STEbus Interface

for Acorn RISC OS-based Computer Systems

User Guide and

Programmer's Reference Manual

Intelligent Interfaces Ltd

30th September 1996

be errors or omissions in this User Guide. Intelligent Interfaces welcomes comments and suggestions relating to the STEbus System for Acorn RISC OS-based Computer Systems and this User Guide.

All correspondence should be addressed to:-

Technical Enquiries
P O Box 80
Eastleigh
Hants
SO53 2YX

Tel: 023 8026 1514
Fax: 087 0052 128
Email: andy@intint.demon.co.uk

This User Guide is intended only to assist the reader in the use of the STEbus System for Acorn RISC OS-based Computer Systems and, therefore, Intelligent Interfaces shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this User Guide, or any in-correct use of the STEbus System for Acorn RISC OS-based Computer Systems.

The STEbus (IEEE Std 1000) Interface comprises two single Eurocards connected by up to 2m of ribbon cable. One card plugs into the expansion backplane of the computer and the other plugs into the STEbus backplane.

The interface enables the computer to directly address either the 4 Kbyte STEbus I/O space or the 1 Mbyte STEbus memory space (in 256 x 4 Kbyte pages).

The card which plugs into the backplane of the STEbus system provides STEbus system controller facilities. It is a potential master of the STEbus and has an on-card arbiter which enables the computer to be used in high performance multi-master systems.

Full software support is provided by a module.

1.1 Fitting the Expansion Card

The card can be fitted in any Acorn Computer with an expansion backplane.

To fit the card in an A300 series, A400 series, A540, A5000, Risc PC or A7000 computer:-

1 Switch off the power to the Computer.
2 Disconnect the Computer from the mains supply.
3 The card can be fitted in any unused expansion card slot.
4 Remove the blanking plate from the rear of the Computer and retain the two screws.
5 Fit the card and secure it in position using the two screws retained at stage 4 (if required, fit a joiner and blanking plate).
6 Reconnect the Computer to the mains supply.
7 Switch on the power to the Computer.
8 Confirm that the card has been fitted correctly by pressing F12 and typing

```
*Podules
```

This should list the STEbus as

```
Intelligent Interfaces STEbus Interface
```

together with any other cards fitted. Press <Return> to return to the Desktop.

The maximum number of STEbus Expansion Cards that can be fitted is limited only by the number of unused expansion card slots.

i) SYSCLK, a 16 MHz system clock signal. This is a link selectable option.

ii) SYSRST*, a bit in the control latch on the STEbus board enables the system reset signal to be under the control of software running on the computer. This is a link selectable option. Note that the STEbus system reset SYSRST* signal and the reset RST* signal of the computer are not directly connected allowing the two systems to be reset independently.

On-board Arbitration

The STEbus board provides an arbiter which, as supplied, implements an arbitration algorithm which initially allocates bus control to requests in the following order of priority - local RQ*, BUSRQ0*, BUSRQ1* and then works on a ' round robin' basis to ensure that no request is ignored forever. Other arbitration algorithms can be supplied to special order.

On-board/Off-board Arbitration

The on-board arbiter is a link selectable option and can be disabled. The local RQ is then linked to either BUSRQ0 or BUSRQ1 to request bus control from an arbiter on another STEbus board.

Master Type and Mode

The STEbus Interface is a potential master which, as supplied, works on a ' Release-On-Request' basis through the detection of another Bus Request BUSRQ* signal becoming active. Alternatively it can, if supplied to special order, work on a ' Release-When-Done' basis. The ' Release-On-Request' mode reduces the number of bus arbitrations required and thereby reduces the average computer STEbus access time. When the computer, via the STEbus Interface, is the only master in an STEbus system, the first arbitration is the only one ever carried out.

A bit in the control latch on the STEbus board enables a request by the STEbus Interface for control of the bus to be locked. This allows the ARM processor of the computer to perform the indivisible read modify write operations essential in a multi-master system.

