Agilent E9524A Inverse Assembler for Xilinx MicroBlaze

Design Guide

Agilent Technologies

Notices

Manual Part Number

E9524-97000

This manual is available online at www.agilent.com. It is not an orderable part.

Revision History

E9524-97000, July 2005

Agilent Technologies, Inc.
1900 Garden of the Gods Road
Colorado Springs, CO 80907 USA

Trademark Acknowledgements

Windows and MS Windows are U.S. registered trademarks of Microsoft Corporation.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

1 Introduction

In This Guide...

This *Design Guide* provides information to assist you in designing a board which will be compatible with the Agilent E9524A Inverse Assembler for Xilinx MicroBlaze. It tells you what signals are required by the decoder, and suggests how to route these signals to a header.

For information on using the decoder, see the online help which is installed with the decoder.

Product Overview

The inverse assembler, used with an Agilent Technologies logic analyzer, allows you to reconstruct program flow by capturing the instruction address of every executed instruction, then looking up the associated opcode in the object file, and then decoding the opcode into a MicroBlaze mnemonic. The inverse assembler traces code flow only; it does not capture operand values or memory values.

The inverse assembler uses a small number of external FPGA signals, plus a clock signal. The number of signals depends on the size of the address space in which the program is executing.

Target System Requirements

The inverse assembler has been designed to work with target systems meeting the following requirements:

Supported processor

- Xilinx MicroBlaze processor core.

Object files

- You must have access to the object files for the code which is executing on your target system.
- You must know the memory address where the object code is loaded.

Supported compilers

- The gcc compiler which is shipped with Platform Studio 7.1.01i, or equivalent.

Headers

- You must provide a way to connect logic analyzer probes to the signals on your target system:
  - You must route the required signals from the MicroBlaze core to external pins on the FPGA.
  - You must provide a way, such as a MICTOR header, to connect the logic analyzer probes to the signals.

Equipment Required

Logic analysis system

You need an Agilent 16900-series or 1680/90-series logic analyzer.

Logic analyzer cards

The logic analyzer card(s) you use must support the speed of the bus you are probing.

The logic analyzer card(s) must provide enough channels to probe the headers on your target system.

Table 1 Logic analyzer channels required

Probing scheme	Number of channels
up to 32 PC_EX signals 1 MICTOR connector	32

Probes

You need an appropriate number of logic analyzer probes (“adapter cables”) to connect the logic analyzer cables to the header on your target system. The probe must match the type of connector you have placed on your board. Agilent recommends a MICTOR connector and an Agilent E5346A or E5380A probe.

1 Introduction

2
Designing Your Board

This chapter describes the factors you need to consider when designing and preparing your target system for logic analysis.

Overview of the Connectors

You must provide either one AMP MICTOR 38 connector with the signal mappings shown in the following sections, or another connector which maps the signals to the same logic analyzer channels.

Designing the Headers

AMP MICTOR 38 connectors

The signal-to-connector mappings shown in this chapter assume you are using AMP MICTOR 38 connectors.

Each MICTOR 38 connector carries 32 signals plus two clocks (CLK1 for two logic analyzer pods). Probes (part number E5346A, sometimes called “high-density termination cables”) are required to connect the logic analyzer cables to the MICTOR connector. These probes contain the required termination. One probe is required for every two logic analyzer pods.

To increase the structural support for the probes, you should use support shrouds on each connector.

For more information, including mechanical dimensions, see the *Agilent Technologies E5346A 38-Pin Probe and E5351A 38-Pin Adapter Cable Installation Note*, available from www.agilent.com.

Design Considerations

The connector must be close enough to the signal source so that the stub length created is less than $^{1}/_{5}$ the t_r (bus risetime). For PC board material, (er = 4.9) and Z_o in the range of 50 - 80Ω, use a propagation delay of 160 ps/inch of stub.

