ARM7 Coprocessor

User Guide

Overview:

During the development of the original ARM microprocessor, the ARM1, a test system was created called "The ARM Evaluation System" which ran an enhanced version of BASIC with up to 4Mbyte of RAM.

This allowed the developers to try out the new core without needing to design an entire computer, though as it wasn't sold as a commercial system very few were ever built.

The coprocessor philosophy (aka Tube® interface) available on all machines except the Electron and Master Compact allows a parasite processor such as the ARM to run the main language application, while leaving the host processor inside the BBC micro to spend its time handling the screen updates and other repetitive IO chores.

The huge success of ARM means that there are now many chip vendors around the world with licenses to use the ARM core in their own products, which means high speed ARMs with useful integrated peripheral devices like serial ports are readily available.

That makes it possible now to build a more up to date ARM evaluation system for the BBC micro.

Specifications:

The board comes with the following features:

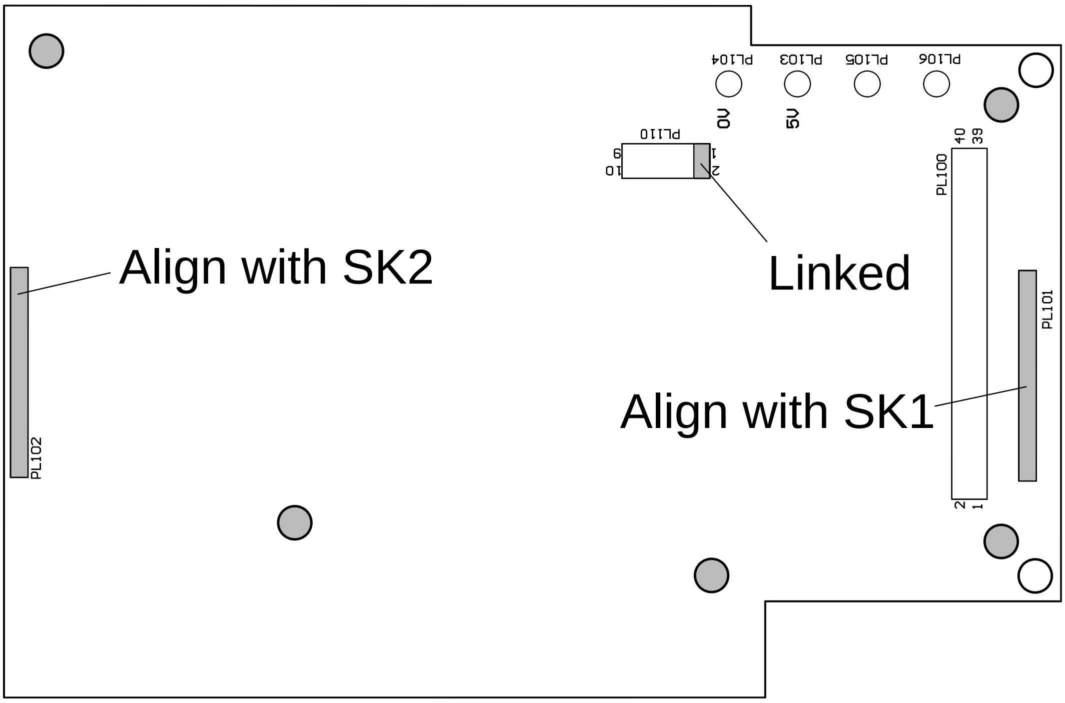
- ARM7TDMI processor at 64MHz
- 8kbyte unified cache to speed program execution
- For internal use within a Master series or external use with the entire BBC microcomputer family
- 16Mbyte high speed SDRAM
- 256kbyte on board flash ROM (containing Tube® system software and BASIC)
- Expandable to up to 64Mbyte RAM and 4Mbyte flash ROM (external)
- Optional serial EEPROM for parameter storage
- Optional RS232 port for debugging use

Fitting the internal coprocessor:

This section is only intended for installation of the board within a Master 128 or ET. With the machine turned off, remove the fixing screws and lid.

If there is already a coprocessor installed internally, this will first need to be removed by gently pulling it upwards and then storing it somewhere safe.