Addressing

A bit in the control latch on the STEbus board enables the selection of either the 4 Kbyte of the STEbus I/O space or the STEbus memory space. A paging register is provided on the STEbus board to enable the 1 Mbyte of STEbus memory space to be addressed as 256 x 4 Kbyte pages. The 4 Kbyte of STEbus address space is mapped directly into the 4 Kbyte x 8 of MEMC podule address space allocated to each expansion card.

A state machine on the expansion card synchronises the MEMC cycle with the asynchronous handshake protocol of the STEbus. This, in turn, is implemented by a state machine on the STEbus board which involves the synchronisation lines address strobe ADRSTB*, data strobe DATSTB*, data transfer acknowledge DATACK* and transfer error TFRERR*.

Interrupts

The Cycle Error CYCERR* and System Reset SYSRST* interrupts can be individually linked to generate either IRQ or FIQ interrupts to the ARM processor of the computer. A control bit in the control latch on the STEbus board allows the two interrupts to be enabled or disabled as a group. A status register is provided to enable the source of the interrupt, CYCERR* or SYSRST*, to be determined.

The two interrupts are reset by a write to a reserved address.

Two control bits in the control latch on the expansion card allow all IRQ and FIQ interrupts from the STEbus interface to be independently enabled and disabled. A status register is provided to enable the source of the interrupt, IRQ or FIQ to be determined.

Bus driver and receivers

The bus drivers and receivers on the STEbus board meet the specification given in the IEEE Std 1000-1987.

2.2 Hardware Details

STEbus I/O or memory space

Computer addresses
Slot 0 - &03000000 to &03003FFC
Slot 1 - &03004000 to &03007FFC
Slot 2 - &03008000 to &0300BFFC
Slot 3 - &0300C000 to &03003FFC

(4 Kbyte x 8 of MEMC podule address space - each byte on a 32 bit word boundary)

Computer cycle - MEMC Read/Write

Computer expansion card status register

Computer addresses
Slot 0 - &03342800
Slot 1 - &03346800
Slot 2 - &0334A800
Slot 3 - &0334E800

Computer cycle - Fast Read

0	IRQ interrupt status (0 = no IRQ interrupt, 1 = IRQ interrupt)
1	0
2	FIQ interrupt status (0 = no FIQ interrupt, 1 = FIQ interrupt)
3	0
4	0
5	0
6	reflects the state of latch[6] IRQ interrupt enable
	(0 = IRQ disabled, 1 = IRQ enabled)

2 A12
3 A13
4 A14
5 A16
6 IRQ interrupt enable (0 = IRQ disabled, 1 = IRQ enabled)
7 FIQ interrupt enable (0 = FIQ disabled, 1 = FIQ enabled)

STEbus board status register

Computer addresses
Slot 0 - &03343000
Slot 1 - &03347000
Slot 2 - &0334B000
Slot 3 - &0334F000

Computer cycle - Fast Read
0 reflects the state of latch[0] ATN interrupt enable
(0 = disabled, 1 = enabled)
1 reflects the state of latch[1] CYCERR and SYSRST interrupt enable
(0 = disabled, 1 = enabled)
2 reflects the state of latch[2] RESET (0 = false, 1 = true)
3 reflects the state of latch[3] MEM I/O*
(0 = STEbus I/O address space selected,
1 = STEbus memory address space selected)
4 reflects the state of latch[4] LOCK
(0 = requests for control of the STEbus not locked,
1 = requests for control of the STEbus locked)
5 CYCERR interrupt status (0 = no CYCERR interrupt, 1 = CYCERR interrupt)
6 SYSRST interrupt status (0 = no SYSRST interrupt, 1 = SYSRST interrupt)
7 SYSRST status (0 = SYSRST* false, 1 = SYSRST* true)

STEbus board control latch

Computer addresses
Slot 0 - &03343000
Slot 1 - &03347000
Slot 2 - &0334B000
Slot 3 - &0334F000

