APPLICATION NOTE 1

Application note about SMT398 Test Examples

SMT398

SUNDANCE MULTIPROCESSOR TECHNOLOGY LTD.

Revision History

Date	Comments / Changes	Author	Revision
18/08/2003		E.P	1.0.0
19/08/2003	Correction of Figure 1 Now T2C1<->T2C0 is right. (used to be T2C2 instead of T2C1)	E.P	1.1.0
27/08/2003	Addition of FPGA design and software detailed explanations	E.P	1.2.0
09/09/2003	Addition of folder and design structure for test examples	E.P	1.2.1
14/10/2003	Addition of SHB connectors letters on picture figure 2	E.P	1.2.2

Table of Content

Table of figures

Test Equipment.

IBM compatible PC with 3.3v rail on motherboard..

SMT376

SMT310Q, 4-slotTIM carrier.

SMT398

2 SHB Tester PCBs (Only available under request)

4 ComPort cables

SDB and Comport connections

Connect one SHB Tester on SHB A and SDB B.

Connect second SHB Tester on SHB C and SDB D.

Connect comport 0 of the SMT398 to comport 1 of the SMT398.

(This set up doesn't respect the ComPort connection rules defined in the SMT398 User Manual because the idea here is only to test the interconnect of the FPGA for the ComPort signals. No ComPort interface is implemented in the FPGA to manage this link, so no conflict can occur at reset).

Connect comport 3 of the SMT398 to comport 0 of the SMT376.

Connect comport 5 of the SMT398 to comport 1 of the SMT376.

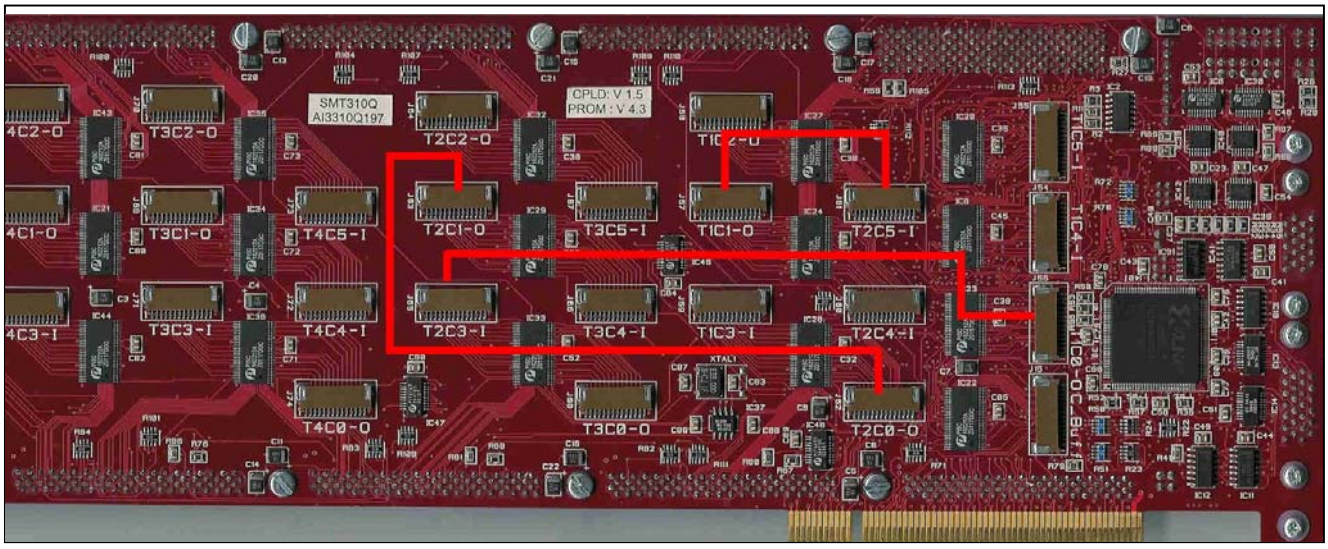

Figure 1: ComPort Connections

Figure 2: Tim modules position

Test Procedure.

