Programmers Viewpoint PCI-bus:

PLX Base addresses:

Address Space	Function
PCIBAR0	Memory mapped configuration Registers
PCIBAR1	IO mapped configuration Registers
PCIBAR2	Local Address Space 0
PCIBAR3	Local Address Space 1
PCIBAR1	Local Expansion ROM

Table 1: PLX Base addresses seen from the PCI bus

How to generate a RST_n on the Local bus:

It is possible to activate the system-reset signal 'RST_n' on the add-on bus to reset for example the SHARC.
This is done by setting bit 30 of the CNTRL Register within the Runtime registers.
The PCI offset of the CNTRL register is 0x6C.

Local Address Space definitions:

Local Address Space	Device	PLX Mode	Burst Support
Space 0	SHARC	Direct Slave / DMA	Yes
Space 1	Not Used	N.A.	N.A.

Table 2: PLX Base addresses seen from the Local bus

Data Formats Local Address Space 0:

- SHARC

Bits	Read/Write	Remarks
D31 .. D0	Read/Write	1

Table 3: Sharc Data Format for Local Address Space 0

Remarks:

1: The address supplied to the SHARC will be the PCI address shifted 2 places right. This results in the following memory map:

PCI address lines	PCI ADR	SHARC address region	SHARC ADR	SHARC address lines
A31..A21 = 'X' : A20..A2 = ADR	0X00000000 : 0X0007FFFC	IOP registers	0X00000000 : 0X0001FFFF	A31 .. A19 = '0' : A18..A0 = ADR
"	0X00080000 0X000FFFFC	Normal Word Address	0X00020000 0X0003FFFF	"
"	0X00100000 : 0X001FFFFC	Short Word Address	0X00040000 : 0X0007FFFF	"

Table 4: PCI <-> Sharc Address Remap

Direct Slave Mode or DMA mode under control of the PLX chip is not the most effective way of transporting data.
The Sharc is accessed in an asynchronous way. This means the PLX chip has got to wait ‘N’ clock cycles for the ‘Redy’ signal of the Sharc.
Writing to the Sharc will cost at least N + 3 cycles and a reading from the Sharc will cost at least N + 4 Cycles.
The Sharc is usually ‘Redy’ within 0 clock cycles when data is written to the Sharc (provided the EPB is not full). This means that writing data to the Sharc can be done every 3 clock cycles.
On the other hand, when data is read from the Sharc, it usually takes 5 clock cycles before the Sharc sends ‘Redy’. This means that data can be read from the Sharc, every 9 clock cycles.
It is more effective to use the PLX chip in Direct Master Mode with the DMA controller of the Sharc, when blocks of data have to be moved to or from the Sharc. In this case full speed data transfer can be done.

Data Formats Local Address Space 1:

Local Address Space 1 is not use.

Programmers Viewpoint SHARC:

Sharc Memory regions:

Memory Region	Device	PLX Mode	EBxWS	EBxWM	Remarks
MS0	PLX	Direct Master	000	00	1, 2
MS1	PLX Internal Registers	Local Bus Access To Internal Registers	000	00	1, 2
MS2	Sharc Control Status Register/ Burst Counter	N.A.	100	01	1, 3
MS3	S-LINK Interface	N.A.	000	00	1, 2

Table 5: Sharc Memory regions

Remarks:
1: EBxWS and EBxWM are sub-patterns of the Wait register described in chapter 5.4.4.1 of the SHARC user's manual.
2: EBxWS = 000 means 0 Wait, 0 Hold Cycles
EBxWM = 00 External Acknowledge only
3: EBxWS = 100 means 4 Wait, 1 Hold Cycle
EBxWM = 01 Internal Wait states only

Data Formats:

- MS0 PLX:

The PLX operates in Direct Master Mode. In this mode, the Sharc is the initiator and can read or write data to or from the PCI bus through the PLX chip. Transfer can be either single cycle or burst.
To initiate a single transfer, write value ‘1’ to the burst counter (see MS2: Sharc Control and Status Registers).
To initiate a burst transfer, write value ‘N’ to the burst counter (see MS2: Sharc Control and Status Registers) where ‘N’ is the number of data words to transfer.
Watch out! Read the section “Possible bus conflict”.

