if you make it and use the circuit shown.

Connections

A particular type of connector has become standard for V24/RS232 type interfaces, although strictly speaking it is not part of the specification. This is available from many manufacturers, for example the Cannon D type, 25 pin. RS Components stock it as 466-191 (male) 466-220 (female),466-258 (cover).

Female connector
(Wiring side)
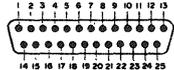
I would suggest that the standard connector is used in hobbyist systems together with the standard pin connections. Although this may cost a little more than minumum solutions, it will greatly aid such activities as joint experiments with other amateurs, also quick connection to borrowed professional equipment such as modems or consoles.

The computer has s female connector, the console is terminated with a male connector. The standard pin numbers for V24 interfaces are as follows;

1 Chassis Ground (available for safety purposes)
2 Transmitted data (console to computer)
3 Received data (computer to console)
7 Common return (signal ground)

In the interests of economy and standardisation it is recommended that any alternative 20mA loop interface provided by the computer be available on the same connector. The convention which I follow is that which Intel use on their MDS teletype connector, and the connections are listed below;

12 Console transmitter -
13 Console receiver +
24 Console transmitter +
25 Console receiver -

(The convention I have used for polarity is that a wire marked + is either the positive side of an active element, or the wire in a passive element which will be connected to it)

I would like to suggest that the above pin connections be adopted as a Club standard, and used by members

References

1) CCITT Recommendation V1. Equivalence between binary notation symbols and the significant conditions of a two condition code. Green Book Vol VIII

2) CCITT Recommendation V24. List of definitions for interchange circuits between data terminal equipment and data circuit termination equipment. Green Book Vol VIII

3) CCITT Recommendation V28. Electrical characteristics for unbalanced double-current interchange circuits. Green Book Vol VIII

4) EIA Standard RS-232-C. Interface between data terminal equipment and data communications equipment employing serial binary data exchange. Electronic Industries Association, Aug 1969

Suppliers

Component	Supplier	V Approx Price
SN75150/154	Anzac	£3 each
MC1488/9	Cramer	£3 each
MCA 2-55 (equivalents Litronix ILCA 2-55, GE H11BX503)	Jermyn	£2
Type D connectors	RS Comp (Try Doram)	£3

Any suggestions as to cheap reliable suppliers of the above parts who are prepared to sell small quantities to the amateur ?

© Pat Crowe 1977

Copyright waived for strictly amateur use.

A folder containing copies of V1,V24,V28 & RSC232 specifications is available on loan to ACC members. Send 15p stamps.   Mike Lord

---

COMMUNICATION STANDARDS   R A MUNT

There are any number of line standards for communications equipment and probably an even greater number of line protocols.

The two most common standards are 'current loop' and EIA RS232-C / V24.

CURRENT LOOP

This has become a standard by use and not by design. There is no formal specification for it, or a standard connector.

There are two variations of current loop, both of which were originally used with slow speed electromechanical equipment.

The original current loop had a value of 60mA for use with teletypes etc. running at speeds up to about 110 baud. This has given way to the 20mA current loop used with all sorts of line equipment at speeds up to 9600 baud. Practically all current loop equipment made during the last 10 - 15 years is of the 20mA type.

Current loop works as follows;

During the idle condition (no data) current is flowing and is equivalent to the 'mark' or binary '1' level. As a character is serially transmitted the current is switched on and off according to the '1's and '0's respectively of the code. Current loop will almost always be an eleven unit transmission for an 8 bit code.

As it is a current loop, it is the current that is specified (20 or 60 mA) and not the voltage causing the current. This can sometimes cause a problem with current loop equipment in two different ways;

Obviously at the transmitting end something has to switch the current on and off. In electromechanical units this is usually an actual switch, which for all practical purposes has zero volt drop when 'on' and has no trouble withstanding, say, 100 V across it when open. However mechanical switches cannot operate at high data rates such as 9600 baud where the bit time is approx 100uS. For electronic equipment

the switching device is usually a transistor, and as high voltage transistors are more expensive than low voltage transistors, the current loop source voltage is usually limited to 30V.

The transistor is also polarity conscious and has a definite though not significant 'on' resistance.

Now comes the second problem, the receive end. As this is a 'current loop' the input cannot usually be related to ground in any way. In electromechanical equipment this is OK as the i/p device is often an electromagnet which only has two wires anyway. The problem here comes with the volt drop, which may be 20 volts or more.

For an electronic system to achieve a 'floating' input either inductive coupling or an optical isolator is used. These have a much lower voltage drop of about 2 V for the opto-isolator to about 5 - 10 volts for inductive couplings.