Computer cycle - Fast Write
0 ATN interrupt enable (0 = disabled, 1 = enabled)
1 CYCERR and SYSRST interrupt enable (0 = disabled, 1 = enabled)
2 RESET if link S7 fitted (0 = SYSRST* false, 1 = SYSRST* true)
3 MEM I/O* (0 = select STEbus I/O address space,
1 = select STEbus memory address space)
4 LOCK (0 = requests for control of the STEbus not locked,
1 = requests for control of the STEbus locked)
5 unused state not relevant

2 unused state not relevant (write to reserved addres clears interrupts)
3 unused state not relevant (write to reserved addres clears interrupts)
4 unused state not relevant (write to reserved addres clears interrupts)
5 unused state not relevant (write to reserved addres clears interrupts)
6 unused state not relevant (write to reserved addres clears interrupts)
7 unused state not relevant (write to reserved addres clears interrupts)

STEbus board Attention Request status register

Computer addresses
Slot 0 - &03343008
Slot 1 - &03347008
Slot 2 - &0334B008
Slot 3 - &0334F008

Computer cycle - Fast Read
0 ATNRQ0* status (0 = no ATNRQ0, 1 = ATNRQ0)
1 ATNRQ0* status (0 = no ATNRQ1, 1 = ATNRQ1)
2 ATNRQ0* status (0 = no ATNRQ2, 1 = ATNRQ2)
3 ATNRQ0* status (0 = no ATNRQ3, 1 = ATNRQ3)
4 ATNRQ0* status (0 = no ATNRQ4, 1 = ATNRQ4)
5 ATNRQ0* status (0 = no ATNRQ5, 1 = ATNRQ5)
6 ATNRQ0* status (0 = no ATNRQ6, 1 = ATNRQ6)
7 ATNRQ0* status (0 = no ATNRQ7, 1 = ATNRQ7)

The Attention Request status register inverts the state of the ATNRQ* lines

STEbus board address latch

Computer addresses
Slot 0 - &0334300C
Slot 1 - &0334700C
Slot 2 - &0334B00C
Slot 3 - &0334F00C

Computer cycle - Fast Write
0 A12
1 A13
2 A14
3 A15
4 A16
5 A17
6 A18
7 A19

The address latch forms an STEbus memory space 4Kbyte page register

The STEbus Interface software provides SWI' s to enable any of the ten sources of IRQ interrupts (ATNRQ*<0...7>, CYCERR* and SYSRST*) to be claimed and released in a similar manner to the SWI' s OS_ClaimDeviceVector and OS_ReleaseDeviceVector provided by the RISC OS operating system. This simplifies the writing of interrupt routines.

3.1 The STEbus Module SWI's

STEbus_rdARCsts (SWI &420C0)

Purpose
To read the status register on the expansion card.

Parameters
R0 - expansion card slot number

Results
R2 - status
status[0] - IRQ interrupt status (0 = no IRQ interrupt, 1 = IRQ interrupt)
status[2] - FIQ interrupt status (0 = no FIQ interrupt, 1 = FIQ interrupt)
status[6] - reflects the state of latch[6] IRQ interrupt enable
(0 = IRQ disabled, 1 = IRQ enabled)
status[7] - reflects the state of latch[7] FIQ interrupt enable
(0 = FIQ disabled, 1 = FIQ enabled)

Example

```
10  computer_slot% = 1
20  SYS "STEbus_rdARCsts",computer_slot% TO ,,status%
```

returns the contents of the status register in the variable status%.

STEbus_wrARClat (SWI &420C1)

Purpose
To write to the control latch on the expansion card.

