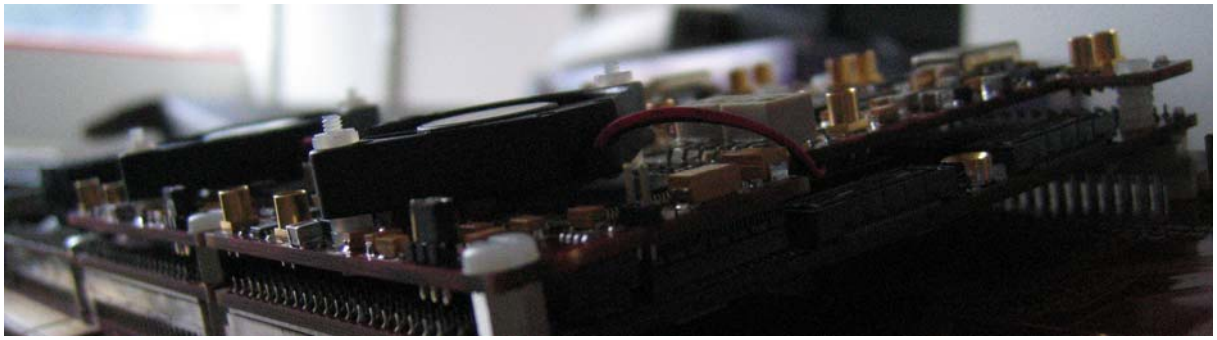
Unit / Module Description:	MIMO development station
Unit / Module Number:	MIMO development station
Document Issue Number:	2
Issue Date:	21/03/08
Original Author:	Jean-Philippe Arnaud

Application Note for MIMO development station



Sundance Multiprocessor Technology Ltd, Chiltern House, Waterside, Chesham, Bucks. HP5 1PS.

Revision History

Issue	Changes Made	Date	Initials
1.0	First release	05/09/06	JPA
2.0	Application ported to the Diamond IDE Updated to new version of ISE	21/03/08	JPA

Table of Contents

Table of Figures

1 Introduction

The Sundance MIMO development station (MIMO-DS) is a modular system made of multiple FPGAs and DSPs that can be used to prototype and test MIMO applications and algorithms. It accommodates four input channels and four output channels.

It is possible to extend it to add more channels and more processing power (DSP and FPGA).

The MIMO-DS is fully supported by 3L Diamond design environment. Diamond allows to easily and efficiently implement algorithms on the multiple DSPs and FPGAs available in the system.

2 Related Documents

2.1 Referenced Documents

2.2 Applicable Documents

SMT310Q user manual

SMT350 user manual

SMT368 user manual

SMT384 user manual

SMT395 user manual

3L Diamond

Sundance Help File

3 Acronyms, Abbreviations and Definitions

3.1 Acronyms and Abbreviations

MIMO: Multiple inputs multiple outputs

MIMO-DS: MIMO development station

3.2 Definitions

4 Quick start

4.1 What's in the kit?

The Sundance MIMO development station is made of a transmission system and a reception system. The following describes the composition of each system.

4.1.1 Transmission system

Name	Qty	Description
SMT310Q	1	PCI carrier board
SMT395-VP30	1	DSP board
SMT350	2	DAC board
SMT368	2	FPGA board
SMT516	1	SHB to SHB inter-module
SMT500	3	FMS cables
SMT507	2	MMCX—BNC analogue cable
MMCX—MMCX	4	MMCX—MMCX cables

4.1.2 Reception system

Name	Qty	Description
SMT310Q	1	PCI carrier board
SMT395-VP30	1	DSP board
SMT384	1	ADC board
SMT368	1	FPGA board
SMT351	2	Storage board
SMT511-320	2	SHB cable
SMT516	1	SHB to SHB inter-module
SMT500	3	FMS cables
SMT507	1	MMCX—BNC analogue cable

4.1.3 Software

Name	Qty
Diamond DSP	1
Diamond FPGA	unlimited

4.2 Assembling the boards

4.2.1 Transmission system

The picture shows the transmission system.

The SMT395-VP30 occupies the first TIM site of the SMT310Q carrier board. The two SMT350 are in TIM site 2 and 3.

An SMT516 connects the SHBD of SMT350#1 to SHBB of SMT350#2.