Before installing the coprocessor ensure that PL110 pins 1 and 2 are shorted with a jumper link.
Carefully line up and press in the two 12 pin mating headers with PL101 above SK1 and PL102 above SK2 on the motherboard.



Five holes (shaded above) can be used to lock the coprocessor in place using 18mm PCB standoffs (not supplied) if required, otherwise replace the lid and fixing screws.

Turn the machine back on, and enter the following commands at the command prompt

```
       *CONFIGURE INTUBE
       *CONFIGURE TUBE
```

this tells the computer that there is an internal coprocessor configured to be present. A CTRL-Break will be required for the new settings to take affect.

To disable the coprocessor without needing to physically remove it from the machine, enter the following command at the command prompt

```
       *CONFIGURE NOTUBE
```

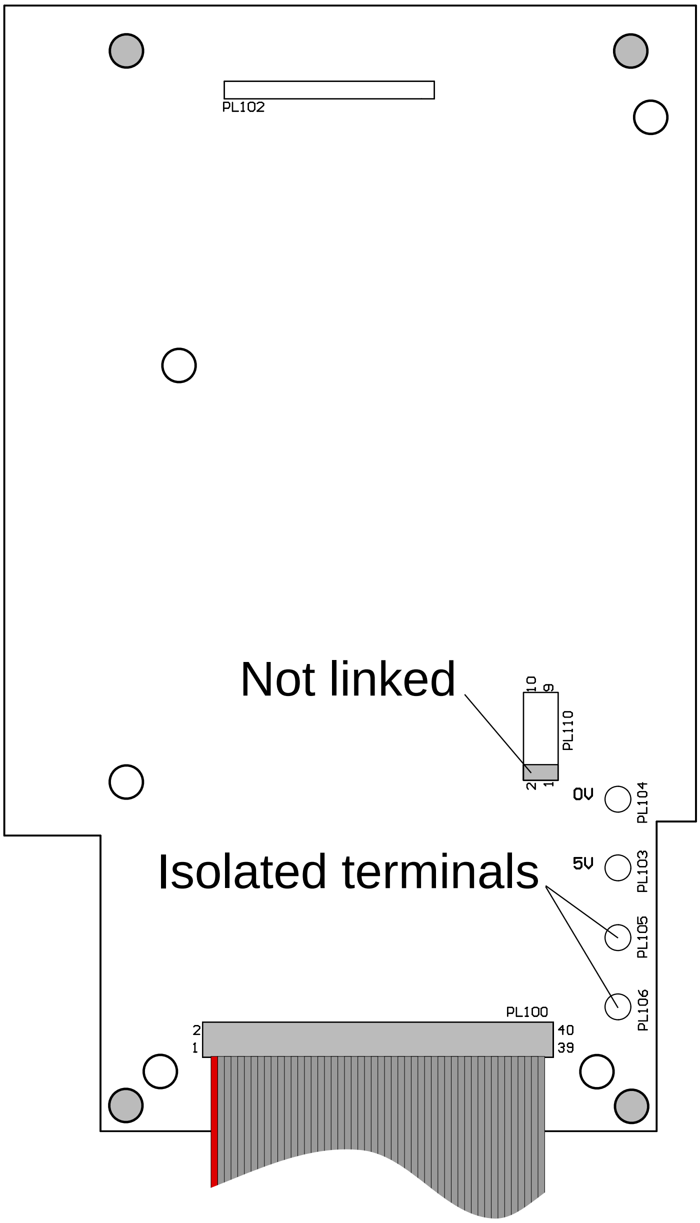
or see page C.5-2 of the Master Reference Manual for details.

Fitting the external coprocessor:

The coprocessor is designed to be fitted into a standard "cheese wedge" expansion box, or other suitable case with integral 5V power supply. These instructions assume a standard expansion box is being used.

Before installing the coprocessor ensure that PL110 pins 1 and 2 are not joined with a jumper link.

With the expansion box power supply turned off, remove the lid and insert the coprocessor feeding the ribbon cable from PL100 out through the bottom of the case.



