HARDWARE DESIGN

FOR IMPLEMENTING

A VIRTEX-4 FX60 POWERPC SYSTEM

APPROVAL PAGE

Name	Signature	Date

AUTHOR/S

Name	Signature	Date
Francois Godreau		13/02/07

DOCUMENT HISTORY

Date	Initials	Revision	Description of change
07/03/19	F.G	1.0	New document

TABLE OF CONTENTS

1. SCOPE

This document describes an example design for an FPGA Virtex 4 FX60 only module.

1.1. INTRODUCTION

The PowerPC reference design is implemented by Sundance on Xilinx Virtex family V4-FX60 targets custom features of these chips to reach maximum performances.

The core is validated on SMT339 modules

Implementing the PowerPC design in a Sundance Module you will allow running application specific systems which contain both hardware and software.

1.2. PURPOSE

The purpose of this development is to implement a PowerPC processor in a Virtex4-Fx60 with a comport interface. Thus, this microprocessor will be able to communicate with other system like the host or another board.

1.3. APPLICABILITY

All the Sundance modules with a FPGA V4-FX60 are able to benefit from the implementation of that design.

2. APPLICABLE DOCUMENTS AND REFERENCES

2.1. APPLICABLE DOCUMENTS

2.1.1. External documents

None

2.1.2. Internal documents

Firmware specifications for FPGA-only TIMs: Specifications for D000071F-spec.pdf.

SMT6004 help

2.1.3. Project documents

N.A

2.1.4. External documents

N.A

2.1.5. Internal documents

N.A

2.1.6. Project documents

N.A

2.2. PRECEDENCE

In the event of conflict between the text of this document, and the applicable documents cited herein, the text of this document takes precedence. Nothing in this document however, supersedes applicable laws and regulations unless a specific exemption has been obtained and is identified in the text of this document.

3. ACRONYMS, ABBREVIATIONS AND DEFINITIONS

3.1. ACRONYMS AND ABBREVIATIONS

TIM	Texas Instruments Module
CP	Comport
SDB	Sundance Digital Bus
CB	Connector Block

3.2. DEFINITIONS

4. TOP LEVEL DESIGN

4.1. SUBSYSTEM BREAKDOWN

The top-level design of the hardware is defined in the “top.vhd” file. Here is the diagram showing the top level design with ISE:

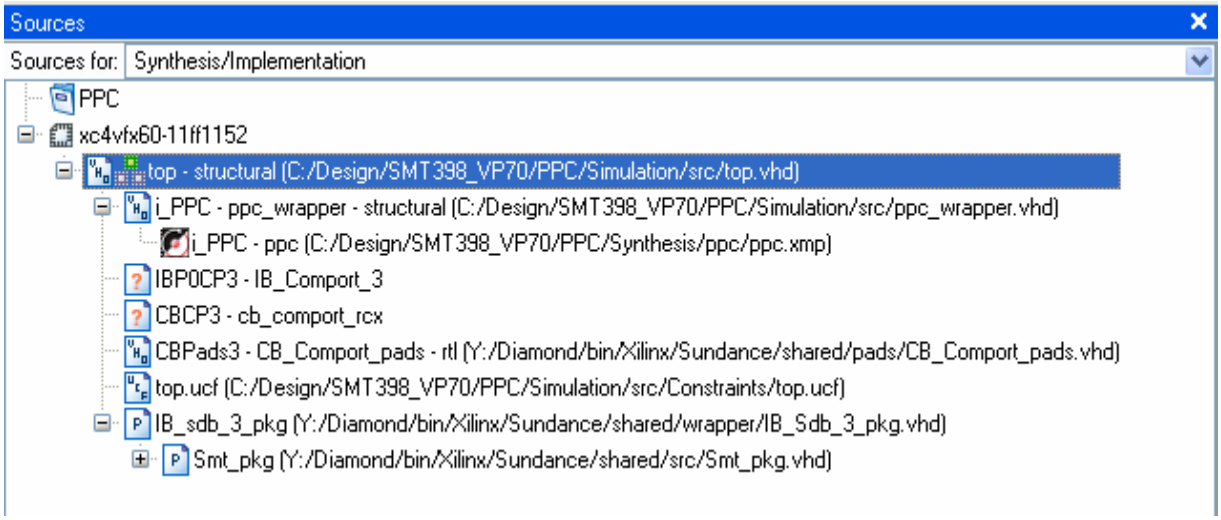

Figure 1 The top level design

The basic block diagram of this application is illustrated in Figure 2.