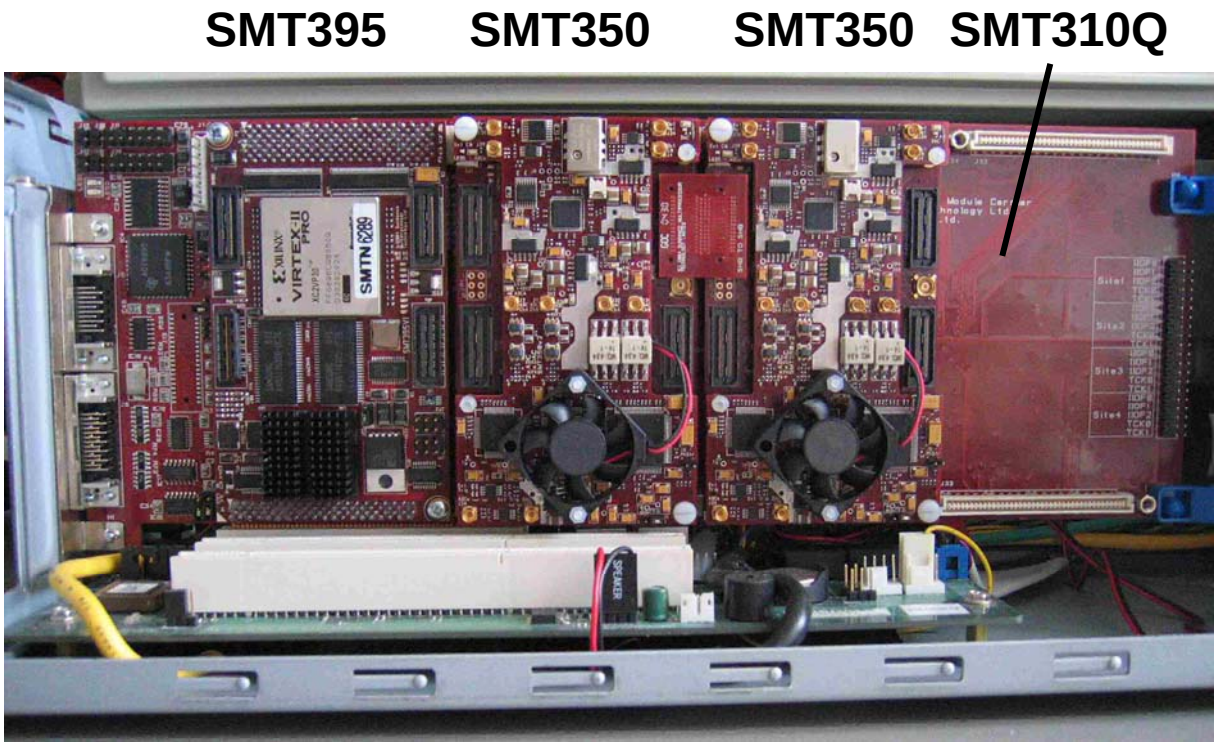


Figure 1: the transmitter system

The following comport are required:

- T1C0 – T2C3
- T1C2 – T2C4
- T1C1 – T3C3

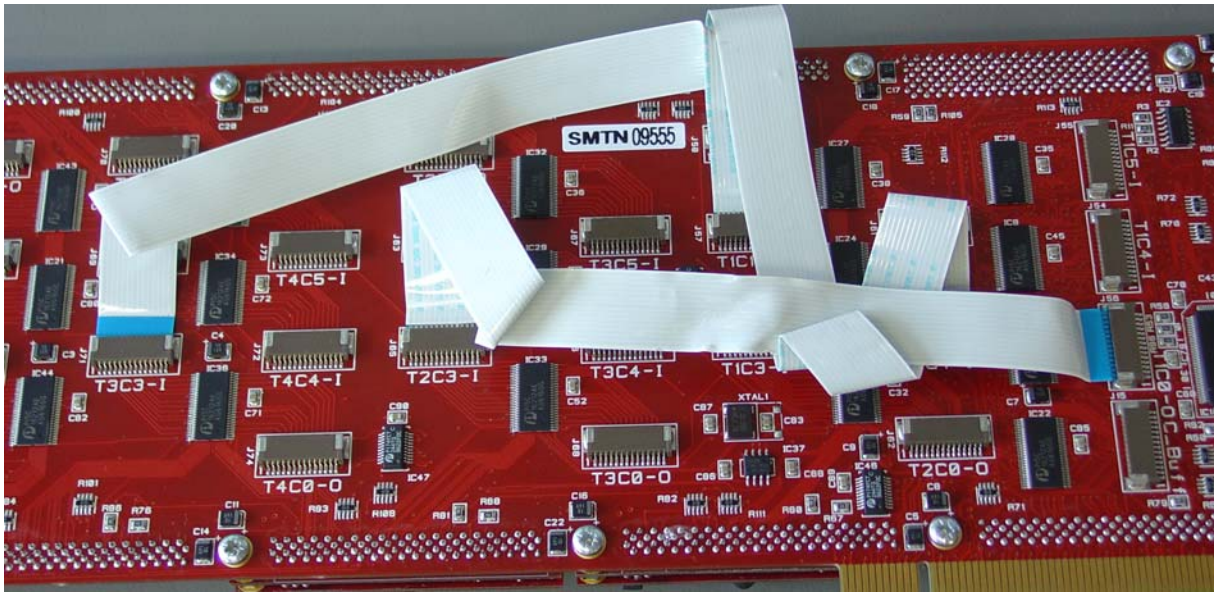


Figure 2: transmitter comport cables connections

4.2.2 Reception system

The SMT395 occupies the first TIM site of the SMT310Q. The SMT351 are plugged in TIM site 2 and TIM site 4. The SMT384 is plugged in TIM site 3.

The SHB connections are as follow:

- SMT395 SHBA – SMT351#1 SHBB
- SMT395 SHBB – SMT351#2 SHBB
- SMT384 SHBB – SMT351#1 SHBA
- SMT384 SHBD – SMT351#2 SHBA

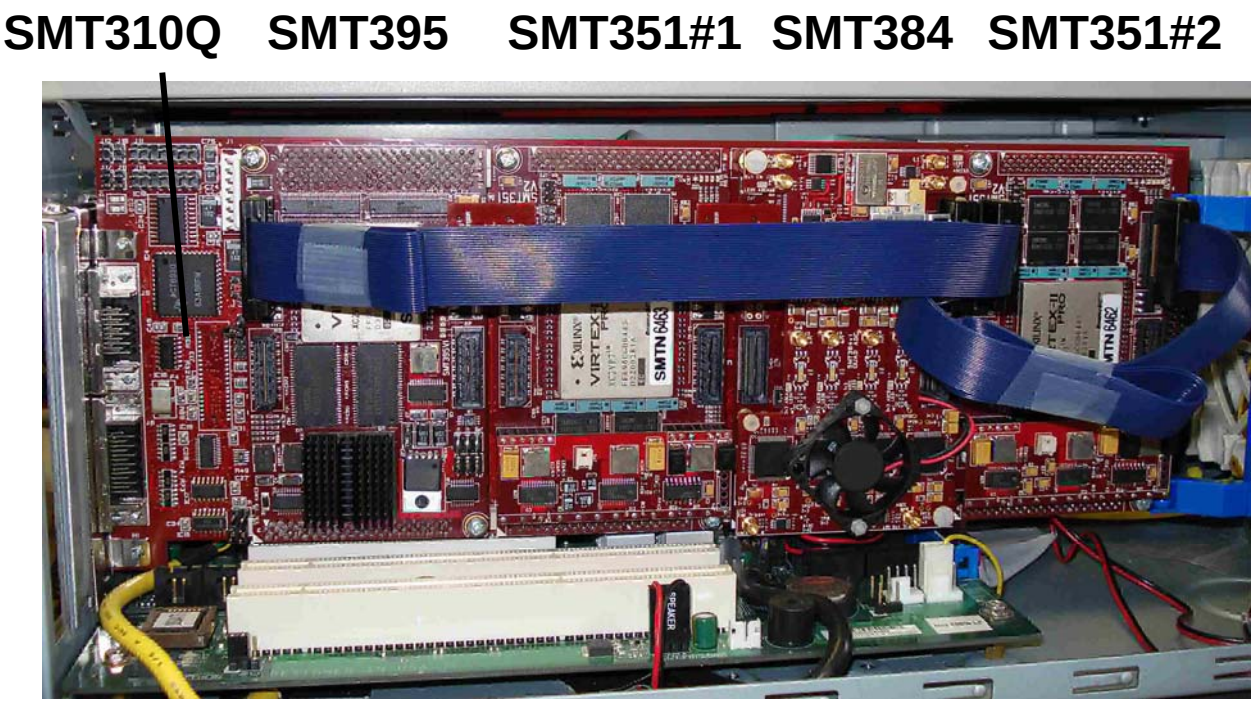


Figure 3: the reception system

Comport connections:

- T1C0 – T4C3 (using SMT6300 – SMTBoardInfo comport switches)
- T1C1 – T2C3
- T1C2 – T3C3
- T1C5 – T3C0

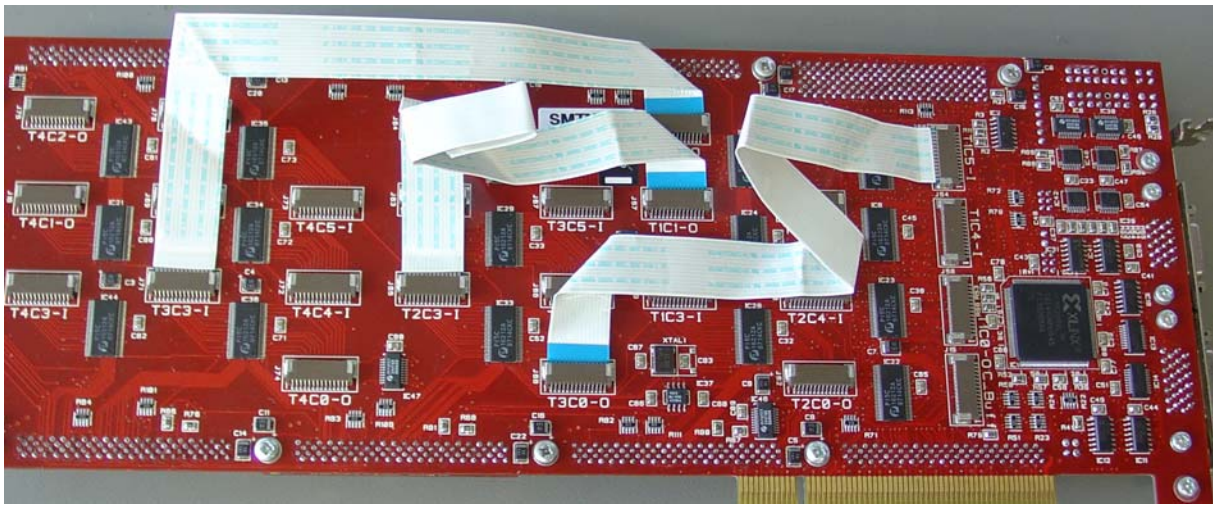


Figure 4: reception comport connections

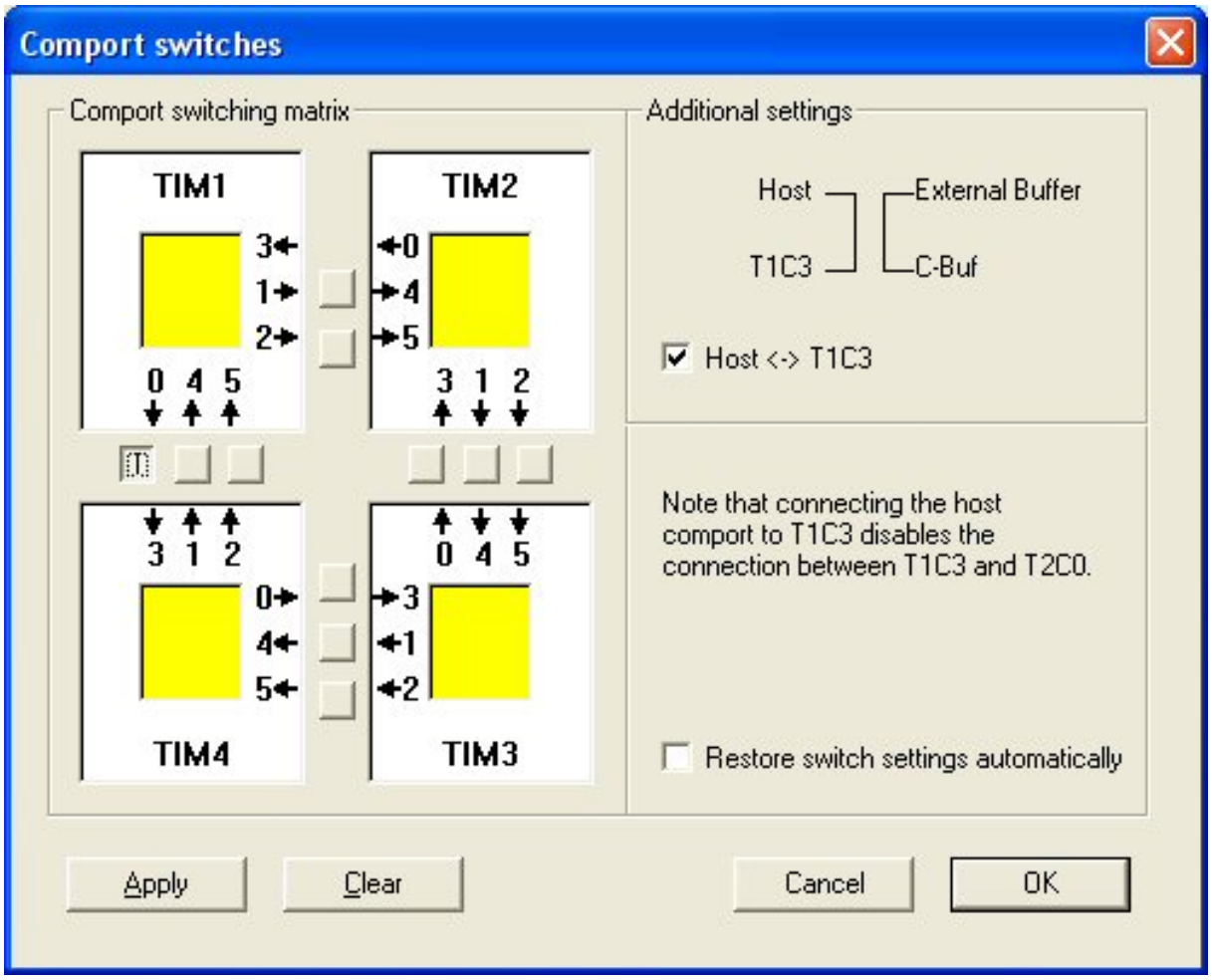


Figure 5: comport switches for reception system

4.3 Example

An example application illustrating the capabilities of the components in the system is provided. The example is developed using 3L Diamond design tool.

5 Overview

The diagram shows an overview of the MIMO-DS.

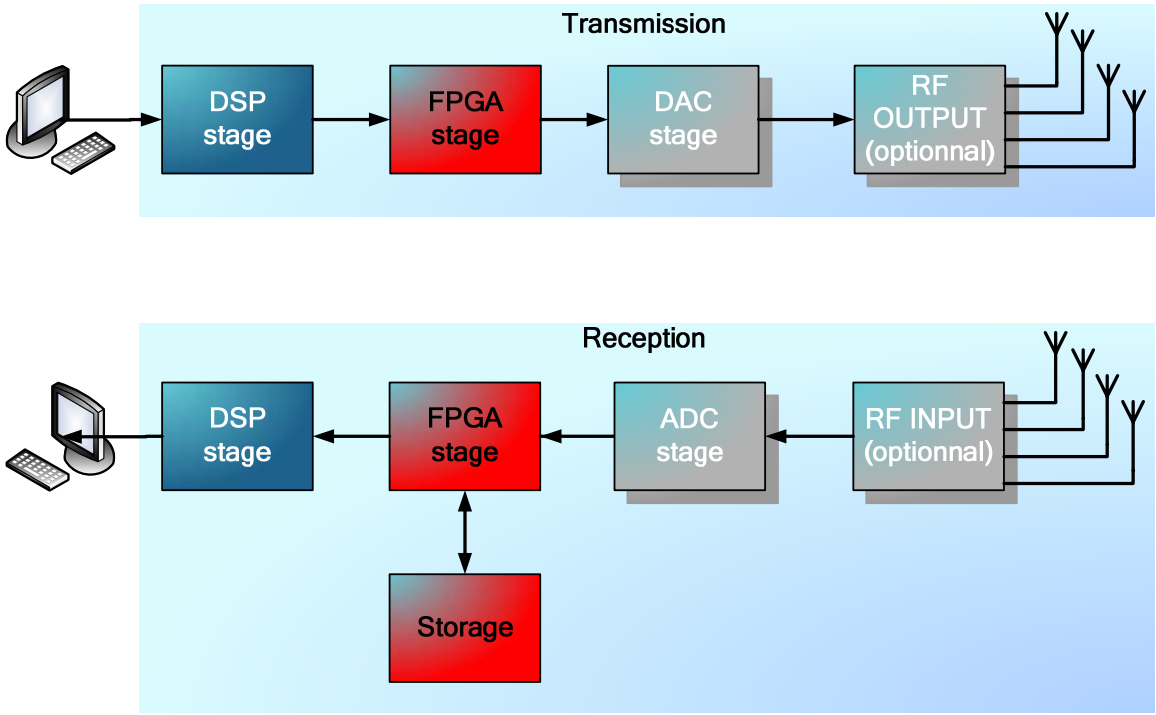


Figure 6: overview of the MIMO development station

Processing takes place in FPGAs and in DSPs and applications are developed using 3L Diamond design tool which allows easy implementation of the algorithm on the multiple DSPs and FPGAs of the system.

Information to be sent is converted in analog wave via a stage of Digital to Analog converters; while at reception, the signal is digitalized via a stage of Analog to Digital converters.