Locate the pair of power terminals and connect the +5V supply lead (usually red) to PL103 and 0V or ground (usually black) to PL104. Two spare terminals, not normally fitted, are available to secure any unused power cables from the integral power supply - these are isolated.

Secure the coprocessor into the expansion box using the holes shown shaded above, and replace the lid. Do not yet turn on the coprocessor.

Connect the female end of the ribbon cable to the rightmost socket underneath the case, labelled Tube®, taking care to line up the red stripe of the ribbon cable with the pin 1 marker on the socket.

Firstly turn on the host computer (eg.BBC Model B). If the host is a Master series microcomputer then enter the following commands at the command prompt

```
       *CONFIGURE EXTUBE
       *CONFIGURE TUBE
```

this tells the computer that there is an external coprocessor configured to be present.

Now turn on the coprocessor power supply, and press CTRL-Break on the host computer.

Additional instructions for non Master users:

A small amount of coprocessor communications software is required in the host microcomputer in order for the Tube® system to run.

This software is included in the DNFS ROM and Acorn filing systems supplied with a 1770 disc controller, as well as some 3rd party filing systems.

Even if you do not intend on using the disc drive or network, one of these ROMs must be fitted in your machine. To find out if it is fitted type *HELP and look for a minimum of either

```
               DFS 1.20
       or      NFS 3.60
```

though the filing system need not be selected as the default filing system.

Starting using the coprocessor:

At start up the ARM attempts to run the default language ROM held in the host. If this is 6502 BBC BASIC then the coprocessor will observe this fact and run ARM BBC BASIC in its place, other non-ARM code will be faulted and the coprocessor will instead default to a command prompt.

With BBC BASIC selected, the first thing shown on the screen will be something like:

```
        ARM7TDMI Co-Processor 16MB

        Acorn DFS

        ARM BBC BASIC V

        Starting with 15,728,640 bytes free

        >
```

This is now waiting at the familiar BASIC prompt to accept either immediate mode keywords, or new lines of a BASIC program.

Try entering the following few lines at the prompt

```
  10 start% = TIME
  20 FOR loop% = 0 TO 359
  30 PRINT SIN(RAD(loop%))/COS(RAD(loop%)),TAN(RAD(loop%))
  40 NEXT
  50 PRINT (TIME - start%)/100;" seconds"
```

which should produce two columns of the same numbers due to the trigonometric identity that tangent = sine/cosine.

Of course, this isn't a very extensive use of the true capabilities of the coprocessor but it should demonstrate the extra processor speed available as the ARM calculates the angles and leaves the host BBC to do the printing in parallel.

For a detailed overview of the extra facilities of BBC BASIC V over BBC BASIC II or IV, see the publication "BBC BASIC Reference Manual" (product code AKJ20).

To start the coprocessor with a language ROM other than BBC BASIC will require a piece of ARM code being present in a sideways ROM slot in the host in a form which the coprocessor recognises. Details of the language ROM format are documented separately.