- MS1 PLX Internal Registers:

In this mode, the Sharc is the initiator and can read or write data to or from the PLX internal registers of the PLX chip to set up, for example, a DMA transfer.
Transfer can be either single cycle or burst.
To initiate a single transfer, write value ‘1’ to the burst counter (see MS2: Sharc Control and Status Registers).
To initiate a burst transfer, write value ‘N’ to the burst counter (see MS2: Sharc Control and Status Registers) where ‘N’ is the number of data words to transfer.
Watch out! Read the section “Possible bus conflict”.

- MS2 Sharc Control Status Register:

For all *EVEN* addresses in the MS2 region, the Sharc Control Status Register is accessed. The data format of the Sharc Control Status Register is described in table 6. Watch out! Bits 6..7 read a '1' when the corresponding signal is asserted (this means the signal itself will be a '0').
For all *ODD* addresses in the MS2 region, the 16 bit Burst Counter is accessed (table 7). The Burst Counter must be programmed when a Direct Master burst transfer or a burst transfer to the internal registers of the PLX chip is done. Write the number of data words to be transferred into the burst counter. The Burst Counter automatically decrements and as soon as the value of the counter reaches '1', a BLAST (Burst LAST) signal is generated for the PLX chip to indicate the last data word is being transferred.
When the Burst Counter reaches '1' it keeps this value. Therefore it is not necessary to reprogram the Burst Counter when a number of single cycles is done.

Data	Description	Write	Read
3..0	S-LINK LRL bits	X	S-LINK LRL(3..0)
4	Enable/Disable LRL Change Interrupt (Sharc IRQ0)	'0': Disable Interrupt '1': Enable Interrupt	'0': Interrupt Disabled '1': Interrupt Enabled
5	Clear LRL Change Interrupt (Sharc IRQ0)	'0': No action '1': Clear Interrupt	0
6	LDOWN	X	'1': S-LINK LDOWN_n asserted
7	LSERR	X	'1': PCI9054 LSERR_n asserted
8	LINT	X	'1': PCI9054 LINT_n asserted
31..9	X	X	0

Table 6: Sharc Control Status Register data format

Data	Description	Write	Read
31..0	Burst Counter	xxxxnnnnH	0000nnnnH

Remarks:
'n' is any hexadecimal number
'x' is don't care

Table 7: Burst Counter data format

- MS3 S-LINK:

For all *EVEN* addresses in the MS3 region, the Sharc writes S-LINK *DATA* words (S-LINK control line LCTRL# = '1').

For all *ODD* addresses in the MS3 region, the Sharc writes S-LINK *CONTROL* words (S-LINK control line LCTRL# = '0').
The data format of the Sharc Control Status Register is described in table 6.

Data	Description	Write	Read (See Remark)
31..0	S-LINK LD(31..0)	Write data to S-LINK	X

Table 8: Sharc to S-LINK data format

Remark: The S-LINK can not be read in memory region MS3. To read the LRL(3..0) lines of the S-LINK, see the Sharc Control Status Register.

Possible bus conflict:

While there is a burst transfer pending (address spaces MS0 and MS1) it is forbidden for the Sharc to inserts another cycle during the burst transfer or a bus collision may follow.
It is possible to prevent the Sharc from generating other cycles while a burst transfer is pending by giving the Sharc IO-Processor priority over the Core-Processor. In this case a pending DMA transfer is always completed first before the Sharc does another access. Therefore a burst transfer will always complete first.
The IO-Processor can be given priority by programming the EBPR bits in the SYSCON register of the Sharc (see page E-24 of the Sharc manual).
If the EBPR bits are programmed '10' then the "PM and / or DM bus accesses will be delayed until all pending IO bus accesses are completed".

When the Sharc inserts another cycle within a burst transfer, the following situations can exist:

1. If the Sharc tries to *read* address space MS2 or MS3 while a Direct Master burst (*read* or *write*) is pending then the databus will not be enabled to prevent a bus conflict. Therefore the result of such a read cycle **will not be valid!**
2. If the Sharc tries to *write* address space MS2 or MS3 while a Direct Master burst *write* is pending then this cycle will complete since there is no danger of a bus conflict. However, the burst transfer is terminated with brute force by the hardware so the last data transfer **will not be valid!**
3. If the Sharc tries to *write* address space MS2 or MS3 while a Direct Master burst *Read* is pending then **a bus conflict will be the result!**

In all cases:
As soon as is detected that the Sharc inserts another cycle within a burst transfer, the Burst Counter will be reset and the burst transfer will be terminated with brute force by the hardware (a BLAST is generated to the PLX). When the burst transfer continues those cycles will be single cycles because the Burst Counter was reset.
Whenever a burst is pending (*read* or *write*), cycles to MS3 (*read* or *write*) will be acknowledged as soon as possible to prevent long periods of bus contention. This means that *writing* to MS3 address space while a *write* burst is pending will output S-

LINK data as normal although data may be lost if the S-LINK fifo is full (LFF# is active) since the MS3 write cycle is acknowledged immediately and not hold off.