Each probed signal line must be able to supply a minimum of 600 mV to the probe tip and handle a minimum of 90 kΩ shunted by 10 pF. The maximum input voltage to the logic analyzer is ±40V peak

Other connectors

You may use other connector/probe combinations, including:

- Agilent soft touch connectorless probe
- Agilent pro series soft touch connectorless probe
- Samtec connector with Agilent Samtec probe

Design information for these connectors can be found on the web at www.agilent.com. If you use one of these connectors, route each signal to the same logic analyzer signal as shown for the MICTOR connector. For example, **PC_EX 15** should always be connected to **D0 even** on the logic analyzer, regardless of which connector/probe combination is used.

Signal-To-Connector Mappings

MICTOR header

Analyzer Pod	MicroBlaze Trace Signal	Mictor pin # (top view)		MicroBlaze Trace Signal	Analyzer Pod
5V	NC	1	2	NC	I2C
5V	NC	3	4	NC	I2C
CLK even	VALID_INSTR	5	6	CLOCK	CLK odd
D15 even	PC_EX 0 (MSB)	7	8	PC_EX 16	D15 odd
D14 even	PC_EX 1	9	10	PC_EX 17	D14 odd
D13 even	PC_EX 2	11	12	PC_EX 18	D13 odd
D12 even	PC_EX 3	13	14	PC_EX 19	D12 odd
D11 even	PC_EX 4	15	16	PC_EX 20	D11 odd
D10 even	PC_EX 5	17	18	PC_EX 21	D10 odd
D9 even	PC_EX 6	19	20	PC_EX 22	D9 odd
D8 even	PC_EX 7	21	22	PC_EX 23	D8 odd
D7 even	PC_EX 8	23	24	PC_EX 24	D7 odd
D6 even	PC_EX 9	25	26	PC_EX 25	D6 odd
D5 even	PC_EX 10	27	28	PC_EX 26	D5 odd
D4 even	PC_EX 11	29	30	PC_EX 27	D4 odd
D3 even	PC_EX 12	31	32	PC_EX 28	D3 odd
D2 even	PC_EX 13	33	34	PC_EX 29	D2 odd
D1 even	PC_EX 14	35	36	PC_EX 30 or GND	D1 odd
D0 even	PC_EX 15	37	38	PC_EX 31 or GND	D0 odd

Bus and signal descriptions

PC_EX Required. The MicroBlaze program counter, from the MicroBlaze trace signals PC_EX.

Only the active bits of PC_EX need to be routed to the mictor connector. In general, the most significant bits and the least significant bits of PC_EX do not need to be routed to the connector because they are static. This reduces the number of FPGA pins that are dedicated to MicroBlaze trace.

Most significant bits The upper n bits of PC_EX do not need to routed to the connector because they are static.

For example, suppose your system has 1M byte of program memory. 1M byte can be addressed with 20 bits. The upper 12 bits of the 32-bit program counter are static and are not needed by the inverse assembler. They do not have to be routed to the connector.

You must route a contiguous set of PC-EX to the connector. For example, if you route PC_EX12 and PC_EX15, then you must also route PC_EX13 and PC_Ex14, even if fPC_EX13 and PC_Ex14 are static.

Least significant bits The least significant two bits of PC_EX are always zero. This is because the MicroBlaze instructions always start on 4-byte boundaries. The least significant two bits of PC_EX are represented by MICTOR pins 36 and 38. Agilent recommends that these pins be grounded. There are two ways to do this:

- Ground the pins by connecting them to MicroBlaze Trace Signal PC_EX 30 and PC_EX 31 on the FPGA, or by connecting them to any other FPGA pins that are ground. This is the most straightforward method and provides maximum flexibility by allowing the connector pins to be used to probe any FPGA signal when they are not being used for MicroBlaze debug. This method has the disadvantage of using two pins of the FPGA to provide two bits of zero when the connector is used for MicroBlaze debug.

- Ground the pins by connecting them to a board ground. This has the benefit of not using two pins of the FPGA, but it has the disadvantage of eliminating pins 36 and 38 of the MICTOR connector from any other use.

It is possible, but not recommended, to use MICTOR pins 36 and 38 for signals totally unrelated to the MicroBlaze core. In this case, set an option in the inverse assembler is to indicate that the least significant two bits of PC_EX have not been provided to the logic analyzer. This has no adverse impact on the inverse assembler, but it does make logic analyzer triggers more complicated. To set a trigger on a program counter address, you will need to shift the address right by two bits. For example, to trigger on an instruction at 001C 1344, the logic analyzer trigger would be set to 007 04D1.