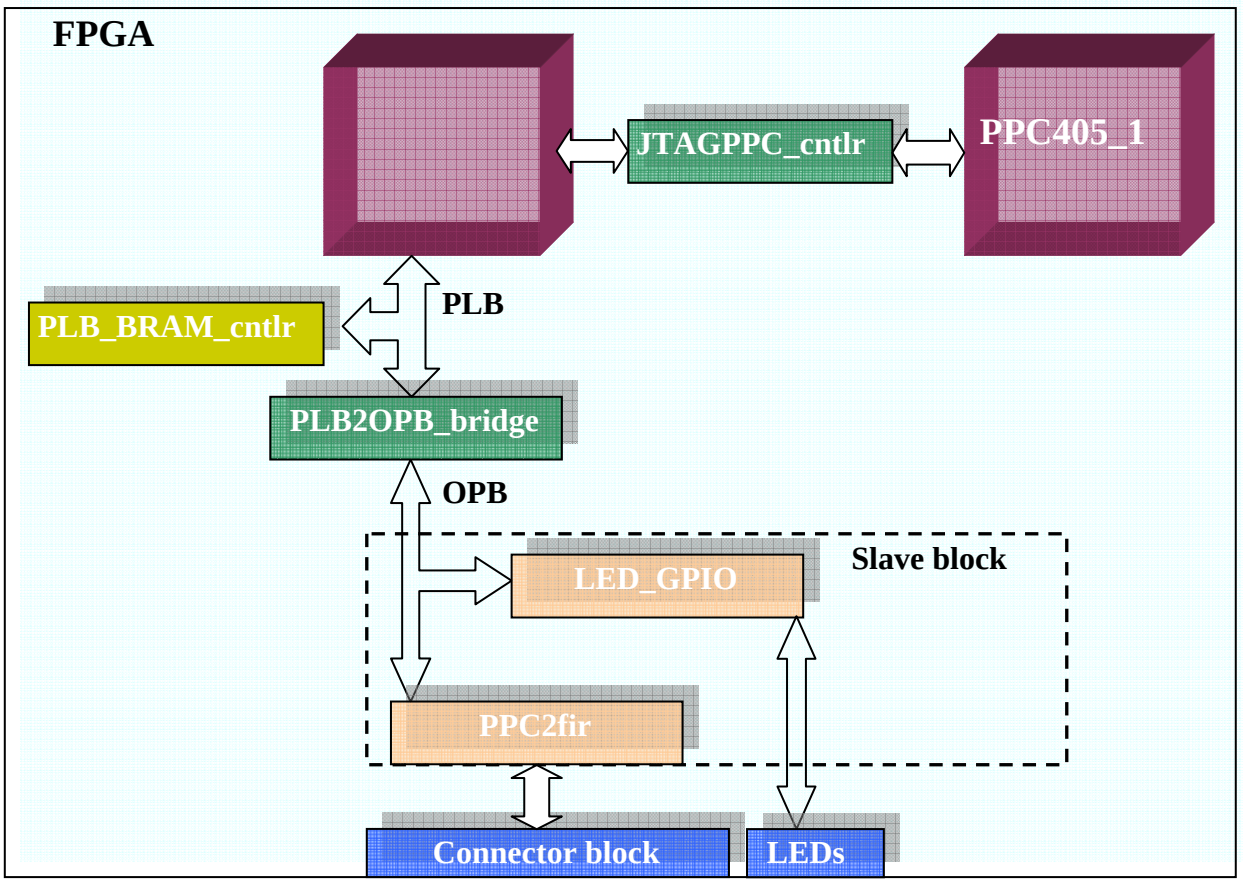

Figure 2 Block diagram (EDK)

Both PowerPC are implemented in this design, but only one is connected on the OPB bus. Two IPs are connected to the PowerPC via OPB: LEDS and ppc2fir_0. The comport interface for the PowerPC is defined in the PPC2fir block. A JTAG controller is added to the design and connected to both PowerPC. Every connection between the PowerPC and its interface is illustrated by the following picture:

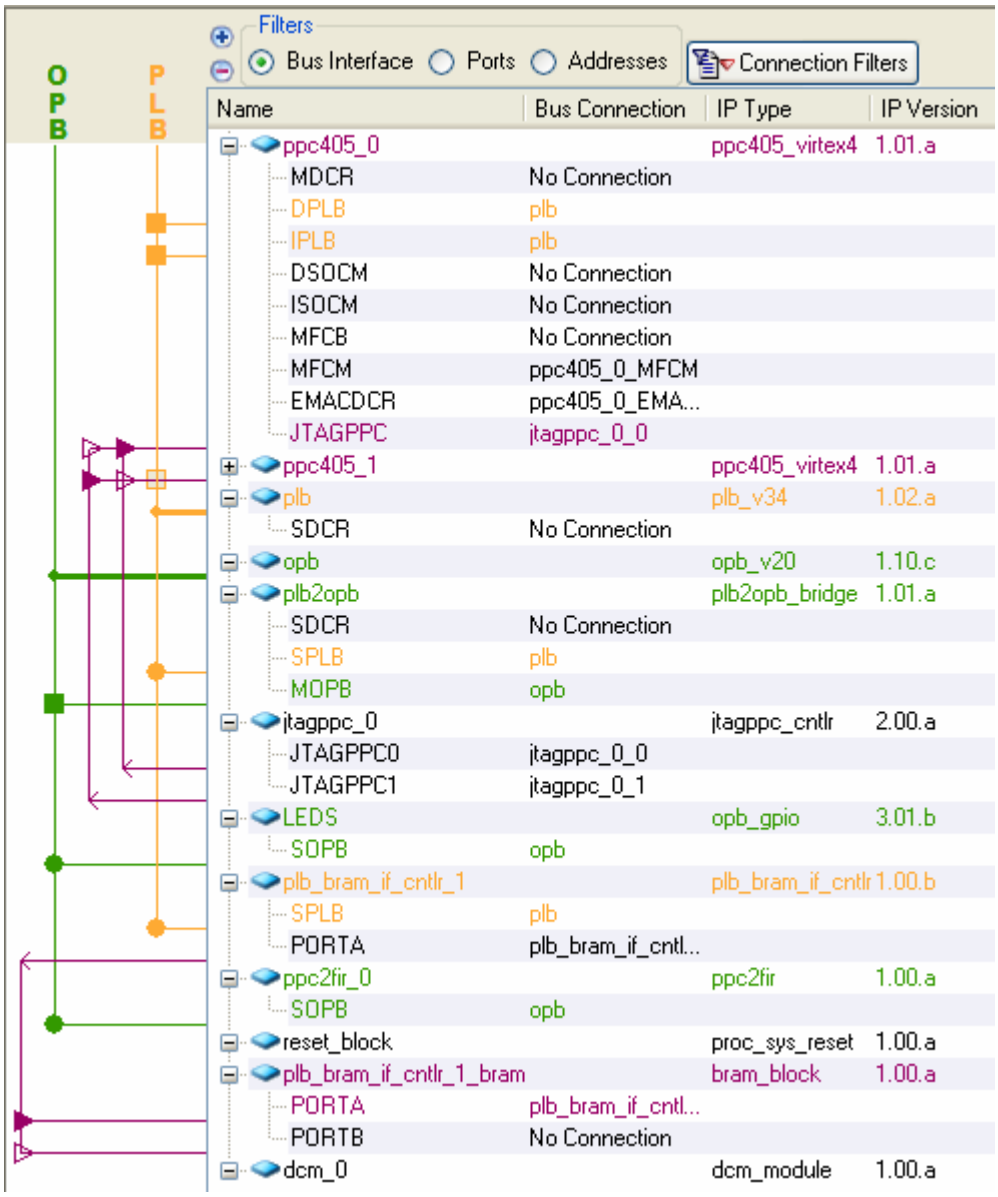

Figure 3 bus interface

4.2. SYSTEM STATES AND MODES

The reference design for this application demonstrates the communication between the PowerPc and another processor via the comport 3. Moreover, this application displays a counter with the LED on the board.

This design can be used with the C function generated automatically by EDK. For this example, we have used the “GpioOutputExample” function to flash the LED on the board and the “XIo_In32(Address)” function to transfer the data from the comport. The comport interface has been validated with the debugger (see paragraph 5.1.5). So, the following figure explains the source file loaded on the PowerPC.