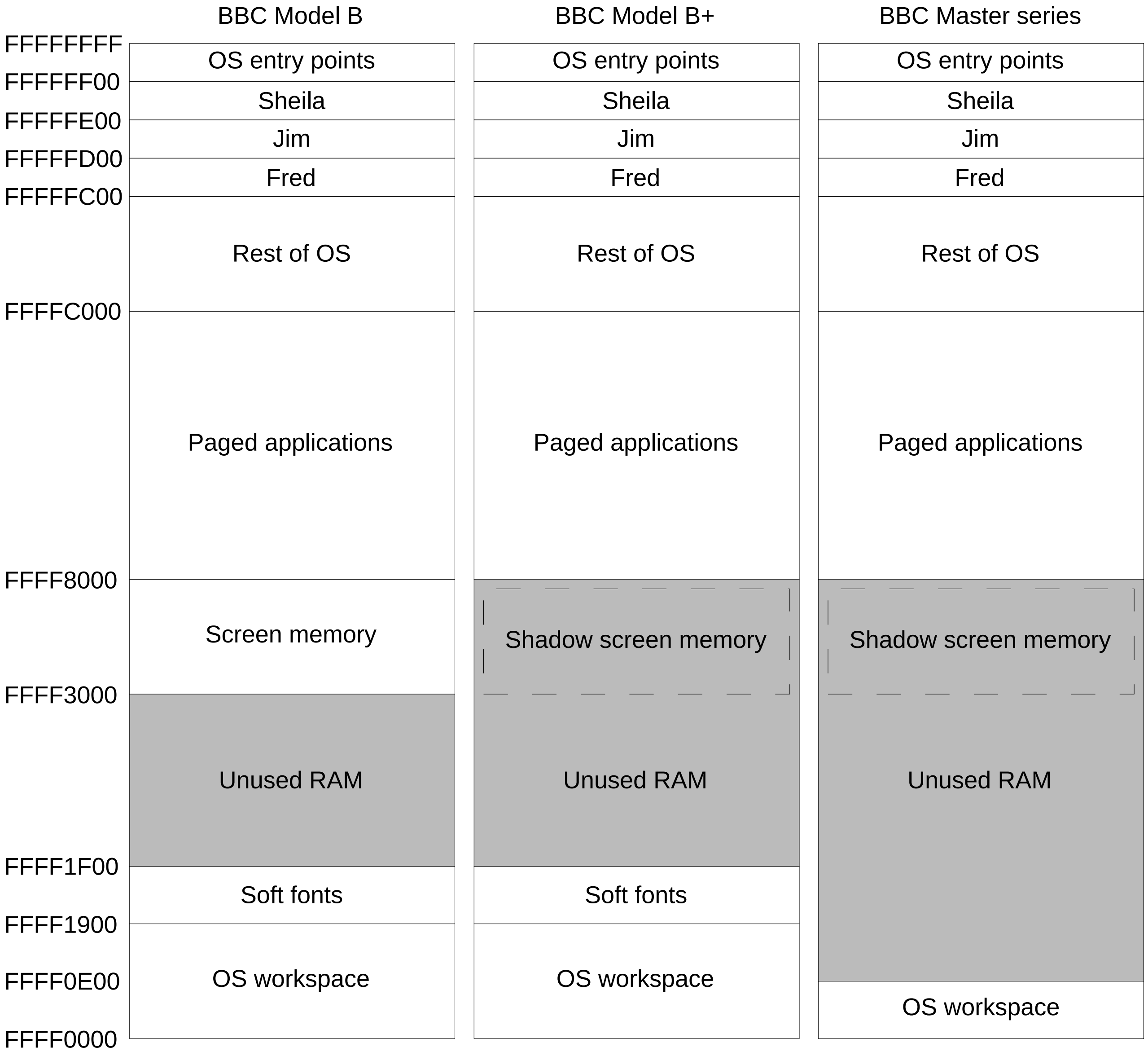
If you wish to quit BASIC in order to run your own machine code applications, type
  the keyword "QUIT"
or
  the star command "*QUIT"
from within BASIC to enter the supervisor prompt. For more details see "New commands".

Memory map:

Inside the host -

When the coprocessor is active, the memory map of the host is slightly modified compared with its normal layout as the application software is now running in the coprocessor the majority of the host RAM is unused.

Except for the Master series the value of OSHWM, which BASIC uses to set "PAGE", will be increased by &600 bytes as this memory is set aside for extra font definitions.



All screen modes are available and will not restrict the amount of application memory available, since the screen image is kept in the host not the coprocessor.

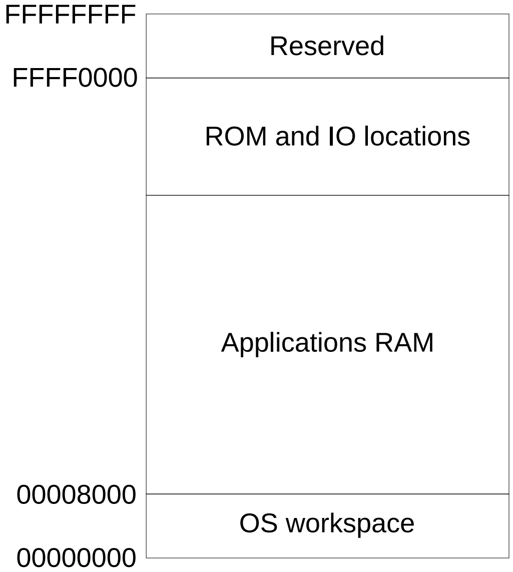
With the coprocessor turned off it is possible to use the shorthand form of an address to refer to memory within the BBC microcomputer by only specifying its bottom 16 bits, but when the coprocessor is in use a full 32 bit address must be used to allow the host and parasite to be referred to unambiguously.