Storage is available to record the signals at reception.

Optionally, it is possible to transmit and receive in the RF spectrum.

6 The Sundance MIMO development station

The Sundance MIMO-DS is made of two systems.

The transmission system is made of four single DAC output channels. Three FPGAs and one DSP are available to implement the algorithms. The system is made of one SMT395 and two SMT350 modules held on a SMT310Q PCI carrier board. All four output channels start transmitting at the same time.

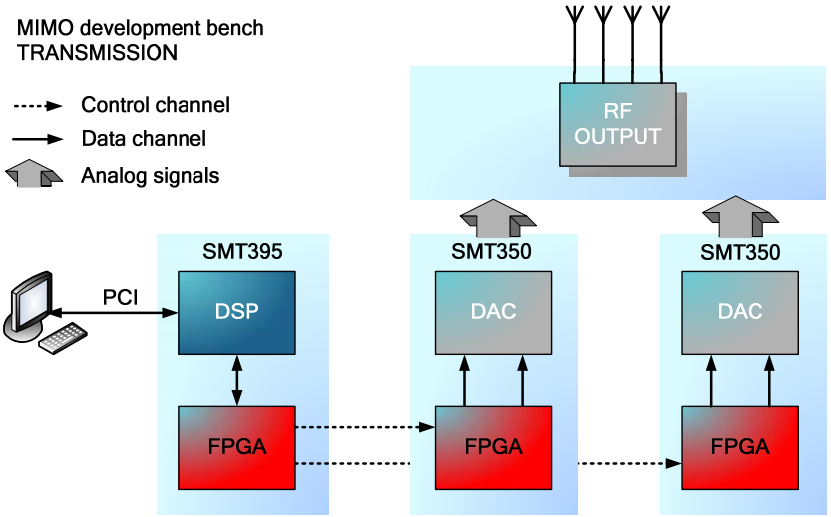


Figure 7: Transmission

The reception system is made of four single ADC input channels. Two FPGAs and one DSP are available for the data processing. The system is made of one SMT395, two SMT351 and one SMT384 modules held on a SMT310Q PCI carrier board. The ADCs are synchronized together to sample at the same time.