VALID_INSTR Required. True when PC_EX is valid. One bit wide. This is the MicroBlaze VALID_INSTR signal.

CLOCK Required. The logic analyzer also uses the MicroBlaze CLOCK signal. A logic analyzer state is captured for each clock cycle; the clock is not displayed as a bus or signal.

NC Pins 1, 2, 3, and 4 must be true no-connects. Other NC signals can be left floating (no connects), or used to measure other signals of interest.

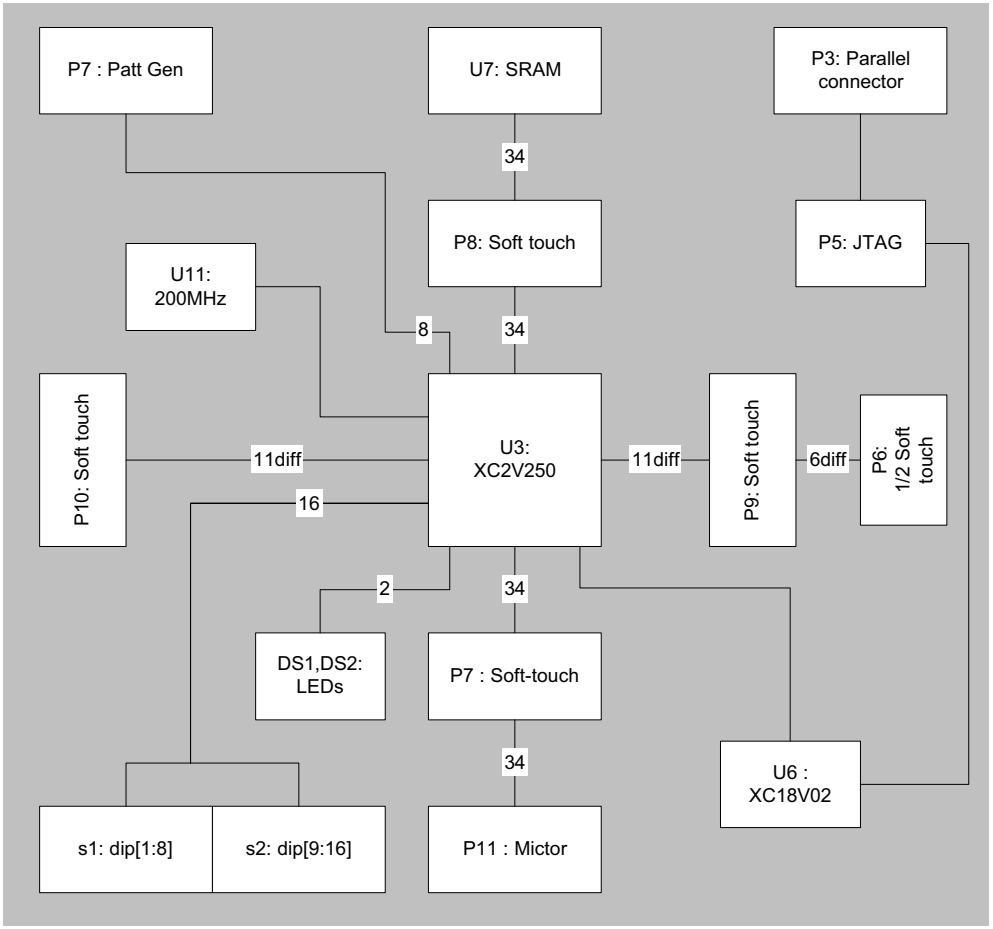
3
Example: Using XPS to Set Up an FPGA for the Inverse Assembler

This chapter gives an example of how to set up an FPGA and get the signals required by the inverse assembler to appear on the FPGA pins, using the Xilinx Platform Studio (XPS) 7.1i.

NOTE This is only an example. The steps you need to follow for your system will be different.