Parameters
R0 - expansion card slot number
R2 - latch
latch[5:0] - ROM page register
latch[6] - IRQ interrupt enable (0 = IRQ disabled, 1 = IRQ enabled)
latch[7] - FIQ interrupt enable (0 = FIQ disabled, 1 = FIQ enabled)

Example

```
10  computer_slot% = 3
20  SYS "STEbus_wrARClat",computer_slot%,,%01000000
```

status[1] - reflects the state of latch[1] CYCERR and SYSRST interrupt enable
(0 = disabled, 1 = enabled)
status[2] - reflects the state of latch[2] RESET (0 = false, 1 = true)
status[3] - reflects the state of latch[3] MEM/IO*
(0 = STEbus I/O address space selected,
1 = STEbus memory address space selected)
status[4] - reflects the state of latch[4] LOCK
(0 = requests for control of the STEbus not locked,
1 = requests for control of the STEbus locked)
status[5] - CYCERR interrupt status
(0 = no CYCERR interrupt, 1 = CYCERR interrupt)
status[6] - SYSRST interrupt status
(0 = no SYSRST interrupt, 1 = SYSRST interrupt)
status[7] - SYSRST status (0 = SYSRST false, 1 = SYSRST true)

Example

```
10  computer_slot% = 1
20  SYS "STEbus_rdSTEsts",computer_slot% TO ,,status%
```

returns the contents of the status register in the variable status%.

STEbus_wrSTElat (SWI &420C3)

Purpose

To write to the control latch on the STEbus board.

Parameters

R0 - expansion card slot number
R2 - latch
latch[0] - ATN interrupt enable (0 = disabled, 1 = enabled)
latch[1] - CYCERR and SYSRST interrupt enable
(0 = disabled, 1 = enabled)
latch[2] - RESET if link S7 is fitted
(0 = SYSRST false, 1 = SYSRST true)
latch[3] - MEM/IO* (0 = select STEbus I/O address space
1 = select STEbus memory address space
latch[4] - LOCK (0 = requests for control of the STEbus not locked,
1 = requests for control of the STEbus locked

Example

```
10  computer_slot% = 0
20  SYS "STEbus_wrSTElat",computer_slot%,,%00000001
```

writes %00000001 to the control latch to enable ATN interrupts.

STEbus_enableIRQ (SWI &420C4)

Purpose

To clear the IRQ interrupt enable bit[0] in the control latch on the expansion card.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 3
20  SYS "STEbus_disableIRQ",computer_slot%
```

disables IRQ interrupts from the expansion card in slot 3.

STEbus_enableFIQ (SWI &420C6)

Purpose
To set the FIQ interrupt enable bit[7] in the control latch on the expansion card.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 1
20  SYS "STEbus_enableFIQ",computer_slot%
```

enables FIQ interrupts from the expansion card in slot 1.

STEbus_disableFIQ (SWI &420C7)

Purpose
To clear the FIQ interrupt enable bit[7] in the control latch on the expansion card.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 1
20  SYS "STEbus_disableFIQ",computer_slot%
```

disables FIQ interrupts from the expansion card in slot 1.

STEbus_enableATNinterrupts (SWI &420C8)

Purpose
To set the ATN interrupt enable bit[0] in the control latch on the STEbus board.

Parameters
R0 - expansion card slot number

Example

Example

```
10  computer_slot% = 1
20  SYS "STEbus_disableATNinterrupts",computer_slot%
```

disables ATN interrupts from the STEbus interface in slot 1.

STEbus_enableSYSRSTandCYCERRinterrupts (SWI &420CA)

Purpose
To set the CYCERR and SYSRST interrupt enable bit[1] in the control latch on the STEbus board.

Parameters
R0 - expansion card slot number

Example

```
10  computer_slot% = 2
20  SYS "STEbus_enableSYSRSTandCYCERRinterrupts",computer_slot%
```

enables CYCERR and SYSRST interrupts from the STEbus interface in slot 2.

STEbus_disableSYSRSTandCYCERRinterrupts (SWI &420CB)

Purpose
To clear the CYCERR and SYSRST interrupt enable bit[1] in the control latch on the STEbus board.

Parameters
R0 - expansion card slot number

Example

```
10  computer_slot% = 2
20  SYS "STEbus_disableSYSRSTandCYCERRinterrupts",computer_slot%
```

disables CYCERR and SYSRST interrupts from the STEbus interface in slot 2.

STEbus_setSYSRST (SWI &420CC)

Purpose
To set the RESET bit[2] in the control latch on the STEbus board if link S7 is fitted SYSRST is set true.

Parameters
R0 - expansion card slot number

Example