A total of two Giga bytes of memory are available to store the high speed data incoming from the ADC stage. Each ADC can store up to 128 Msamples.

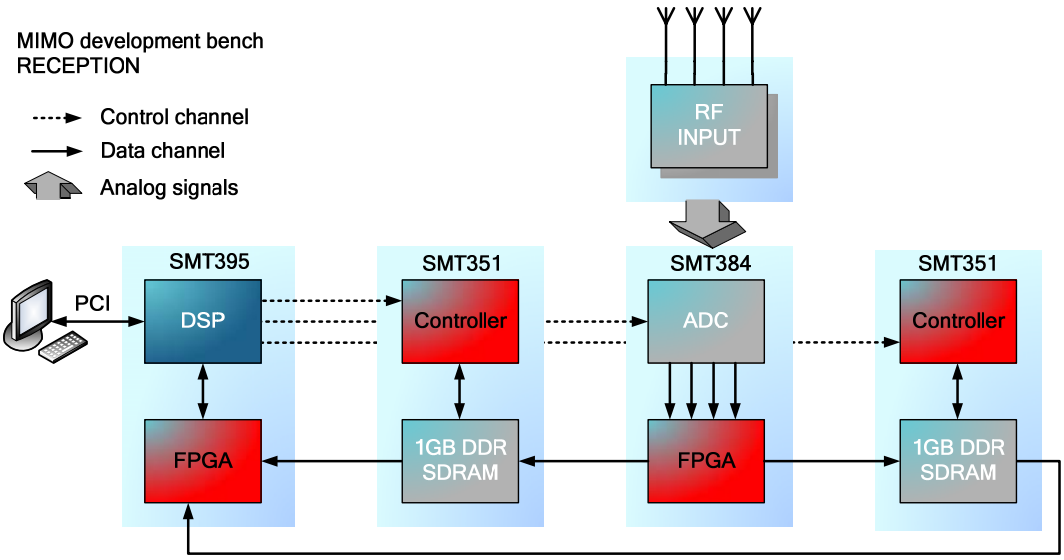


Figure 8: reception side

Optionally, both transmission and reception systems can be completed with the RF/IF front-end module SMT349.

7 Hardware

SMT395

SMT395 is based on the 1GHz 64-bit TMS320C6416T DSP, manufactured on the latest 90nm wafer technology and offers the highest fixed-point processing power ever. Sundance have not stopped here and have implemented a scalable solution using Xilinx Virtex II Pro FPGAs .

SMT350

The SMT350 is a single width expansion TIM that plugs onto the SLB base module SMT368 (Virtex4 FPGA) and incorporates two Texas Instrument Analog-to-Digital Converters (ADS5500) and one Texas Instrument dual-channel Digital-to-Analog Converter (DAC5686).

The SMT350 implements a comprehensive clock circuitry based on a CDCM7005 chip that allows synchronization among the converters and cascading modules for multiple receiver or transmitter systems as well as the use of external clocks. External Triggers are also available.

ADCs are 14-bit and can sample at up to 125 MHz. The DAC has a resolution of 16 bits and is able to update outputs at up to 500MHz. All converters are 3.3-Volt.

The Xilinx FPGA (Virtex 4) on the base module is responsible for handling data going/coming to/from one of the following destination/source: TI converters, Comport (TIM-40 standard), **S**undance **H**igh-speed **B**us (SHB), ZBTRAM Memory.

SMT384

The SMT384 is a single width expansion TIM that plugs onto an SLB base module SMT368 (Virtex-4 FPGA) and incorporates 4 Texas Instrument Analog-to-Digital Converters (ADS5500).

The SMT384 implements a comprehensive clock circuitry based on a AD9510 chip that allows synchronisation among the converters and cascading modules for multiple receiver systems as well as the use of an external reference clock.

ADCs are 14-bit and can sample at up to 125 MHz.

The Xilinx FPGA (Virtex-4) on the base module is responsible for handling data or control commands coming from the TI converters, Comports (TIM-40 standard), Sundance High-speed Bus (SHB).

SMT351

The SMT351 is a single width TIM compatible module populated with hi-density BGA package memory devices that is able to store up to 1GB of data at 400MB/s. The SMT351 can be cascade with more modules for even greater memory storage capability.

The data flows in and out of the module are controlled by a Xilinx Virtex-II Pro FPGA XC2VP7. Connections to the module are made by two Sundance High-speed Bus (SHB) connectors.

Control of the SMT351 is achieved by sending control words via comport.

SMT310Q

The SMT310Q is a quad (4) site module carrier developed to provide access to TIM Modules over the PCI bus running.

The card has an 'on-board' JTAG controller allowing Code Composer Studio and 3L Diamond applications to be used to debug/upload software to Modules.

8 A simple example

This simple example illustrates the functioning of the MIMO-DS. This example can be used as a starting point for users to develop their own applications on the MIMO-DS.

The transmitter is configured to "play" a predefined waveform.

The output channels are connected to the input channels.

On the reception side, the signals are sampled by the ADCs and the high speed data produced are stored in the memory of the SMT351 modules. In the example, each SMT351 stores 1GByte of data.

Data is processed by the FPGA and the DSP. The result of the processing and the original data are send to the host to be displayed. In this example, the DSP calculates the FFT of the signal and the FPGA lets the data go through without processing them it.

The data are also stored in files for further analysis. The content of the files can be used to check that the four channels are in phase.

The waveform used is a sinewave hard-coded in the FPGA of the transmitter. It would also be possible to send a "live" waveform from the DSP or to use the memory on-board the SMT350 to store various waveforms that will be sent to the DACs; but this is not part of this example.

The output channels are synchronized to start playing their waveforms exactly at the same time. To be able to do so the same sampling clock is provided externally using connector J29 of the SMT350. A trigger signal is routed from the SMT350#1 to SMT350#2 via the SMT511 to allow both boards to be synchronized.

The same sampling clock should be provided to receiver on connector J29 of the SMT384.

