Functional test checklist for the ShaSLINK

Slreset
ShaSLINK System reset via the PCI-bus

Shslintr
Watch out! Slidas set to 2 (“Switches “)
LDOWN interrupt may cause more then one interrupt because of jitter in the LDOWN switch.

Shslreg
Test the ShaSLINK Registers

Ppshsl
Flag3 -> Red LED
Flag0 -> S-LINK Reset

Flag 0 -> ‘0’	-> Reset LED on Slidas on
Flag 0 -> ‘1’	-> Reset LED still on
LDOWN Switch	-> Nothing should happen but normally reset goes off because of jitter in the LDONW switch
LDOWN Switch back	-> Reset off

PP 0x00430000	Test S-LINK Data
PP 0x00430001	Test S-LINK Control
PP 0x00430000	Test LFF_n, Ppshsl program waits until LFF_n is switched back

PP 0x0041003B	Rd:	0x180**F**767E	
	Wr:	0x180**E**767E	Green LED turns on
	Wr:	0x180**F**767E	Green LED turns on

Use c:\PLX\PCISDK21\BIN\PLXMON98.exe.
Start Natalia’s common buffer program (IntEvt.exe) and get buffer base address.
For example buffer base is PCI 0x00692000.

Fill Natalia’s buffer or fill address space 0x00033000 on the SHARC and transfer.
Try PLX_DMA (ppshsl option 0x9, 0xC)
PCI_Source or PCI_Destination = Natalia’s buffer address (0x00692000).
SHARC_Source or Destination = 0x00033000

Set PCI Local configuration registers for direct master to PCI.
The SHARC base address for the MS0 space (PLX Direct Master mode) starts at address 0x00400000 and ends at address 0x004FFFFF. These are 32 bit addresses. So for the PLX the base address for Direct Master to PCI is 0x00400000 shifted << 2 which results in 0x01000000 (byte addresses). Therefore DMLBAM (Local bus base address register for Direct Master to PCI memory) should be 0x01000000. The software on the SHARC already sets this value.
The MS0 address range is 0x00410000 – 0x00400000 = 0x00010000 32 bit words. In bytes this is 0x00010000 shifted << 2 which is 0x00040000 (256 Kbytes). This means that bits 31..18 must be included in the address decoding so DMRR (Local Range register for Direct Master tot PCI) should be 0xFFFC0000.
When bits 31..18 are included in the address decoding then DMPBAM[31..16] (PCI base address re-map register for Direct Master to PCI memory) should be 0x0068 in

this example where the base address of Natalia’s buffer is 0x00692000. Note that the 9 turns into an 8 because bit 16 is not included in the address decoding.
Try ppshsl modes 0xA, 0xB and 0xD, 0xE. Sharc_DMA or Sharc_Core_DMA

Fill Natalia’s buffer or fill address space 0x00033000 on Sharc and transfer.
PCI Re-map = 0x00680000 thus we have to add 0x00012000 hex as offset. These are byte addresses so divide by 4 for word addresses so 0x12000/4 = 0x4800 hex
PCI_Source or PCI_Destination = Natalia’s buffer address (0x4800).
SHARC_Source or Destination = 0x00033000
Transfers can be seen on the PCI-Analyzer while triggering on address 0x00692000.

Shslout and Stest / Intest

Put the FC-SLINK boards between the ShaSLINK and the CRUSH
Run SHSLOUT –d on the ShaSLINK and STEST on the CRUSH to test the data integrity.
Run SHSLOUT –d on the ShaSLINK and INTEST on the CRUSH to test the transfer speed.

SHARC Linktest

Put a SHARC-link between the ShaSLINK and the CRUSH. Test the link in two directions, with a transfer rate of 20 MB/sec. This makes sure that the link is electrically correct.
Run sndr_sl –l0 –c1 on the ShaSLINK and recv_sl –l0 –c1 –d on the CRUSH. This will test the connection L0 -> L0. These programs can be found on ‘H:\n48\Sharc\ShaSLINK\Linktest’. Also test the following links:

ShaSLINK		CRUSH
L0	->	L0
L0	<-	L0
L1	->	L1
L1	<-	L1
L2	->	L2
L2	<-	L2
L3	->	L3
L3	<-	L3

Note that Link is the boot-link so this one can not be tested in this way. But since the SHARC does boot the change of a malfunction in link 4 is low. Link 5 is within the same cable as link 4. Therefore it is not possible to test link 5 since it is connected to a PCI-SHARC card which runs a server program.