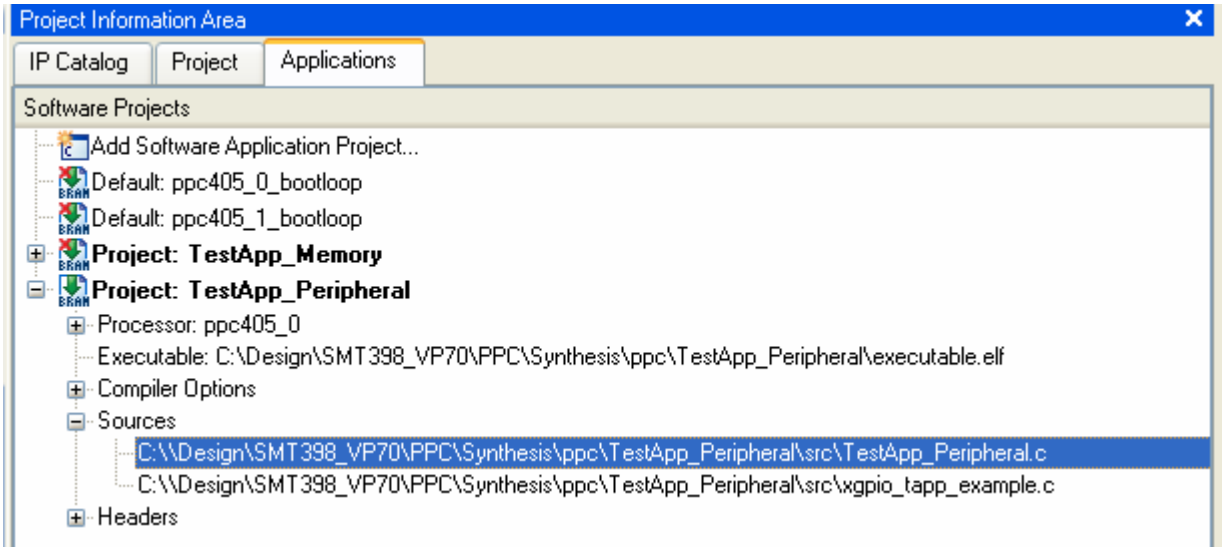

Figure 4 Source file

Figure 5 shows a simplified system level block diagram in which the PowerPC 0 transfers data to another processor.

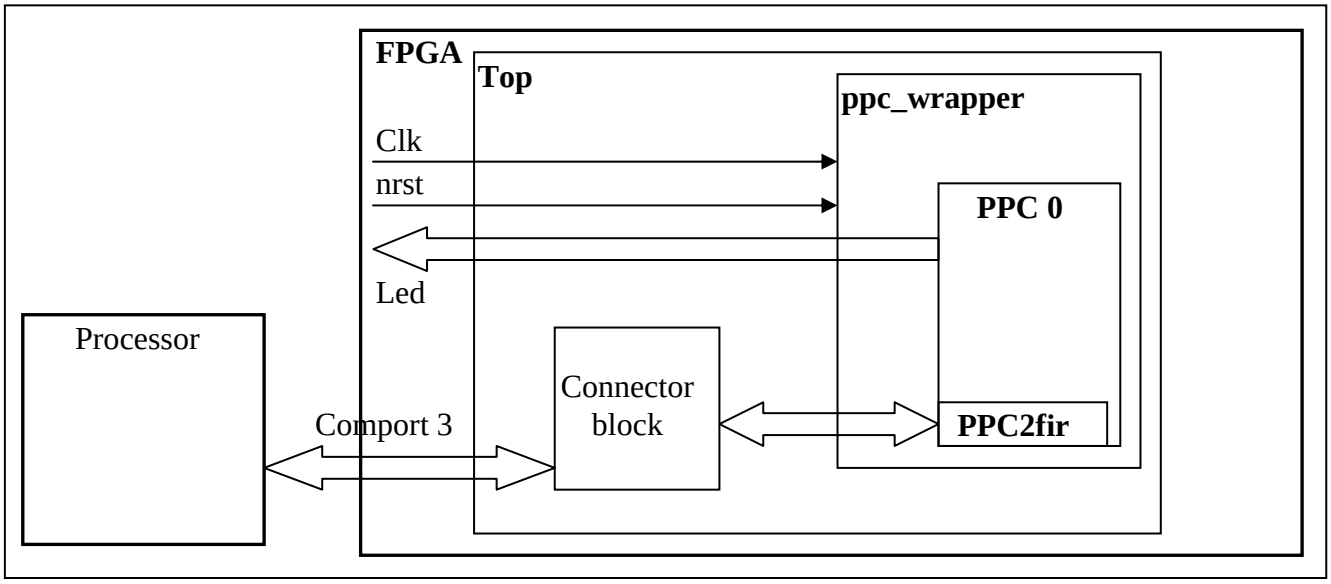

Figure 5 hardware architecture

4.3. CLOCK MANAGEMENT

This reference design uses one dedicated input clock (the on-board oscillator). This input clock feeds one of the enhanced DCM in the Virtex 4. This DCM boots the input clock and adjusts the phase of the clock of the PowerPC. It is running at 100MHz and its name is “sys_clk_s”. This DCM is generated and configured automatically by EDK.