Overview of the Example System

The sample design is an Agilent Technologies demo board.



At the center of the board is a Xilinx XC2V250 device. Connected to this device are logic analyzer connectors, P6, P7, P8, P9, P10 and P11. Also connected to the FPGA is an 8 bit pattern generator connector P4. For memory operations component U7 is used. This component is a Cypress 16K static ram is with an address to data valid time (taa) of 10 nS.

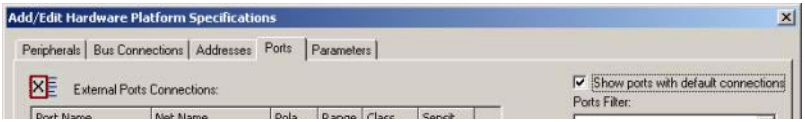
Routing the Required Signals to FPGA Pins

Route the signals to external pins by adding external ports and modifying the UCF file:

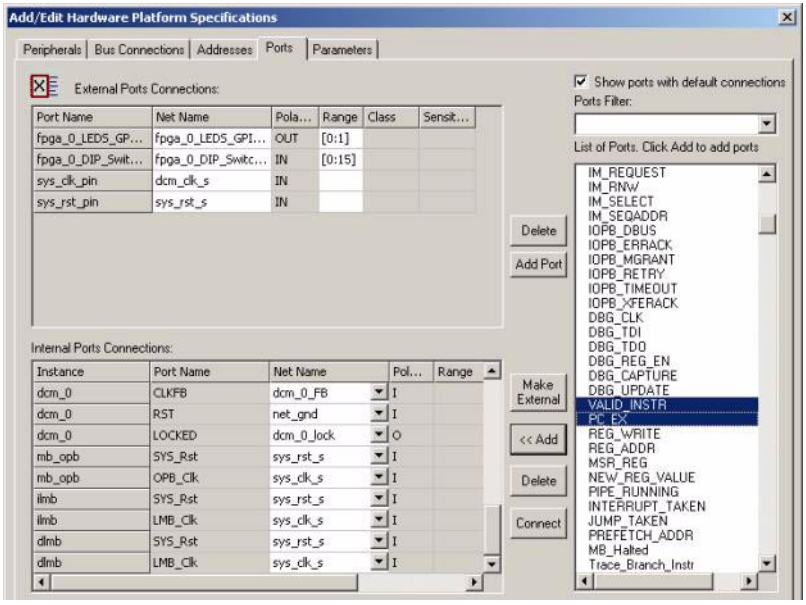
1 In XPS, select **Project->Add/Edit Cores**.

2 Select the **Ports** tab.

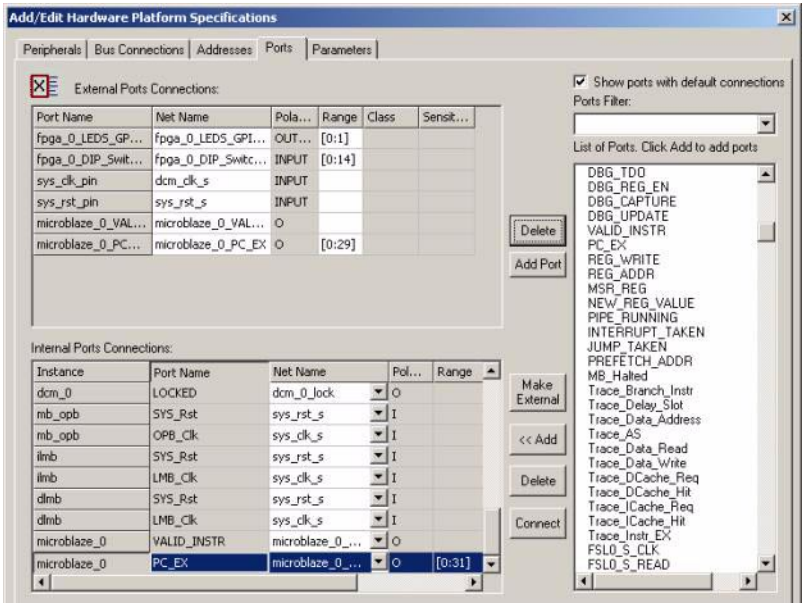
3 Check the **Show ports with default connections** box. This makes all MicroBlaze signals visible.