```
10  computer_slot% = 1
20  SYS "STEbus_setSYSRST",computer_slot%
```

if link S7 is fitted on the STEbus interface in slot 1 SYSRST is set true

```
10  computer_slot% = 1
20  SYS "STEbus_clearSYSRST",computer_slot%
```

if link S7 is fitted on the STEbus interface in slot 1 SYSRST is set false.

STEbus_selectMEM (SWI &420CE)

Purpose
To set the MEM/IO* bit[3] in the control latch on the STEbus board to select STEbus memory address space.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 1
20  SYS "STEbus_selectMEM",computer_slot%
```

selects memory address space for the STEbus interface in slot 1.

STEbus_selectIO (SWI &420CF)

Purpose
To clear the MEM/IO* bit[3] in the control latch on the STEbus board to select STEbus I/O address space.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 1
20  SYS "STEbus_selectIO",computer_slot%
```

selects I/O address space for the STEbus interface in slot 1.

STEbus_lock (SWI &420D0)

Purpose
To set the LOCK bit[4] in the control latch on the STEbus board to lock requests for control of the STEbus. This enables the computer to perform indivisible read modify write operations essential in a multi-master system.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 1
```

```
20  SYS "STEbus_selectARP",computer_slot%
```

unlocks requests for control of the STEbus connected via the STEbus interface in slot 1.

STEbus_reset (SWI &420D2)

Purpose
To set the reset bit in the control latch on the STEbus board for 350mS. If link S7 is fitted (as supplied) this resets the STEbus.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 2
20  SYS "STEbus_reset"
```

STEbus_STEclri (SWI &420D3)

Purpose
To clear Cycle Error CYCERR and System Reset SYSRST interrupts.

Parameters
R0 - expansion card slot number

Example
```
10  computer_slot% = 1
20  SYS "STEbus_STEclri",computer_slot%
```

clears Cycle Error CYCERR and System Reset SYSRST interrupts from the STEbus interface in slot 1.

STEbus_rdATN (SWI &420D4)

Purpose
To read the Attention Request status register on the STEbus board.

Parameters
R0 - expansion card slot number

Results
R2 - status

status[0] ATNRQ0* status (0 = no ATNRQ0, 1 = ATNRQ0)
.
.
status[7] ATNRQ7* status (0 = no ATNRQ7, 1 = ATNRQ7)

STEbus_wrADR (SWI &420D5)

STEbus_rdIO (SWI &420D6)

Purpose
To read a byte from STEbus IO address space.

Parameters
R0 - expansion card slot number R1 - STE IO address

Results
R2 - data

Example

```
10  computer_slot% = 0
20  SPINCadr% = &3C0
30  SYS "STEbus_rdIO",computer_slot%,SPINCadr%+2 TO ,,data%
```

reads Port A of the Z8536 CIO1 (address at Base+2) on an Arcom SPINC board and returns the value in variable data%.

STEbus_wrIO (SWI &420D7)

Purpose
To write a byte to STEbus IO address space.

Parameters
R0 - expansion card slot number R1 - STE IO address R2 - data

Example

```
10  computer_slot% = 0
20  SPINCadr% = &3C0
30  data% = %11000100
40  SYS "STEbus_wrIO",computer_slot%,SPINCadr%+2,data%
```

writes the bit pattern 11000100 to Port A of the Z8536 CIO1 (address at Base+2) on an Arcom SPINC board.

STEbus_rdMEM (SWI &420D8)

Purpose
To read a byte from STEbus memory address space.

Parameters
R0 - expansion card slot number R1 - STE memory address

Results
R2 - data

R1 - STE memory address
R2 - data

Example