8.1 Opening the example

The MIMO-DS application is made of two Diamond projects; these projects need to be imported into your workspace before you can use them. Refer to the Diamond IDE user manual for the instructions on how to do that.

Once the projects are imported in the workspace it should look like the picture below.

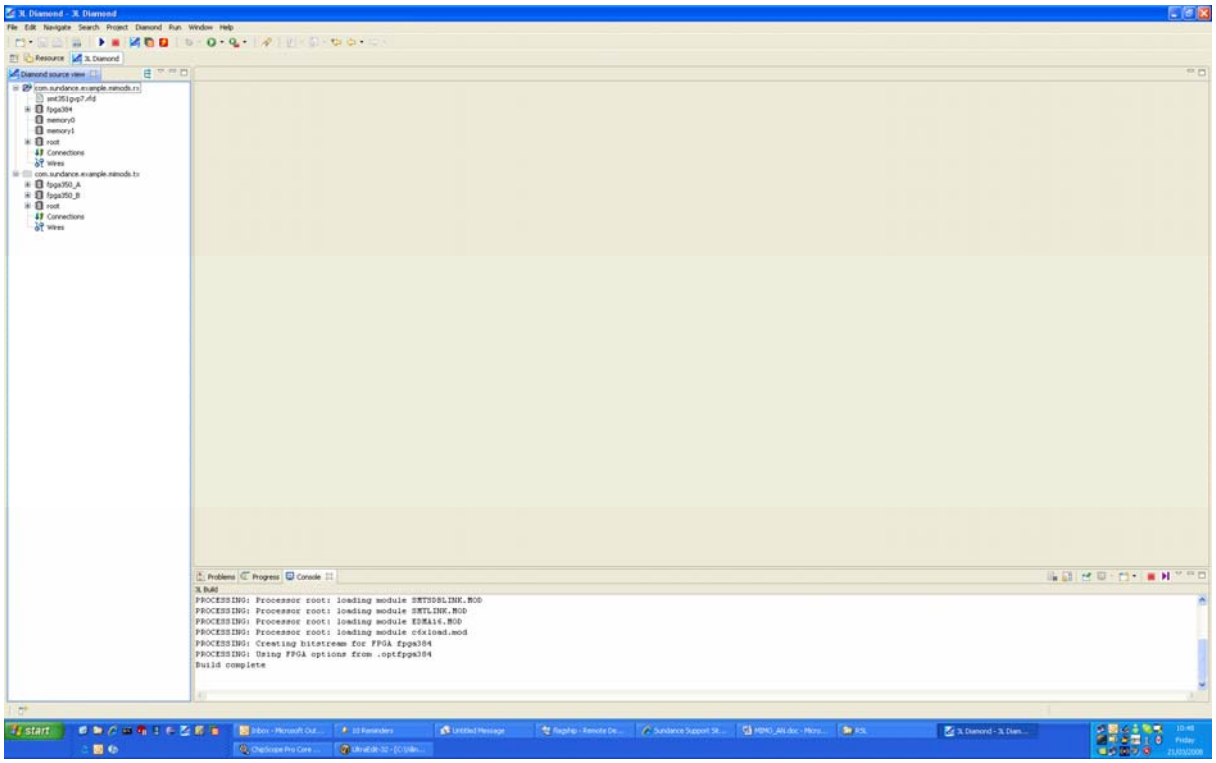


Figure 9: the MIMO-DS workspace

The receiver uses two SMT351 modules (memory0 and memory1) to store the samples. In this example we use a pre-built bitstream to configure the SMT351.

The bitstream file is "smt351gvp7.rfd". Place this file somewhere on the hard-drive of the PC and point the project to it.

The location of the bitstream is set in the ***Advanced*** properties of the SMT351 processors (memory0 and memory1) as shown on the picture.

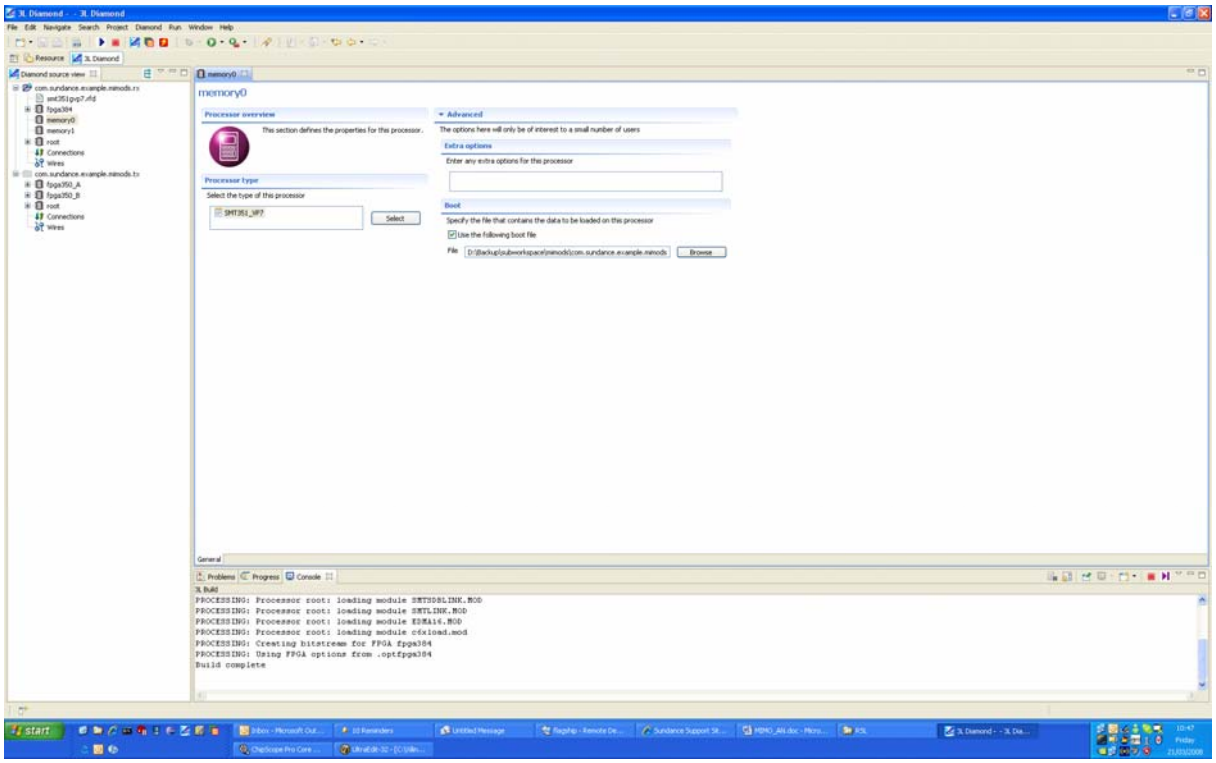


Figure 10: Setting the bitstream for the SMT351 processors

8.2 Running the example

8.2.1 Connections

The two systems can be plugged either in two different PCs or in the same PC.