Sharc Flags:

Table 9 gives an overview of the Sharc flags.
Flag-0 must be configured as an output and resets the S-LINK if flag-0 is set to '0'. The S-LINK will force LDOWN# low until the initialisation phase of the S-LINK is complete. LDOWN# will then go high again. The S-LINK is now up and running again.
Flag-1 must be configured as an output and sets the S-LINK in test mode if flag-1 is set to '0'.
Flag-2 must be configured as input. It is connected to the PCI9054 DMPAF pin. The DMPAF signal is asserted as soon as the Direct Master Write FIFO in the PCI9054 is almost full.
Flag-3 must be configured as an output and sets a LED on or off. The LED will burn if Flag-3 = '0'.

Flag Number	Description
0	S-LINK Reset
1	S-LINK Test
2	PCI9054 Direct Master Write FIFO almost full status
3	Red LED

Table 9: Sharc Flags

Sharc Interrupts:

Table 10 gives an overview of the Sharc interrupt lines.
Interrupt-0 is asserted if it is enabled (see: MS2, Sharc Control Status Register) and the LRL lines of the S-LINK change.
Interrupt-1 is asserted as soon if a burst transfer is pending (MS0 and MS1) and a transfer is done to MS2 or MS3. **Note that this should never occur!** (see: Possible bus conflict).
Interrupt-2 is asserted when the S-LINK signal LDOWN_n is asserted, **or** the PCI9054 LINT_n signal is asserted, **or** the PCI9054 LSERR_n signal is asserted. Which of the signals is causing the interrupt can be read from the Sharc Control Status Register (See MS2 Sharc Control Status Register)

IRQ Number	Description
0	LRL-Change
1	Interrupted Burst Transfer
2	LDOWN or LSERR or LINT

Table 10: Sharc Interrupts

Software Considerations:

Sharc:

Configure:
Flag0 as an output
Flag1 as an output
Flag2 as an input
Flag3 as an output

PCI 9054:

DP(3:0) Not Connected:
Disable LSERR# for parity error's (PCI9054 datasheet page 255 and 209, INTCSR[7:6].

Burst forever:
PCI9054 datasheet page 107; Burst = '1' en Bterm = '1', BTERM# = '1' pulled-up.

Direct Master with Master/Target abort:
Don’t do a burst transfer if it is not known if the PCI target is present (PCI9054 datasheet page 129).

Deadlock:
BREQo is connected to the SBTS_n pin of the Sharc. Read Page 135 of the PCI9054 datasheet.

DMA:
Don’t use Demand mode DMA. DMAMODE0[12]='0' (PCI9054 datasheet page 259).
USERo/DREQ0#/LLOCKo# pin becomes USERo of LLOCKo# and is not used.
USERi/DACK0#/LLOCKi# pin becomes USERi of LLOCKi# and is not used.
EOT# disable. DMAMODE0[14]='0' (PCI9054 datasheet page 259).
DMPAF/EOT# pin becomes DMPAF and is connected to Sharc flag 2.

LINT_n:
A PCI interrupt can be generated by using the doorbell register L2PDBELL (PCI9054 datasheet page 254).
The LINT_n pin should be configured as an output (INTCSR[11] PCI9054 datasheet page 255). LINT_n is connected to IRQ2_n of the Sharc.

USERo_n:
The USERo_n pin of the PCI 9054 is driving a LED. CNTRL[19] should be written ‘1’. Writing a ‘0’ to CNTRL[16] will make the LED burn. See PCI 9054 datasheet page 257.

BIGEND: The PCI 9054 should be configured as Little Endian. Therefore the BIGEND# pin is ‘1’. The BIGEND[1] should be ‘0’ (PCI 9054 datasheet page 245).