After running a test, the test program comes back to the test menu so that you can select the next test to run.

This requires that the SMT398 supports the TIM CONFIG reset feature:

- Your CPLD code version must be 2.0 or above, indicated by a sticker on the CPLD. No sticker means that the CPLD code doesn't support this feature. (Description in the SMT398 User Manual Reset Control chapter).
- Your SMT398 PCB version must be version 3 (indicated on TOP of the SMT398 PCB). If your SMT398 is version 3 PCB but the CPLD is not version 2.1 or above, you can update your CPLD code with version 2.2.

If your board doesn't support this feature, you need to exit and you must run again the program to execute the next test.

This is to ensure that a reset is inferred to the boards before downloading another configuration in the FPGA. Otherwise, the program will hang.

Please read about Configuring the FPGA in the SMT398 User Manual and about the software routines in "Application note about bitstream download software.pdf" for more detailed explanations on the configuration protocol.

The test will ask you first to select the Virtex which is fitted on your SMT398.

Then you can select a test.

Tests 1 to 4

They make use of ComPort 4 on the SMT398 and of ComPort 1 on the DSP TIM.

They require to set the pipe T1C1 <-> T2C4 using the ComPort switches of the SMT310Q. See Figure 3: ComPort switches set up.

You don't need to unplug the link T1C1 <-> T2C5.

You can set up the switches using the 3L Diamond server (In Board\options\Configuration) or using the SMT6300 application called BoardInfo.

Tests 5 to 9

You need to deselect the ComPort switches setting that sets the pipe T1C1 <-> T2C4 .
They make use of ComPort 5 on the SMT398 and of ComPort 1 on the DSP TIM.

Tests 10 and 11

To be able to run tests 10 and 11 you will need to unplug the SMT310Q and change T2C1<->T2C0 to T1C4 <-> T2C0.

Test 12

The Leds flash at about 1 second interval in reverse order: 4, 3, 2 and 1.