If one PC only is used refer to section "**8.2.3**".

The following connections should be made to be able to run the example.

The output channels and the input channels receive the same sampling clock.

Connect an external signal generator to connector J29 of the SMT350s and the SMT384. Set the frequency at 100Mhz.

Connect the output channels to the input channels:

- SMT350#1 J32 – SMT384 J3 (ChA)
- SMT350#1 J31 – SMT384 J11 (ChB)
- SMT350#2 J32 – SMT384 J6(ChC)
- SMT350#2 J31 – SMT384 J7 (ChD)

8.2.2 Running the software

We assume here that each system is plugged in a different PC.

First, start the transmission side by running project "com.sundance.example.mimods.tx".

Once this project is running, the DACs are outputting the sine-wave programmed in the FPGA.

Then start the reception side by running project "com.sundance.example.mimods.rx".

After few seconds you should see the following windows appearing:

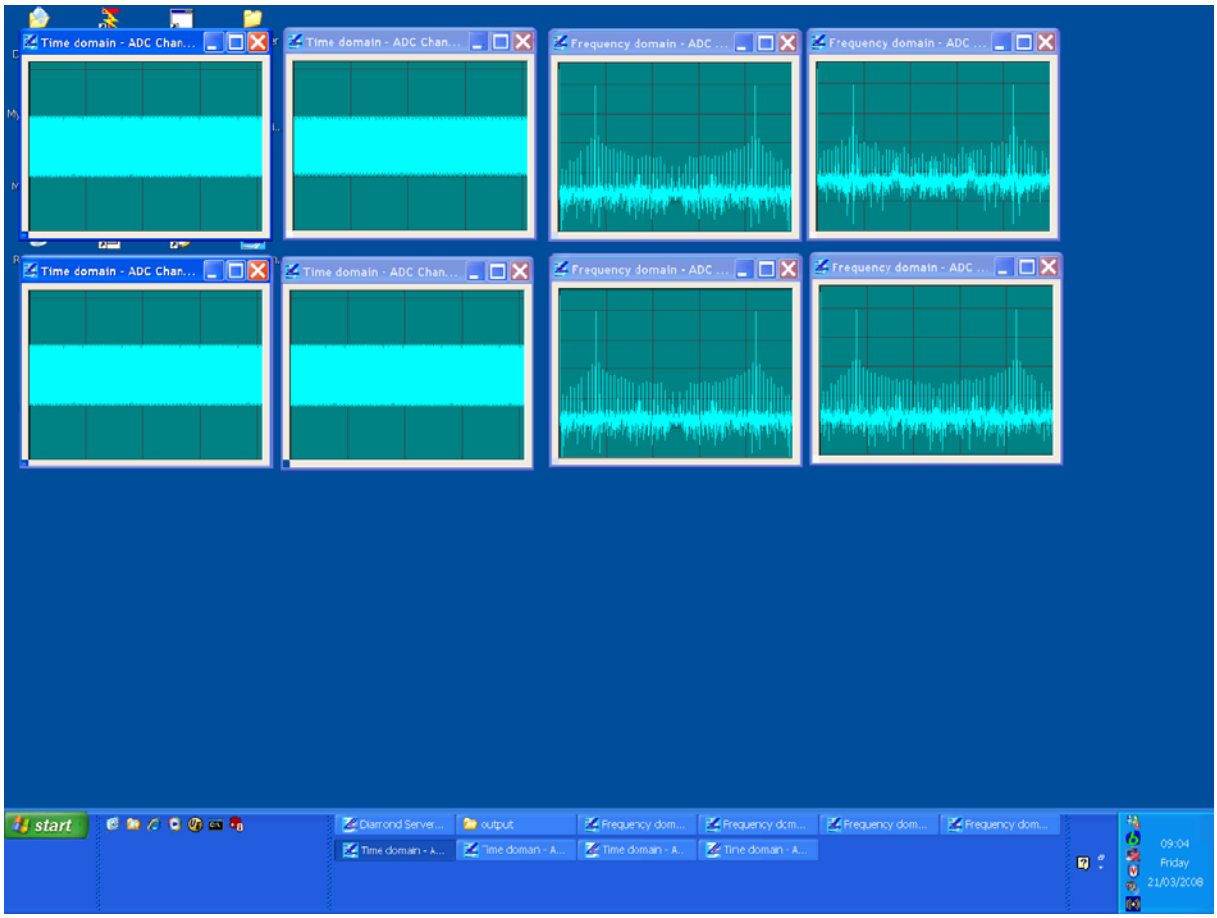


Figure 11: reception software

The four windows on the left hand side show the raw signals received from each one of the input channels. The four windows on the right hand side show the result of the FFT of the signal processed by the DSP.

In addition, the samples can be stored into files on the host PC. The example creates four files, one per channel, in directory "\workspace\com.sundance.example.mimods.rx\output\".

File "\workspace\com.sundance.example.mimods.rx\output\A" contains the data of channel A.

File "\workspace\com.sundance.example.mimods.rx\output\B" contains the data of channel B.

File “\workspace\com.sundance.example.mimods.rx\output\C” contains the data of channel C.

File “\workspace\com.sundance.example.mimods.rx\output\D” contains the data of channel D.

8.2.3 Running transmitter and receiver in the same PC

If one PC only is used make sure that the power supply of the PC is powerful enough to support both systems.

Because the two systems are plugged in the same PC, you cannot directly run the projects from the Diamond IDE. Instead you need to use the Diamond server to run the applications.