Locations in the host have the top 16 bits as '1', therefore to read the currently selected ROM number which is held at &F4 in zero page on the host, its full address is &FFFF00F4.

However, remember that to access locations in the host the appropriate system call (OSWord 5 and 6) must be used as the shorthand ?&F4 in your programs running on the parasite will not access the value expected.

Inside the parasite -

The memory map of the parasite varies depending on how much memory is installed and whether any of the optional memory is soldered to the board. Applications authors should use the "OS_Memory" system call to find out where the memory is and what type it is.



For example, to find out how much ROM is installed you would use

```
  SYS "OS_Memory",776 TO ,pages,pagesize
  PRINT pages * pagesize;" bytes"
```

and details of the supported system calls are documented separately.

New commands (ALL USERS):

```
 Syntax: *GOS
         Starts the command supervisor and makes it the
         current language.
         Minimum abbreviation is *GO.
 Syntax: *GO [<address>] [; environment]
         Goes to the given address (default &8000) and runs
         the code found there, optionally passing the given
         environment string.
         Minimum abbreviation is *G.
 Syntax: *QUIT
         Ends the current application.
         Minimum abbreviation is *Q.
 Syntax: *HELP [subject]
         Reports the Tube® software version, then passes the
         command to the host machine to look for specific
         help on the subject requested.
         Minimum abbreviation is *H.
 Syntax: *BASIC [-help|-chain|-load|-quit] <filename>
         Starts the BBC BASIC interpreter, optionally with
         the given filename LOADed. Alternatively the
         filename can be CHAINed at start, or set to QUIT
         when an END statement is encountered.
         Minimum abbreviation is *B.
 Syntax  *CACHE [on|off]
         Controls the use of the processor cache. Enabling
         the cache makes programs run faster, but may cause
         some poorly written software to fail. Issuing the
         command with no parameters returns the current
         setting.
         Minimum abbreviation is *CAC.
 Syntax: *SHOWREGS
         Displays the register state at the last fatal
         internal error.
         Minimum abbreviation is *SHOWR.
```

Syntax: *ERROR [<number>] <text>

Forces an error to occur with the message set to the supplied text and an error number of zero (unless the optional error number is also supplied).

Minimum abbreviation is *ER.

Syntax: *INITSTORE [<data|reg>]

Wipes all of application memory by overwriting it with an invalid opcode value, or the value supplied after the command. The value can either be a number or register (see also the description *MEMORY).

Minimum abbreviation is *INI.

```
Syntax: *MEMORY [B|H] <addr1|reg>
                [[+|-] <addr2|reg2> [+ <addr3|reg3>]]
```

Displays a region of memory as words or bytes or half words plus their ASCII equivalents, the display is easiest to read in an 80 column mode such as MODE 3. When a program crashes this can be used to inspect the contents of memory to find what may have lead to the crash.

The region displayed is chosen by up to 3 additional parameters, each of which can either be a number or value taken from the exception registers (the ones that are displayed by the *SHOWREGS command).

Registers can either be referred to by number, such as 'R3', or by one of the special aliases 'PC' = R15; 'LR' = R14; 'SP' = R13.

The region displayed can be specified in four ways

```
  addr1
  addr1 addr2
  addr1 +offset1
  addr1 -offset1 +offset2
```

The first displays a default number of bytes starting at 'addr1'; the second displays 'addr2 - addr1' bytes starting at 'addr1'; the third displays 'offset1' bytes starting at address 'addr1'; the fourth displays 'offset1 + offset2' bytes starting at address 'addr1 - offset1'.

Minimum abbreviation is *MEM.

Enquiries:

Any enquiries should be made in writing to:

R.P.Sprowson,
6 Bollinbrook road,
MACCLESFIELD,
Cheshire.
SK10 3DJ.