```
10  computer_slot% = 1
20  memadr% = &3A173
30  data% = &A5
40  SYS "STEbus_wrMEM",computer_slot%,memadr%,data%
```

writes the byte &A5 to the STEbus memory at address &3A173.

STEbus_rdMEMblk (SWI &420DA)

Purpose
To read a block of bytes from STEbus memory address space into the computer' s memory.

Parameters
R0 - expansion card slot number
R1 - start STE memory address
R2 - start computer memory address
R3 - number of bytes to read

Example

```
10  computer_slot% = 2
20  STEmemadr% = &00C00
30  DIM computer_memadr% 1024
40  count% = 1024
50  SYS
"STEbus_rdMEMblk",computer_slot%,STEmemadr%,computer_memadr%,count%
```

reads 1024 bytes from STEbus memory starting at address &00C00 into the computer' s memory reserved by the DIM statement (start address defined by the variable computer_memadr%).

STEbus_wrMEMblk (SWI &420DB)

Purpose
To write a block of bytes to STEbus memory address space from the computer' s memory.

Parameters
R0 - expansion card slot number
R1 - start STE memory address
R2 - start of computer' s memory address
R3 - number of bytes to write

Example

```
10  computer_slot% = 1
20  STEmemadr% = &A0000
30  DIM computer_memadr% 2048
```

Parameters
R0 - expansion card slot number
R1 - interrupt source (0-9)
(0 - ATNRQ0*)
.
.
(7 - ATNRQ7*)
(8 - CYCERR*)
(9 - SYSRST*)
R2 - address of interrupt service routine
R3 - value to be passed in R12 when interrupt service routine called

STEbus_releaseIRQvector (SWI &420DD)

Purpose
To release an STEbus IRQ vector.

Parameters
R0 - expansion card slot number
R1 - interrupt source (0-9)
(0 - ATNRQ0*)
.
.
(7 - ATNRQ7*)
(8 - CYCERR*)
(9 - SYSRST*)
R2 - address of interrupt service routine
R3 - value to be passed in R12 when interrupt service routine called

STEbus_claimFIQ (SWI &420DE)

Purpose
To claim the FIQ interrupt, move the FIQ handler to &0000001C, change to FIQ mode and initialise FIQ registers, return to SVC mode and set the FIQ interrupt enable bit in the control latch on the expansion card.

Parameters
R0 - expansion card slot number
R1 - address of FIQ handler
R2-R5 - values to be passed in FIQ Regs R10-R13

STEbus_releaseFIQ (SWI &420DF)

Purpose
Clear the FIQ interrupt enable bit in the control latch on the expansion card and release the FIQ interrupt.

- each byte on a 32 bit word boundary)
R1 - address of status register (read) and control latch (write) on the computer expansion card
R2 - address of status register (read) and control latch (write) on the STEbus board
R3 - address of location which when written to clears CYCERR and SYSRST interrupts
R4 - address of Attention Request status register on the STEbus board
R5 - address of Address latch on the STEbus board

STEbus_adrmemADR (SWI &420E1)

Purpose
To return the address of the memory copy of the STEbus memory space page.

Parameters
R0 - expansion card slot number

Results
R1 - address of the memory copy of the STEbus memory space page

STEbus_checkpresent (SWI &420E2)

Purpose
To determine whether an STEbus interface is present.

Parameters
R0 - expansion card slot number

Results
R1 - >0 expansion card present, 0 expansion card not present

STEbus_rdmodwrIO (SWI &420E3)

Purpose
To perform an indivisible read modify write of I/O address space, requests for control of the bus are locked during the execution of this SWI. The new value = (old value AND R3) EOR R2.

Parameters
R0 - expansion card slot number
R1 - STE IO address
R2 - EOR mask
R3 - AND mask

Example

```
 10  computer_slot% = 0
20  IOadr% = &3C0
30  EORmsk% = %10110000
40  ANDmsk% = %00001111
50  SYS "STEbus_rdmodwrIO", computer_slot%,EORmsk%,ANDmsk%
```

Example

```
10  computer_slot% = 0
20  memadr% = &3A000
30  EORmsk% = %10000000
40  ANDmsk% = %01111111
50  SYS "STEbus_rdmodwrMEM",computer_slot%,memadr%,EORmsk%,ANDmsk%
```

the new value of memory address &3C0 = (old value AND %00001111) EOR %10110000.