5. DETAILED DESIGN

5.1. MODULE 1

Just a DSP is required for this module. For the test, we have used the SMT319 board. The DSP on this board will allow communicating with the PowerPC via a comport.

5.1.1. Module's interface

For this design reference, you have to connect physically the comport 1 of the module 1 (smt319) to the comport 3 of the module 2 (smt339).

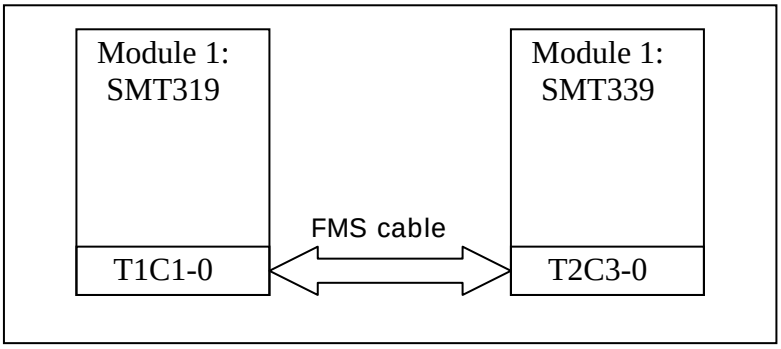

Figure 6 Comport connection

5.1.2. Functions

We use this module to transmit and check the data receive by the module 2. In fact, we can use Code Composer Studio to transfer a data to the PowerPC via the comport. Refer you to the memory map. (For the SMT319, the base address of the comport 1 is 0x90008000).

5.1.3. Implementation

You have to program the DSP and the FPGA with the latest versions of bootloader and of bistream.

5.2. MODULE 2

A FPGA Virtex 4 FX60 is required for this module. For the test, we have used the SMT339 board. This reference design will load in this board.

5.2.1. Module's interface

For this module, the comport 3 interface is used thus the LEDs on the board.

Caution: the pin allocation for the comport 3 and the LEDs depends of the Sundance board used.

Reminder: for this design reference, you should connect the comport 1 of the module 1 (smt319) to the comport 3 of the module 2 (smt339).

5.2.2. Functions

The function of this module is flashed the LEDs on the board with the PowerPC and transferred data to and from another processor, in our case the DSP on the SMT319

5.2.3. Timing specification

All the timing specifications are defined in the both files: top.ucf and ppc.ucf.

5.2.4. Implementation

In first time, you have to load the bistream with ISE. For that, click on configure Device.

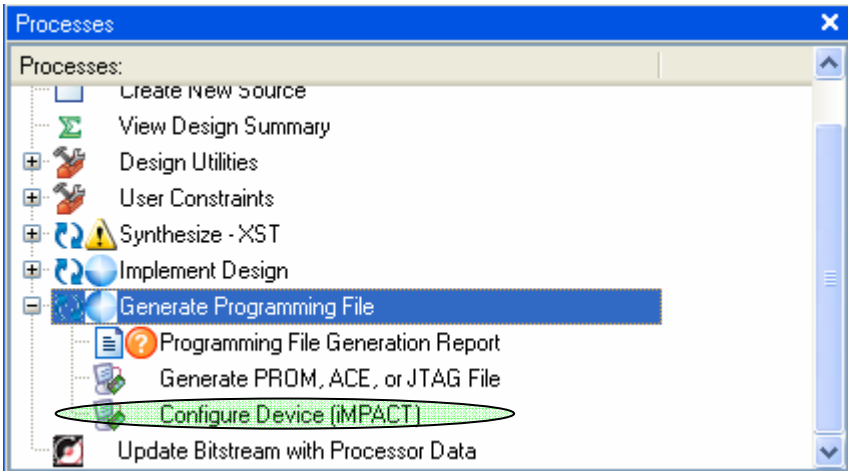

Figure 7 Loading the bistream with ISE

After this state, you can open the debugger with Xilinx Plateform Studio. Firstly, start XMD to configure the debugger and open the software debugger. Load the program and run it.