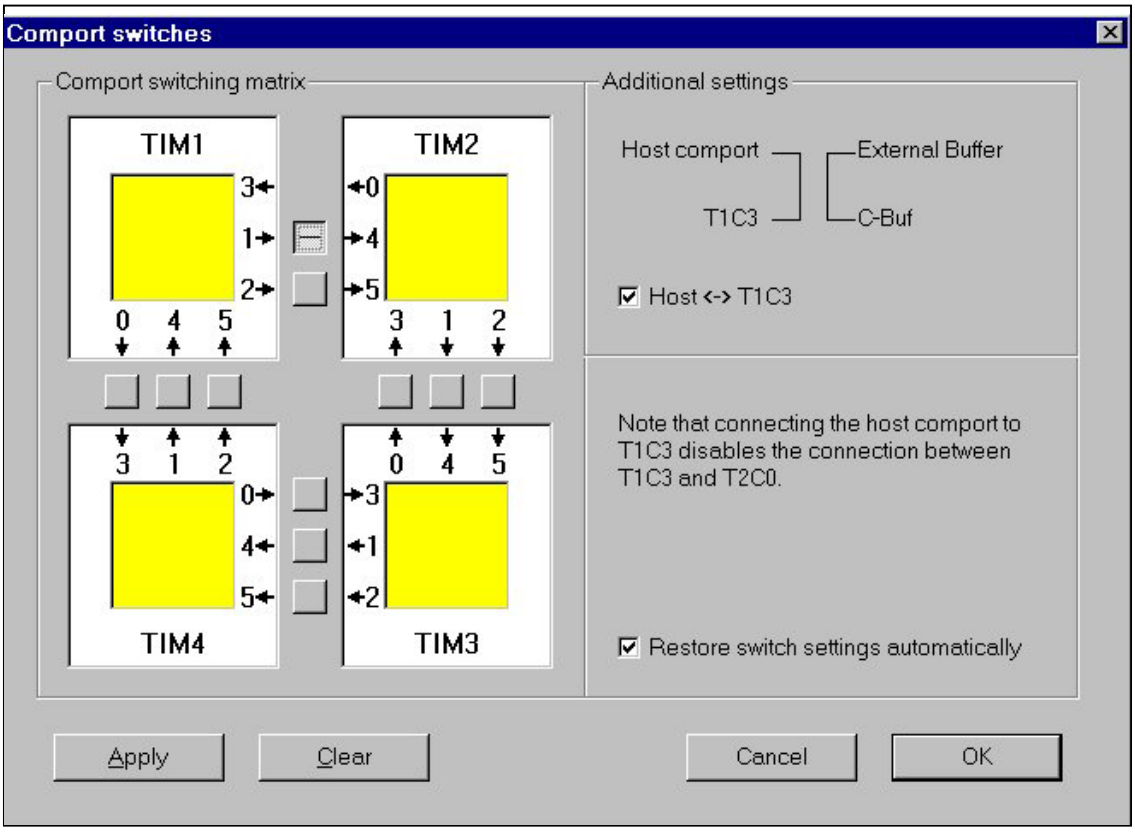

Figure 3: ComPort switches set up.

FPGA Test Designs

The SMT398 example designs are:

CxPTest: Implements a design to test connectors on the SMT398.

QDRTest Implements a design to test the QDR memory on the SMT398.

ZBTTest Implements a design to test the ZBT Memory on the SMT398.

All these designs have been implemented using ACTIVE-HDL from ALDEC tools and Xilinx ISE Foundation tool suite.

The designs folder structure respects the one imposed by a typical design in ACTIVE-HDL:

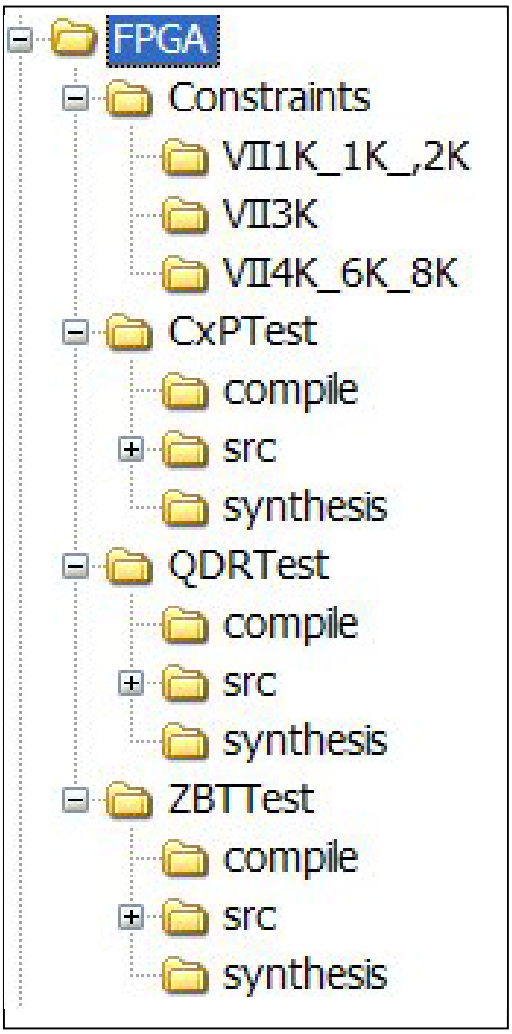

Figure 4: Folder structure

FPGA constraints files

See SMT398 User Manual.

Test designs folder structure

A design is stored in a folder whose name is the same as the design name.(ZBTTest, QDRTest or CxPTest)

Right underneath, the design description file (with the extension ADF), compile.cfg and bde.set are 3 files needed by ACTIVE-HDL to set up its project.

The content of the subfolders is as follows:

compile

VHDL files generated from block and state diagrams.