```
    16K                 A         A         A
+   32K                 B         A         A
    64K                 B         B         A
    128K                B         B         B
```

4.2 STEbus BOARD

Bus request

```
+   S1 A        local request to on board arbiter
    S1 B        request on BUSRQ0*
    S1 C        request on BUSRQ1*
```

Bus acknowledge

```
+   S2 A        local ackonwledge from on board arbiter
    S2 B        acknowledge from BUSAK0*
    S2 C        acknowledge from BUSAK1*
```

Master Mode (Release on request enable)

```
+   S3 A        release on BUSRQ0
+   S3 B        release on BUSRQ1
```

On board arbiter connections

```
+   S4 A        connect arbiter to BUSAK0
+   S4 B        connect arbiter to BUSAK1
```

Link Selection Options S1 S2 S3 and S4 Summary

On board arbitration

Bus request - local request (as supplied)

```
S1 A fitted    S2 A fitted    S3 A fitted    S4 A fitted
S1 B open      S2 B open      S3 B fitted    S4 B fitted
S1 C open      S2 C open
```

Bus request on BUSRQ0*

```
S1 A open      S2 A open      S3 A open      S4 A fitted
S1 B fitted    S2 B fitted    S3 B fitted    S4 B fitted
S1 C open      S2 C open
```

Bus request on BUSRQ1*

```
S1 A open      S2 A open      S3 A fitted    S4 A fitted
S1 B open      S2 B open      S3 B open      S4 B fitted
S1 C fitted    S2 C fitted
```

External arbitration

Bus request on BUSRQ0*

S5 3A ATNRQ2* interrupts connected to the computer's IRQ
S5 3B ATNRQ2* interrupts connected to the computer's FIQ
S5 4A ATNRQ3* interrupts connected to the computer's IRQ
S5 4B ATNRQ3* interrupts connected to the computer's FIQ
S5 5A ATNRQ4* interrupts connected to the computer's IRQ
S5 5B ATNRQ4* interrupts connected to the computer's FIQ
S5 6A ATNRQ5* interrupts connected to the computer's IRQ
S5 6B ATNRQ5* interrupts connected to the computer's FIQ
S5 7A ATNRQ6* interrupts connected to the computer's IRQ
S5 7B ATNRQ6* interrupts connected to the computer's FIQ
S5 8A ATNRQ7* interrupts connected to the computer's IRQ
S5 8B ATNRQ7* interrupts connected to the computer's FIQ
S5 9A CYCERR* (Cycle Error) connected to the computer's IRQ
S5 9B CYCERR* (Cycle Error) connected to the computer's FIQ
S5 10A SYSRST* interrupts connected to the computer's IRQ
S5 10B SYSRST* interrupts connected to the computer's FIQ

SYSCLK select

\+ S6 SYSCLK from the STEbus board (connects the STEbus SYSCLK bus signal to the 16 MHz signal on the STEbus board)

SYSRST select

\+ S7 SYSRST* from the STEbus board (connects the STEbus SYSRST* bus signal to the open collector buffered output from the control latch on the STEbus board)

STEbus board Attention Request status register	R	&03343008	&03347008	&0334B008	&0334F008
STEbus board clear CYCERR and SYSRST interrupts	W	&03343004	&03347004	&0334B004	&0334F004
STEbus board status register/latch control	R/W	&03343000	&03347000	&0334B000	&0334F000
Computer expansion card status register/latch control	R/W	&03342800	&03346800	&0334A800	&0334E800
Computer expansion card ID	R	&033C0000	&033C4000	&033C8000	&033CC000
STEbus I/O or memory space	R/W	from &03000000 to &03003FFC	&03004000 &03007FFC	&03008000 &0300BFFC	&0300C000 &0300FFFC

Country (byte 7) = 0 UK