Figure 8 Debugger XPS

5.2.5. Validation

Now, you can check the design in using CCS and the debugger. You can transfer a data from module 1 to the module 2 and check it.

Note: the address of the comport 3 is 0x7001C000 for this reference design. You can use EDK to verify it.

Filters

Bus Interface | Ports | Addresses | Generate Addresses

Instance	Name	Address	Base Address	High Address	Size	Lock	ICache	DCache	Bus Connection	IP Type	IP Version
opb					U						
ppc405_0	MDCR	IDCR			U				No Connection		
ppc405_1	MDCR	IDCR			U				No Connection		
plb	SDCR				U				No Connection		
plb2opb	SDCR	DCR			U				No Connection		
LEDS	SOPB		0x00000000	0x000003FF	1K				opb		
ppc2fir_0	SOPB		0x00000400	0x00000400	1				opb		
ppc2fir_0	SOPB	AR0	0x70000000	0x7007FFFF	512K				opb		
plb_bram_if_cntlr_1	SPLB	c_baseaddr:c_highaddr	0xffffc000	0xffffffff	16K				plb		
plb2opb	SPLB	RNG0	0x00000000	0x7FFFFFFF	2G				plb		
plb2opb	SPLB	RNG1			U				plb		
plb2opb	SPLB	RNG2			U				plb		
plb2opb	SPLB	RNG3			U				plb		

Figure 9 Interfaces address (XPS)

Figure 10 shows an example to send a data from the module 1 with CCS to the module 2. We have sent the data 1 on the comport 1 (address 0x90008000). So we can see this value on the comport 3 at the address 0x70018000.