We will open two instances of the Diamond server. One of them will be used to connect to the transmitter and the other to the receiver.

To open a new instance of the Diamond server, click on file->new server instance as shown below. A new server window should appear.

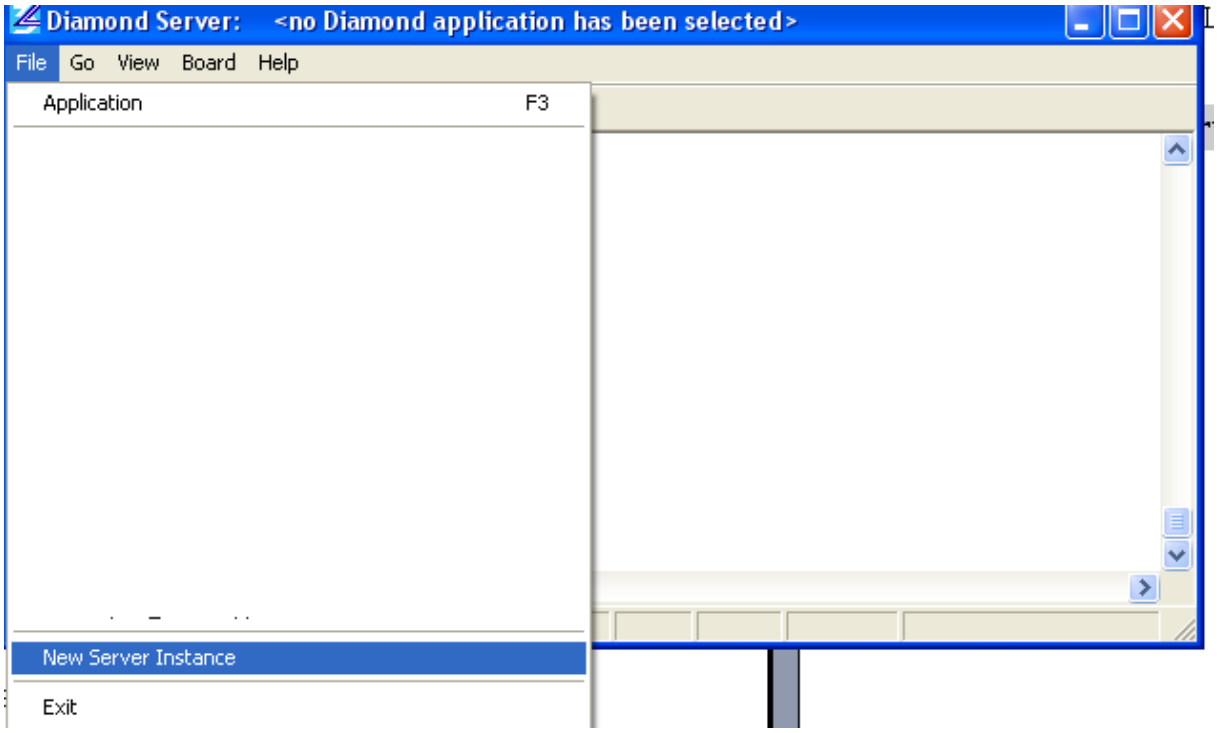


Figure 12: opening a new instance of the Diamond server

Connect each instance of the server to one of the systems plugged in the PC. To do so, click on Board->Server. The following window appears, allowing you to connect the server instance to one of the SMT310Q plugged in the PC.

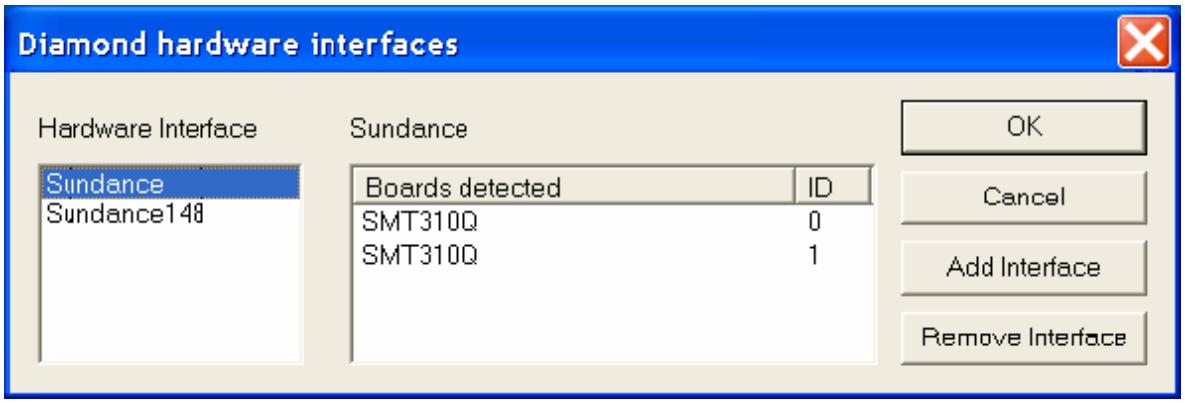


Figure 13: Connecting a server instance to a SMT310Q

Select SMT310Q ID0 for the first instance of the Diamond server.

Select SMT310Q ID1 for the second instance of the Diamond server.

Send a reset on the server instance connected to SMT310Q ID0 and look at the hardware to identify the system to which this server is connected (transmitter or receiver). This done, make sure the comport switch matrix of each serer instance is set-up properly.

You can now run each one of the two applications in the appropriate sever instance.

The applications are located at the following addresses in your Diamond IDE workspace:

Transmitter:"workspace\com.sundance.example.mimods.tx\output\ com.sundance.example.mimods.tx.app".

Receiver:"workspace\com.sundance.example.mimods.rx\output\ com.sundance.example.mimods.rx.app".

8.3 Implementation

The example is developed with 3L Diamond.

The example is split in two independent applications. There is one application for the transmission and one application for the reception.

Each application is made of a number of tasks running on the DSP and the FPGA in each system.

Source code for the example is provided.

8.3.1 Software requirement

Make sure that you have Xilinx ISE9.2sp4 as well as Diamond V3.1.10 or later installed on your PC.