Src

Design source files:

Source Type	Description
VHDL Module	.vhd
VHDL Testbench	.vhd
VHDL Package	.vhd
Implementation Constraint file	.ucf
Block diagrams	.bde
State diagrams	.asf
SDF data file	.sdf
file with simulation results (waveform files)	.awf
Macro files to run simulations	.do

Table 1:Design source files

synthesis

The files used/produced by the synthesis and implementation tools from Xilinx.

Source Type	Description
Project Notebook	.npl
VHDL Module	.vhd
VHDL Package	.vhd
Implementation Constraint file	.ucf
Synthesis netlist	.ngc
SDF data file	.sdf
Place and route report	.par
Pad report to check pin assignement	.pad
FPGA bitstream	.bit

Table 2: Synthesis and Implementaton tools source files

All designs contain the following subfolders under the **Src** folder:

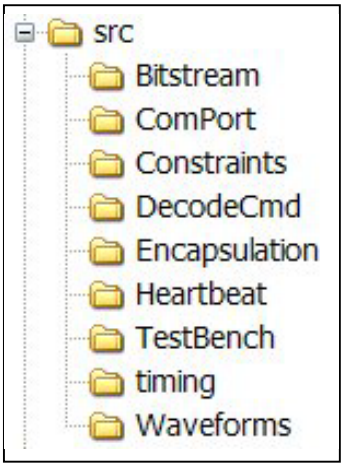

Figure 5: Src subfolders

Folder	Description
Bitstream	You can copy your bitstream in this folder
ComPort	Contains the ComPort Interface core
Constraints	Contains the constraints files needed for the design implementation. • VII1000 is for XC2V1000, XC2V 1500 and XC2V2000. • VII3000 is for XC2V3000, XC2V4000 and XC2V6000.
DecodeCmd	Contains the Command decoding interface.
Encapsulation	Contains an Encapsulation design that instantiates the top level Top.vhd and connects it to a ComPort Interface and to the designs specific I/O peripheral model FOR SIMULATION only. Top.vhd is the highest level synthesisable.
Heartbeat	Contains a design that toggles about every seconds with a 50Mhz clock.
TestBench	Encapsulates the design Encaps.vhd to provide clock and reset stimuli. Also contains .do macro files for functional and Timing simulation.
Waveforms	file with simulation results (waveform files)