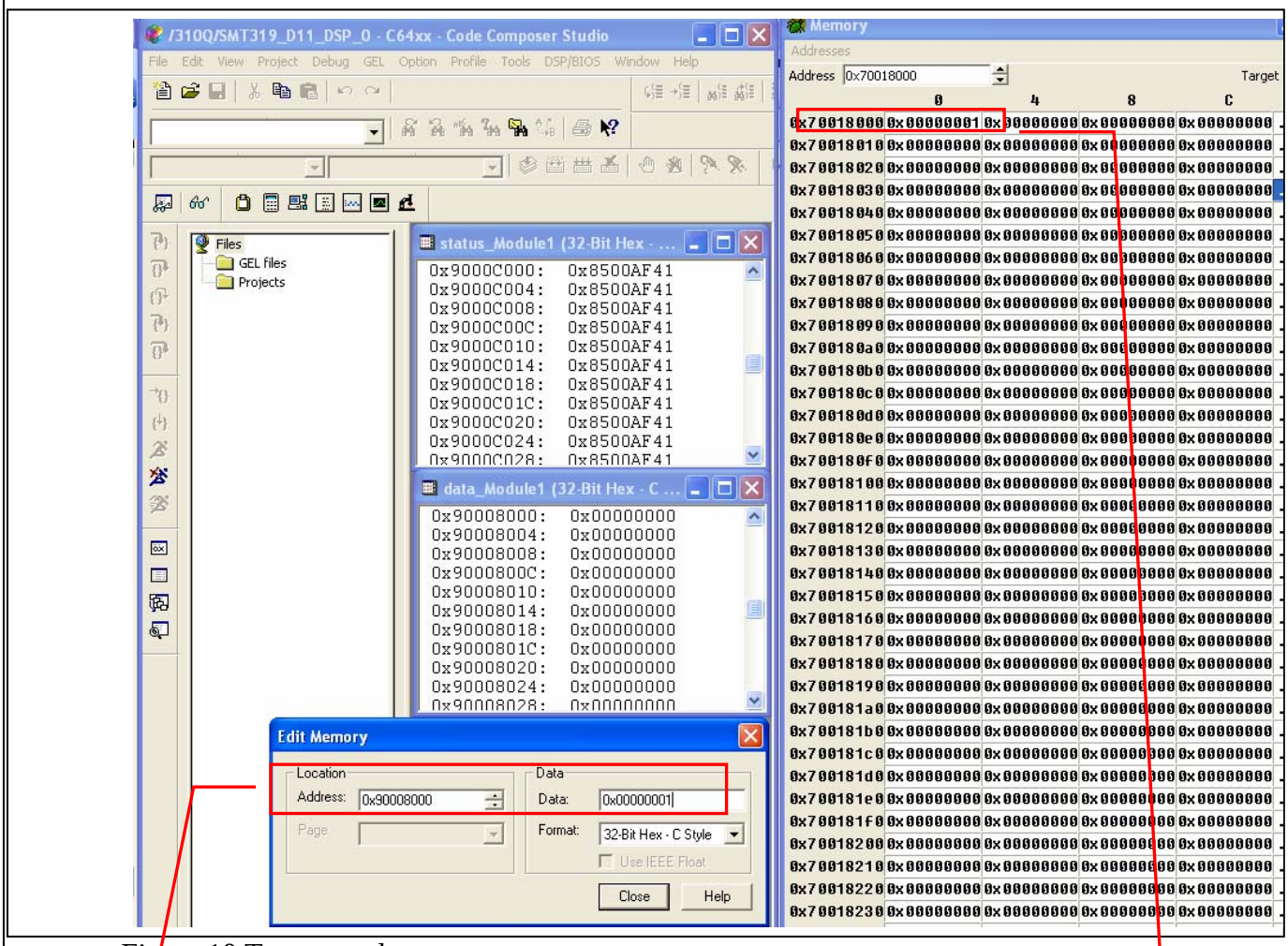

Figure 10 Test procedures

Data sent from the module 1 to the module 2

Data received by the module 2

Here is another example which uses the polling example of the SMT6400. You have to send a data from the module 1 to the module 2 with CCS, like the previous example. The value read from the comport 3 can be checking with the variable “read” in the local variables widow.