What this amounts to practically is that it may be impossible to connect an electronic transmitting device to an electromechanical receiver because by the time the source voltage has reached a sufficiently high value to get 20mA flowing the driving transistor will be over-volted.

There should not be a problem the other way round where an electromechanical device drives an electronic receiver.

I have encountered source voltages for current loops ranging from 5V to 100V.

The one remaining snag with current loop relates to the lack of a formal specification, and in particular a tolerance on the 20 mA nominal. Many equipments will work with a - 50% error in the current value, but some will not. A reasonable figure is probably 10%.

The two most useful features of current loop are;

- Its relative insensitivity to interference.

- Due to the 'floating' nature of the i/p, ground loop problems are avoided.

To my knowledge no IC manufacturer makes any chips specifically intended to interface 20 (or 60) mA current loop to any logic family. However Hewlett Packard produce some opto-isolator chips that are very easy (chip + 2 resistors) to use with DTL, TTL or 5V CMOS.

EIA RS232-C / CCITT V24

These are formal standards for a voltage interface. The two standards are almost identical, EIA RS232-C is the American version and CCITT V24 the European.

While no particular connector is specified, the Cannon (etc.) 25D type is commonly used. Signals are allocated to specific pins of a 25 way connector as follows ;

Pin	RS232 name	V24 name	Description
1	AA	101	Protective Ground
2	BA	103	Transmitted Data
3	BB	104	Received Data
4	CA	105	Request To Send
5	CB	106	Clear To Send
6	CC	107	Data Set Ready
7	AB	102	Signal Ground
8	CF	109	Received Line Signal Detector
9			Reserved
10			"
11			Spare
12	SCF	122	Secondary Received Line Signal Det.
13	SCB	121	" Clear To Send
14	SBA	118	" Transmitted Data
15	DB	114	TX Timing (From Modem)
16	SBB	119	Secondary Received Data
17	DD	115	RX Timing (From Modem)
18			Spare
19	SCA	120	Secondary Request To Send
20	CD	108.2	Data Terminal Ready
21	CG	110	Signal Quality Detector
22	CE	125	Ring Indicator
23	CH/Cl	111/112	Data Rate Selector
24	DA	113	TX Timing (From Terminal)
25			Spare.

The voltage levels are + 3V to ± 15V for RS232-C and ± 6V to ± 15V for V24.

Signals are all active high (+) except for data where the binary '1' is active low (-). All signals are not normally used on any one piece of equipment. Various combinations of the signals are intended to control the various modems (a modem is a piece of equipment designed to allow the transmission of data over telephone lines).

Where the equipment (e.g. VDU) is close to the computer (say 100 - 1000 feet or so depending upon baud rate) and wire links are used, a three / four wire system (data in / data out and single or dual grounds) would normally be used.

Several IC manufacturers produce chips intended for use with these standards, both receive and transmit;

Quad drivers / receivers DTL/TTL ;
  Motorola MC1488, 1489
  National LM1488, 1489
  Texas   75188, 75189

Triple drivers /receiver DTL/TTL; Fairchild

Dual drivers / receivers DTL/TTL ;
  Signetics 8T15 8T16 (also for MIL88 standard)
  Texas 75150 75154

In addition to these standards there are occaisionally seen equipments MIL88 (+ 6V) and the PO + 80V.

PROTOCOL

Protocol relates to particular character sequences usually at the beginning and end of a 'message'.
Protocols are only 'standard' to a particular computer manufacturer and operating system(s).

FORMATS / CODES

There are many codes used for data communication. Where a computer is involved the code is usually 8 bits.

Many computer manufacturers have, in the past, used their own codes. Now the American ASCII code is becoming fairly universal and ISO7 along with it. ISO7 is identical to ASCII except for the provision of approx 8 of the codes for 'National Useage'. The English version of ISO7 changes 3 characters from ASCII. These are ASCII '#' changes to '£', '←' to '—' and '↑' to '∧'.

There are essentially two data formats, asynchronous and synchronous. In this context asynchronous means a character at a time where the interval between the characters may be from 2/3 bit times to infinity.
Synchronous means a group of characters (group size not defined) without stop/start bits, at a time. The interval between groups may be from one special synchronising character to infinity.

Current practice is for synchronous to be used only for high speed use from one computer to another and similar applications. Most terminal equipment uses the asynchronous format.