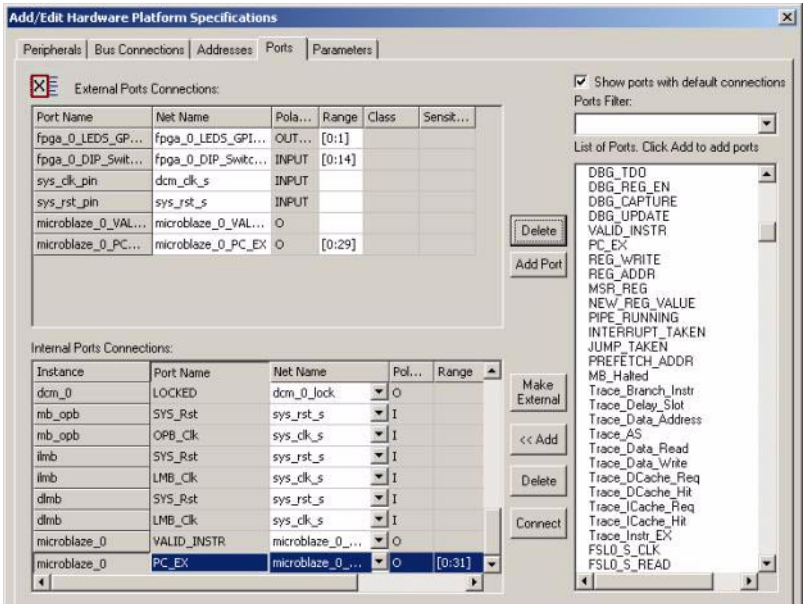
4 Select PC_EX and VALID_INSTR and click **<<Add**.



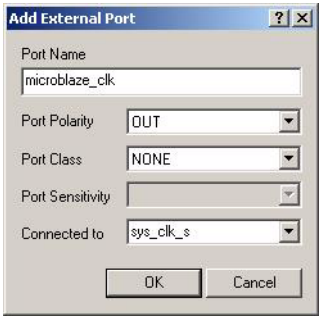
5 Select “VALID_INSTR” in internal port connections and click **Make External**.



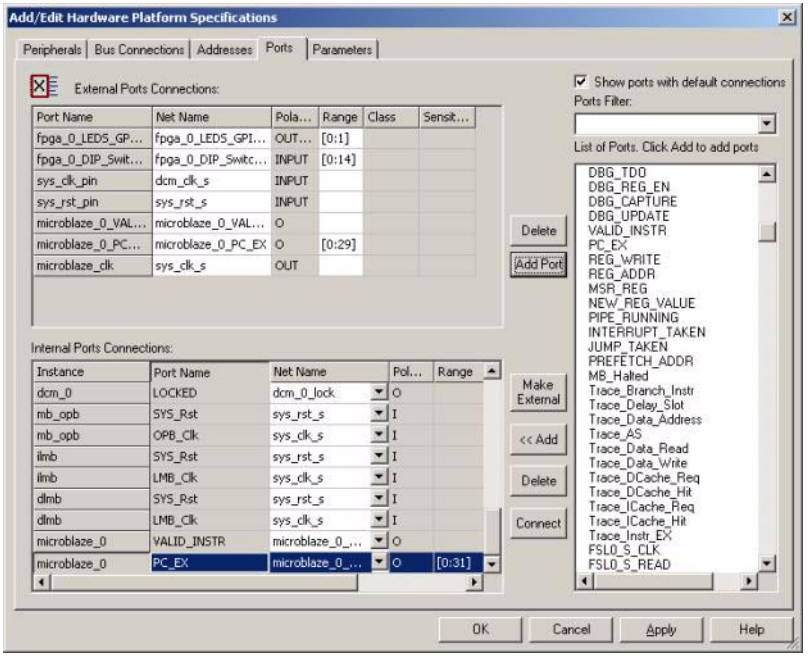
6 Do the same for PC_EX. For PC_EX set the range to be [0:29].