Address correct December 2008.

Technical information:

Link PL110 -

This 2X5 way header is used to define the operation of the coprocessor.

Pins 1&2
Determines power source, these links should be made if the coprocessor is being used internally or open if external.
Pins 3&4
Unused at present.
Pins 5&6
Unused at present.
Pins 7&8
Unused at present.
Pins 9&10
Unused at present.

Link PL112 -

This 1X2 way header connects HOSTIRQ to the host computer, and should be left open.

Link PL107 and PL111 -

These two 1X3 way headers set the processor boot mode on startup, this is used in manufacture for test purposes. To set high (H) join pins 1&2, to set low (L) join pins 2&3.

Mode	PL111	PL107	Notes
Run from internal ROM	L	L	Default
Run from external ROM	L	H	If U202 fitted
Factory use	H	X	

Tube® registers and programmer's model -

The ARM processor communicates with the host via a set of byte wide registers which appear in its memory map. These buffer communications between the processors, in theory allowing for as close to parallel processing as possible.

The byte wide registers appear slightly differently to the host and to the parasite, as defined below:

Parasite view

	Read								Write							
	7							0	7							0
+0	A1	F1	P	V	M	J	I	Q	▨	▨	▨	▨	▨	▨	▨	▨
+2	Register 1								Register 1							
+4	A2	F2	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
+6	Register 2								Register 2							
+8	N	F3	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
+A	Register 3								Register 3							
+C	A4	F4	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
+E	Register 4								Register 4							

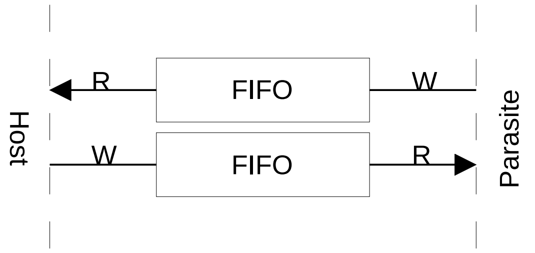
Host view

	Read								Write							
	7							0	7							0
+0	A1	F1	P	V	M	J	I	Q	S	T	P	V	M	J	I	Q
+1	Register 1								Register 1							
+2	A2	F2	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
+3	Register 2								Register 2							
+4	A3	F3	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
+5	Register 3								Register 3							
+6	A4	F4	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
+7	Register 4								Register 4							

Note that for the parasite register layout, the byte wide registers are spaced on 16 bit boundaries in hardware, so must be accessed using LDRB (or LDRH) to avoid accessing two registers at once.

The base address of the Tube® registers can be read with system call OS_Memory, details of which are documented separately. In the host they are fixed at &FEE0 within Sheila.

Each transfer register (1-4) is buffered in each direction by a FIFO, though the FIFO depth may only be 1 in some cases.



Alternate registers are used for control purposes, full details of the control protocol are documented separately, with the following meanings:

```
An = 1 = data available for collection in register n
   = 0 = no data available in register n
Fn = 1 = register n is not full
   = 0 = register n is full
P = 1 = activate parasite reset line
  = 0 = normal operation
V = 1 = register 3 has a two byte FIFO
  = 0 = register 3 has a one byte FIFO
M = 1 = enable parasite NMI line from register 3
  = 0 = disable parasite NMI line from register 3
J = 1 = enable parasite IRQ line from register 4
  = 0 = disable parasite IRQ line from register 4
I = 1 = enable parasite IRQ line from register 1
  = 0 = disable parasite IRQ line from register 1
Q = 1 = enable host IRQ line from register 4
  = 0 = disable host IRQ line from register 4
S = 1 = set flags indicated by set bits in the rest
        of the byte
  = 0 = clear flags indicated by set bits in the rest
        of the byte
T = 1 = clear all registers to the reset defaults
  = 0 = do not clear registers
N = 1 = action required on register 3
  = 0 = no action required on register 3
```

Power supply requirements -

The coprocessor is powered from an external supply when housed in an external coprocessor box or from the BBC Master power supply when housed internally.

Voltage: 5V ±10%

Current: 180mA max with the cache on
140mA max with the cache off