Figure 6: Src Subfolders

Test Designs Fundamentals

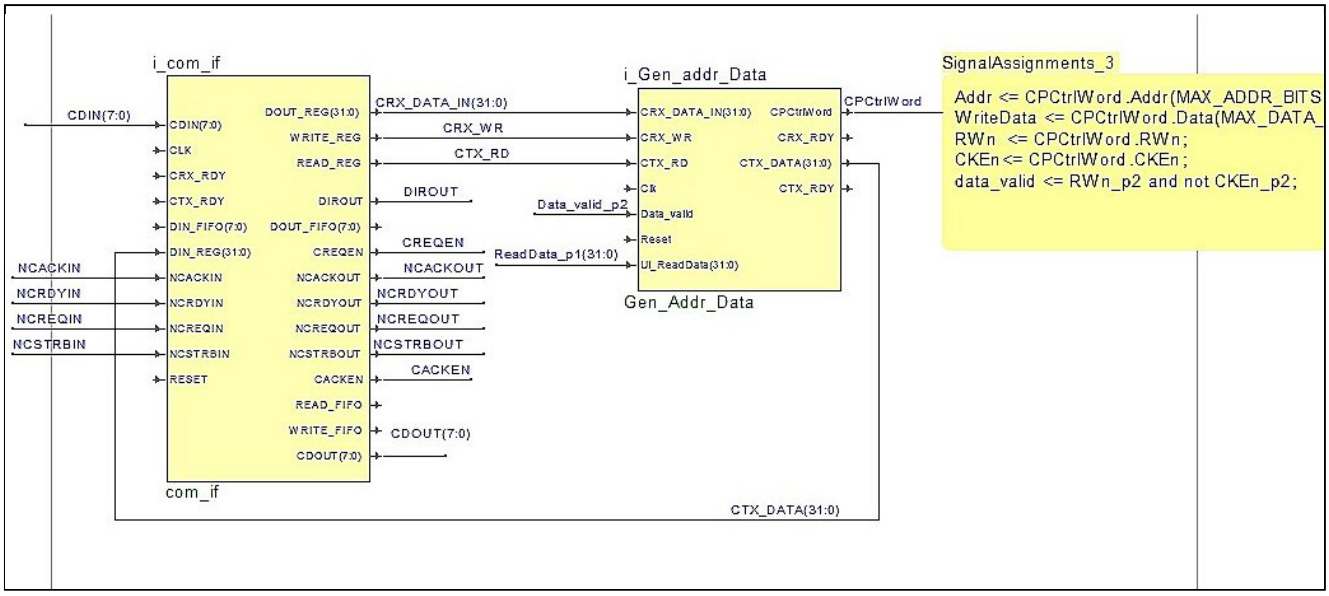

Figure 7:Common modules

The SMT398 example designs are:

CxPTest: Implements a design to test connectors on the SMT398.

QDRTest Implements a design to test the QDR memory on the SMT398.

ZBTTest Implements a design to test the ZBT Memory on the SMT398.

They behave the following way:

All of the test designs implement a ComPort Interface module, Com_if and a decoding module Gen_addr_data as shown in.Figure 7:Common modules

The ComPort interface module is used to communicate between the DSP TIM and the SMT398.

The ComPort Interface receives 32-bit words respecting a format defined in the package file: COMMPORTIO_pkg.vhd under CPControlWordType type, described in Figure 8: Write Command first word and Figure 9: Write Command second word.

The ComPort interface sends 32-bit words representing a data value resulting from a Read command sent by the DSP TIM. (It is the only the case when the SMT398 design ComPort interface transmits words)

The decoding module Gen_addr_data decodes the 32-bit words received by the ComPort interface into Address, Data, and Command fields according to the format defined in the package file: COMMPORTIO_pkg.vhd under CPControlWordType type.

The decoding module provides Address, Data, Read/Write, Clock Enable to the rest of the design.

Depending on the external device accessed, the designs implement a ZBT interface, a QDR Interface or a connector interface to which the decoding module interfaces.

The connector interface, in CxPTest can target ComPort connector pins or SHB pins.

The ZBT Interface Design implementation and the QDR Interface Design implementation are based on Xilinx free cores and adapted to the SMT398 architecture.

Write Command Format

A WRITE command received by the FPGA design is composed of 2 ComPort words:

Address and Write Command(note that the address can be padded with 0s if smaller than 31 bits. Fill LSbits first)

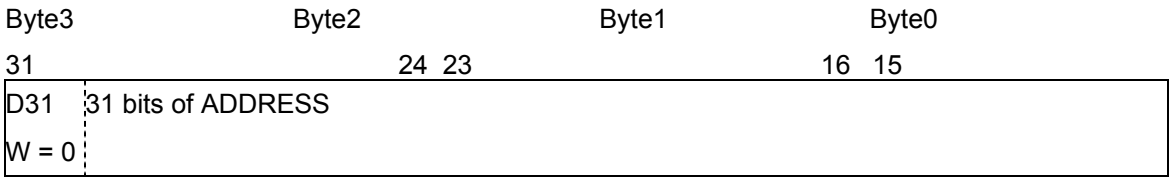

Figure 8: Write Command first word

Data: (note that the data can be padded with 0s if 16 bits wide Fill LSbits first)

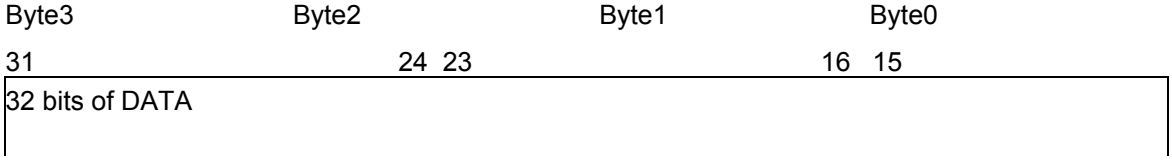

Figure 9: Write Command second word

Read Command Format

A Read command received by the FPGA design is composed of 1 ComPort word:

Address and Read Command

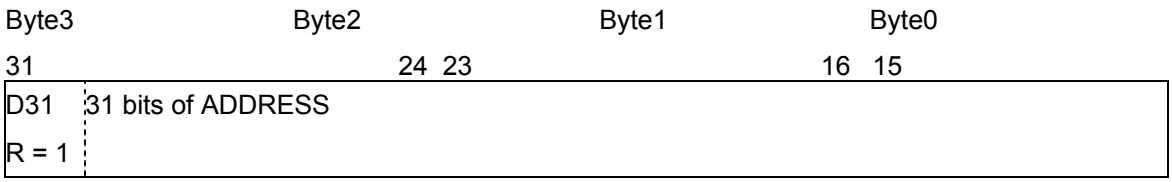

Figure 10: Read Command

The DSP TIM ReadMemory() or WriteMemory() functions are used to read and write data on the DSP side according to the format expected by the decoding module Gen_addr_data.

Software functions

Function: void memTestInit(UINT32 baseaddress, UINT32 nbytes, UINT32 nbits, UINT32 nCmd_port)

Description:

Initialise a memory region base address: baseaddress,

The size in bytes of the region to access: nbytes

The memory data bus size in bits: nbits

The ComPort which transmits the commands to the FPGA board.

Notes:

This function must be called before accessing one of the following functions.

Returns:

Nothing.

Function: extern UINT32 memTestDataBus()

Description:

Test the data bus wiring in a memory region by performing a walking 0 test and a walking 1 test at a fixed address within that region.

Notes:

The address and memory data bus size are set up by the MemTestInit function described above.

Returns:

NULL if the test succeeds.

A non-zero result is the first pattern that failed.

Function: memTestAddressBus()

Description:

Test the address bus wiring in a memory region by performing a walking 1's test on the relevant bits of the address and checking for aliasing.

This test will find single-bit address failures such as stuck high, stuck-low, and shorted pins.

Notes:

The base address, the memory data bus size and memory region size are set up by the function MemTestInit described above.

For best results, the selected base address should have enough LSB 0's to guarantee single address bit changes.

For example, to test a 64-Kbyte region, select a base address on a 64-Kbyte boundary.

Also, select the region size as a power-of-two--if at all possible.

Returns:

NULL if the test succeeds.

A non-zero result is the first address at which an aliasing problem was uncovered.

Function: UINT32 memTestDevice()

Description:

Test the integrity of a physical memory device by performing an increment/decrement test over the entire region.

In the process every storage bit in the device is tested as a zero and a one.

Notes:

The base address, the memory data bus size and memory region size are set up by the function MemTestInit described above.

Returns:

NULL if the test succeeds. Also, in that case, the entire memory region will be filled with zeros.

A non-zero result is the first address at which an incorrect value was read back.

Function: void ReadWriteMemInit(UINT32 nCmd_port)

Description:

Initialise the ComPort which transmits the commands to the FPGA board.

Notes:

This function must be called before accessing one of the next functions.

Returns:

nothing

Function: ReadMemory(VUINT32 address)

Description:

Sends over Comport “nCmd_port” the address of the Memory location selected to be read.

Receives 1 word on Comport nCmd_port containing the data stored at that memory address.

Notes:

The ComPort number used is set up by the function ReadWriteMemInit described above.

Returns:

The data read at memory location selected by address.

function: WriteMemory(UINT32 address, UINT32 data)

Description:

Performs a Write to Memory. The first 32-bit word represents the address where the 2nd 32-bit Word representing the data (if the word is smaller than 32 bits, the data must be on the LSbits) is written.

Notes:

The ComPort number used is set up by the function ReadWriteMemInit described above.

Returns:

nothing