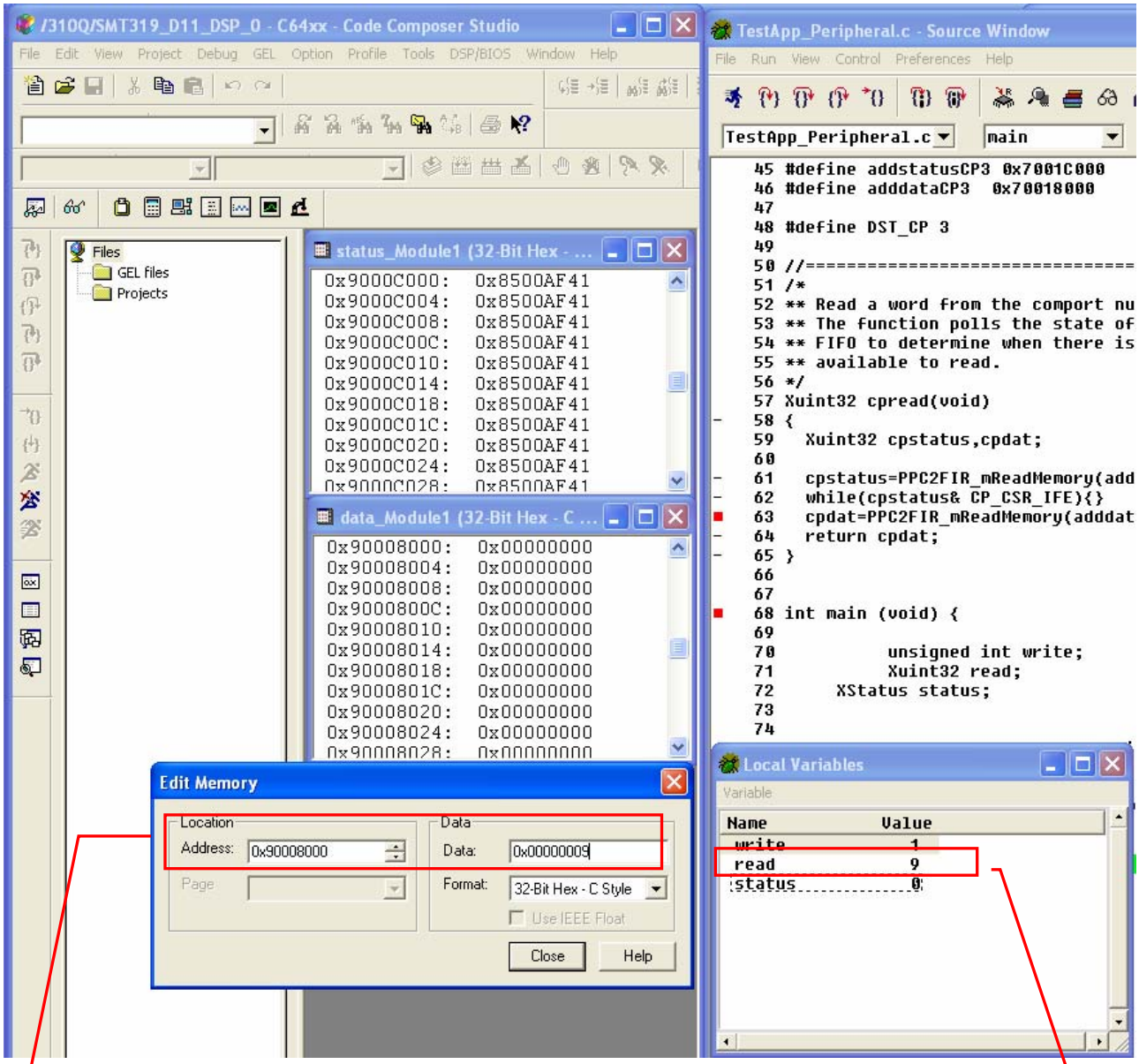

Figure 11 Test procedures with the polling example

6. DESIGN INTALLATION

6.1. DIRECTORT STRUCTURE

The reference design is provided as VDHL compressed file. To install the files, extract the contents of the PPCV4.zip file.

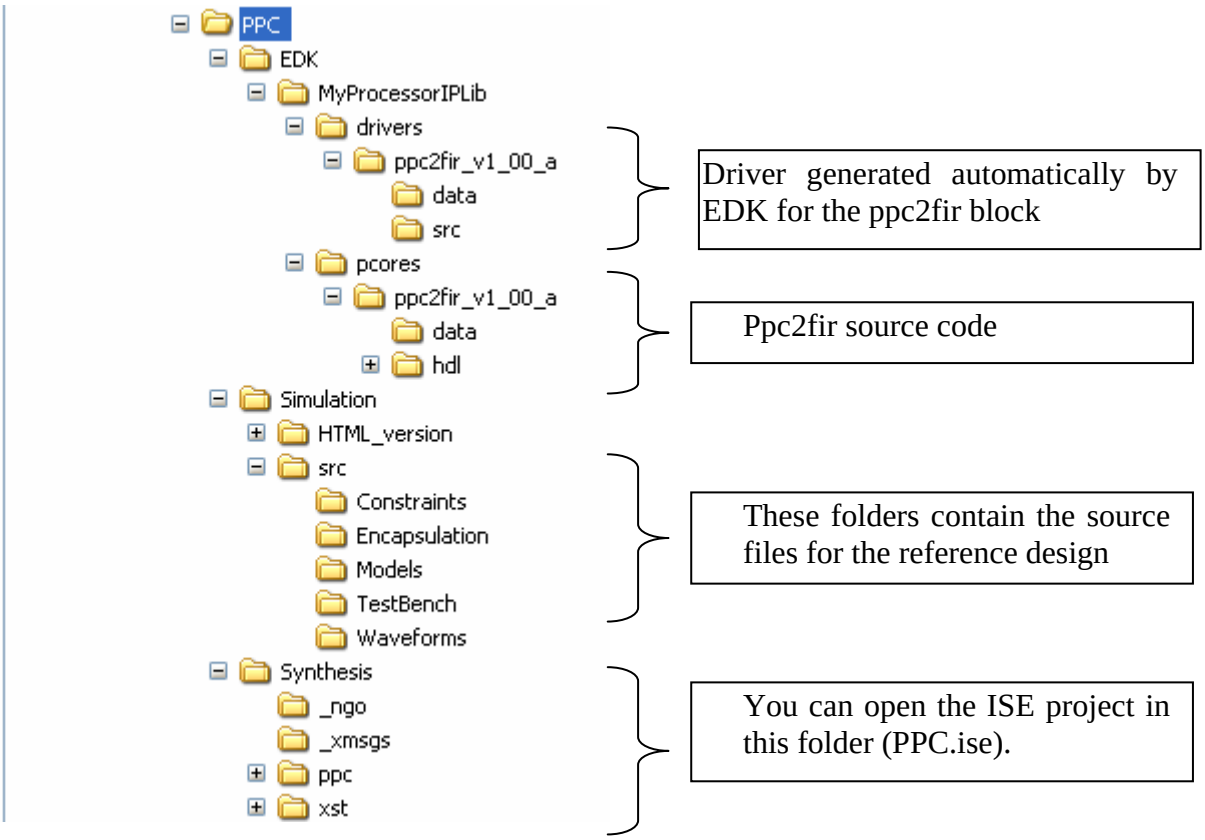

Figure 12 Shows the directory structure created

6.2. VERSION SOFTWARE