This may be 10 or 11 bits in length, electromechanical equipment always uses 11 bits for speeds up to 110 baud. Over about 110 baud 10 or 11 bits may be used, but 11 bits is still most common.
This format is in the logical '1', 'marking' state during idle conditions. When a character is being transmitted the first bit (start bit) is always a logical '0', 'spacing'. The next is the low order bit of the character code, followed by the next highest bit and so on until eight bits have all been sent. After the eigth bit there is either one (10 bit) or two (11 bit) stop bits, these are logical '1' levels.

```
'1'  ┌─ ─┬─ ─┬─ ─ ┬─ ─┬─ ─ ─┬─ ─ ─┬─ ─ ─┬──────┐
    ─┤St│1 │2 │4 │10│20│40│100│P│ Stp    11 bit
'0'  └──┴──┴──┴──┴──┴──┴──┴───┘
```
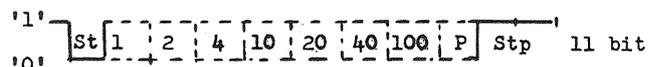
ASCII & ISO7 are both 7 bit codes i.e. they have 128 states. Although occaisionally transmitted as just 7 bits, they are usually sent with an eigth

bit. The eighth bit is usually used for 'even'
parity. Parity is a simple check that the character
has been received correctly, for even parity the
eighth bit is inserted as a '1' or '0' to make the
total number of '1's an even number - ignoring the
start & stop bits - for the character. Parity can
also be odd - to give an odd number of '1's.

Frequently the eighth bit, although present, is
ignored, merely being wired to one logic level or
the other in the serialisation logic. If wired to
logic '1' it is said to be a 'mark' or if logic '0'
a 'space'.

Baudot is a five level code (32 states) which is
only intended for national / international teleprinter links i.e. Telex. It is not really intended
or suitable for computer communications. However
there is quite a lot of Baudot equipment on the
surplus market, usually Creed.

FPA

Several thoughts have recently occurred to me on
the subject of floating point processing and other
topics.

1) Although there are many processors in existence,
all with different instruction sets, many software
ideas can be (or need to be) implemented on each
machine. A typical example is that of floating point
arithmetic (FPA). I do not see how it is possible to
formulate any 'universal machine language' due to
the differences between processors such as number of
accumulators and methods of conditional branches.
(If such a thing exists could someone let me have
details of it) As a result I feel that software
needs to be described by flowcharts, leaving implementation to the individual concerned.

2) Before coming to the reason to section 3, I'm
inserting this section as a comment.

FPA is done by hardware on some big machines, but
on minis and micros it is done by software. For those
who don't know, I'll summarise the traditional
methods of doing FPA (there are other ways).

a) Numbers are represented by 2 binary numbers, the
mantissa (M) with a sign bit, and the exponent (E)
in 2'c complement integer form.

So $x = \pm M * 2^E$ with $1 > M \geq \frac{1}{2}$ or M=0
Also, let the sign of M be a single bit stored in 'S'
so a number is defined by S,M,E

e.g. $11_{decimal} = 1011_{binary} = .1011 * 2^{100}$ (4 decimal)
so this would be stored as S,M,E = +,1011,0100

Addition, subtraction, multiplication and division
are detailed in flowcharts 1,2 & 3

Points to beware of;
- Exponent overflow (fatal)
- Exponent underflow (warning, set number to zero)
- Significance error (Big - Big = Small)

Note that 1 sign bit allows + & - numbers, an 8
bit exponent allows numbers less than $1.7*10^{38}$
down to $3*10^{-39}$ and zero, and a 23 bit mantissa
allows about 7 significant figures accuracy.

3) This brings me to my main point. Using section 2
it is now possible to do a lot of FPA, my problem
is how does one convert between S,M,E binary format
and BCD standard scientific format. For example, if
S = +, M=101000000..., E=00011110 then $x = +.625_{dec}$
$*2^{30}$ but for output to a teletype or display then
it is required to output this as $+6.71088 * 10^8$
Does anyone know a good method for doing this?

4) Reading matter. If you like playing with
numbers and analysing games to see if 'black' or
'white' has the winning strategy, then try getting
hold of 'On Numbers and Games' by J H Conway (of
LIFE fame) published by Academic Press at £6.50.
Do not buy it until you have read it, you might
find you get lost !