7 Add the MicroBlaze system clock (CLK) to an output. For this case the clock signal is called “sys_clk_s”. This is done by clicking **Add Port**. Fill in as shown below and click **OK**.



This is how your port setup should look:



8 Modify your UCF file. Add pin locations for the MicroBlaze trace outputs as shown in the example below. Note that the actual pin locations are user-defined.

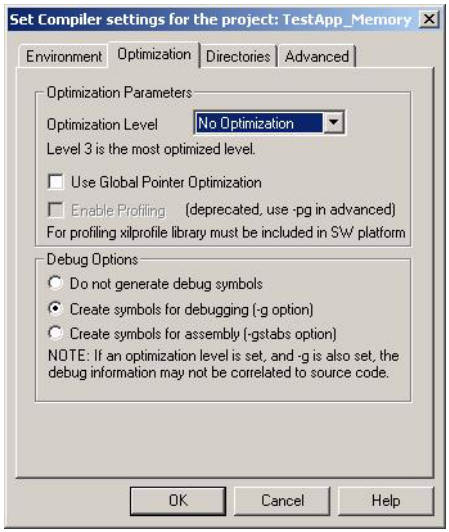
```
# IA FPGA pins
Net microblaze_clk LOC=L16;
Net microblaze_0_VALID_INSTR LOC=L15;
```

```
Net microblaze_0_PC_EX<29> LOC=C12;
Net microblaze_0_PC_EX<28> LOC=B13;
Net microblaze_0_PC_EX<27> LOC=D16;
Net microblaze_0_PC_EX<26> LOC=D15;
Net microblaze_0_PC_EX<25> LOC=D14;
Net microblaze_0_PC_EX<24> LOC=E16;
Net microblaze_0_PC_EX<23> LOC=E15;
Net microblaze_0_PC_EX<22> LOC=E14;
Net microblaze_0_PC_EX<21> LOC=F16;
Net microblaze_0_PC_EX<20> LOC=F15;
Net microblaze_0_PC_EX<19> LOC=G16;
Net microblaze_0_PC_EX<18> LOC=G15;
Net microblaze_0_PC_EX<17> LOC=K16;
Net microblaze_0_PC_EX<16> LOC=K15;
Net microblaze_0_PC_EX<15> LOC=B4;
Net microblaze_0_PC_EX<14> LOC=A5;
Net microblaze_0_PC_EX<13> LOC=B5;
Net microblaze_0_PC_EX<12> LOC=C5;
Net microblaze_0_PC_EX<11> LOC=A6;
Net microblaze_0_PC_EX<10> LOC=B6;
Net microblaze_0_PC_EX<9> LOC=D6;
Net microblaze_0_PC_EX<8> LOC=A7;
Net microblaze_0_PC_EX<7> LOC=B7;
Net microblaze_0_PC_EX<6> LOC=C7;
Net microblaze_0_PC_EX<5> LOC=A10;
Net microblaze_0_PC_EX<4> LOC=B10;
Net microblaze_0_PC_EX<3> LOC=D10;
Net microblaze_0_PC_EX<2> LOC=A11;
Net microblaze_0_PC_EX<1> LOC=B11;
Net microblaze_0_PC_EX<0> LOC=C11;
```

Building the Software

Once you have added the ports and modified the UCF file, build your software and download the FPGA bit file.

1. In XPS, select **Tools>Generate Bitstream**.
2. Select **Tools->Generate Libraries and Bsps**.
3. Set the compiler options.



4. When you are done creating the application and setting the options, right-click the project name in the tree view and select **Build Project** to create the executable. Alternatively, select **Tools> Compile Program Sources** in XPS to build all of the applications.
5. After building the applications, XPS allows you to select the application that must be initialized in the generated bitstream. To initialize the hello_world_app executable in the bitstream, right-click on the project name hello_world_app in the tree view and select **Mark to Initialize BRAMs**. From the XPS main window, select **Tools>Update Bitstream** to initialize the BRAMs with the application's executable information.
6. Download fpga bit file "implementation/download.bit"

Index

Index