Steve Bailey

FLOW CHART 1

ADDING (OR SUBTRACTING) $x := a \pm b$
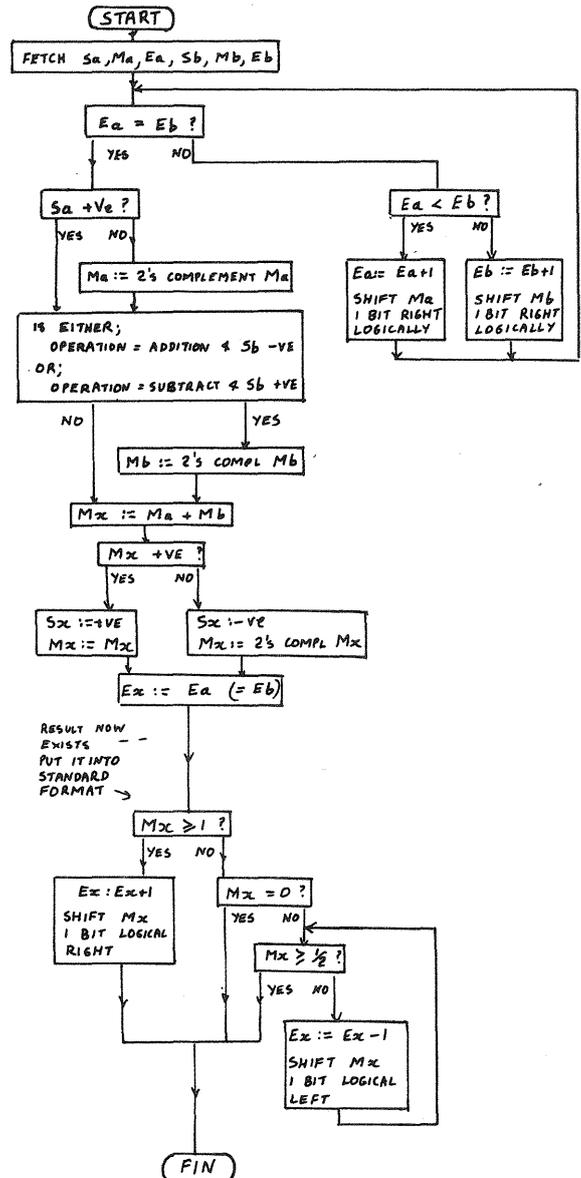
FLOW CHART 2

MULTIPLY $x := a * b$

FLOW CHART 3

DIVIDE $x := \dfrac{a}{b}$

```
              ( START )
                  |
    [ FETCH Sb,Mb,Eb,Sa,Ma,Ea ]
                  |
            [ Mb = φ ? ]
             YES   NO
              |     |
   [ FATAL        [ Ma = φ ? ]
     ERROR         YES   NO
     STOP! ]        |     |
                          [ Ex := Ea - Eb ]
                          [ Mx := Ma ÷ Mb ]
                                  |
                            [ Mx ≥ 1 ? ]
                             YES   NO
                              |     |
   [ Sx := +VE ]      [ Ex := Ex +1 ]
   [ Mx := φ ]        [ SHIFT Mx ]
   [ Ex := φ ]        [ 1 BIT RIGHT ]
                      [ LOGICALLY ]
                              |
                        [ Sa = Sb ? ]
                         YES   NO
                          |     |
              [ Sx := +VE ] [ Sx := -VE ]
                  |
         [ STORE Sx, Mx, Ex ]
                  |
              ( FINISH )
```
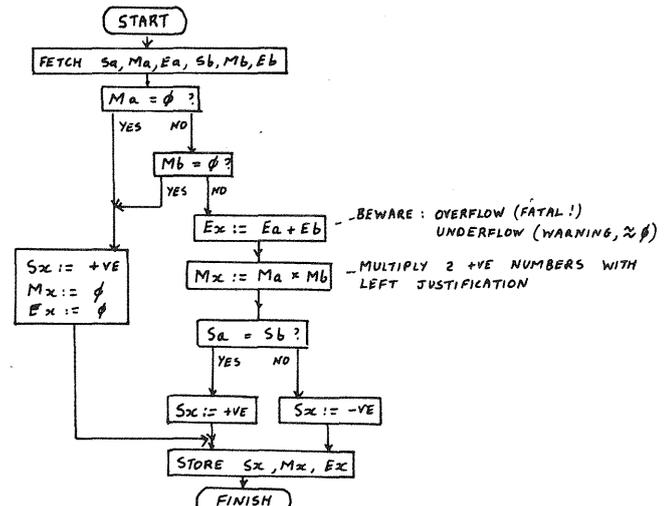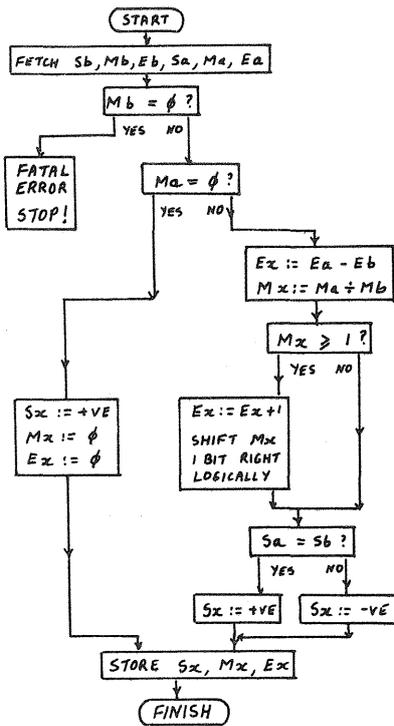

PROPOSED 'UNIVERSAL' 8-BIT MICROPROCESSOR BUS *

Anyone interested in a review of the above, which uses double width Eurocards, or interested in contributing to its design, please send A4 SAE to:
Pat Crowe, 43 Cypress Rd.,Woodley,Reading RG5 4BD

* to be featured in the next newsletter ... ed

ACC AGM

Held at the Polytechnic of the South Bank, London on 31st March 1977. 29 Members attended.
The Agenda was as printed in Vol 4 Iss 6.

a) Retiring Officers' Reports & Club Accounts

Bob Warren, Chairman, stated that from his point of view the past year had been generally successful. Membership had increased to over 600 and all Club activities had been well attended. Up to now the Club's main expenditure had been in the publishing and mailing of the Newsletter. Inflation had caused the members subscription fee to become inadequate to support this expenditure, and after a recent members' meeting it was agreed to increase the fee from £1 to £2. This will also allow an enlarged scope of activities to be supported and if necessary a larger newsletter to be produced.

Mike Reeve, Secretary, in his report also said that in his opinion the year had been a good one for club activities.

Mike Lord, Treasurer and Newsletter Editor gave the following statement of Club accounts;

At 10 March 1977 (just after distribution of V4 I6);

Membership; 633

Balance c/f from 1975/6		£ 22.80
Receipts for year 1976/7;		
V4 subscriptions	621.00	
Other - mainly back issues	174.60	
	£ 795.60	795.60

Expenditures for 1976/7;			
Printing		401.10	
Postage		305.70	
Other - mainly envelopes		28.88	
		£ 735.68	(735.68)
Balance @ 10.3.1977			£ 82.72

b) Election of Officers & Committee Members for 77/8

All of the previous year's Officers & Committee members were returned unapposed;

Chairman; R.H.Warren 90 Tudor Rd.,Hampton,Middx.

Secretary;M.Reeve 6 Limes Ave, N.Finchley,London
        N12 8QN

Treasurer & Newsletter Editor; M.Lord, 7 Dordells, Basildon, Essex

Committee Members;
        R.C.Selby 145 Bedfont Lane, Feltham, Middlesex TW14 9NH

        A.Keene The Galdor Centre, 52 Brighton Rd., Surbiton, Surrey KT6 5PL

        J.Creutzberg 135 Thornton Rd.,London SW12 0LJ

c) Discussion of Special Projects

The committee stated that the special project for the year 1976/7 (The TV VDU) was not totally successful because of problems with design co-ordination. Also the publishing of the details of a similar device in the ETI magazine largely took away the need for an ACC project.

After discussion it was decided not to adopt a specific project for 1977/8, but to publish as many members' original designs and modifications to existing designs in the ACCN as possible.

The design for a new computer, the 7768, was announced.

Some members advocated the progressing of the WB project, and to this end it was announced that a WB User Group was to be started by Tim Moore.

Also discussed was the problem of common bus standards for computer systems, but no firm conclusions were reached.

d) Visits & Lecture Programme

The next year's program was discussed. So far only two events; a visit to Signetics and a kit manufacturers' evening, were being firmed up. The following possibilities were suggested by members;

- A computer games evening at City University

- Visit to the Galdor Computer Centre at Surbiton (ICL 1900 now up & running !)

- A one-day club meeting, with demonstrations, bring and buy, etc.

The committee will look into these to see whether they are possible.

It was announced that the Club will be exhibiting at a forthcoming DIY computing convention at the IEE. Depending upon the success of this venture it may spread the gospel at the RSGB Convention later this year.

e) Any Other Business

- Tony West gave a short description of a Veroboard based 6800 computer system. Details will be released in a couple of month's time.

- Tim Moore described progress of the 6800 User Group Library.

- Bob Selby asked if any member had FORTRAN 4 for the PDP8.

At the end of the meeting the Chairman thanked all those who had attended, especially those who had travelled considerable distances.