

ECEN 449: Microprocessor System Design
Department of Electrical and Computer Engineering
Texas A&M University

Prof. Peng Li
TA: Andrew Targhetta
(Lab exercise created by A Targhetta and P Gratz)

Laboratory Exercise #3

Creating a Custom Hardware IP and Interfacing it with Software

Objective

The purpose of lab this week is to familiarize you with the process of creating and importing a custom IP module for a MicroBlaze based system. We will be using the 'Create and Import Peripheral Wizard' in XPS in conjunction with ISE to develop a custom peripheral for performing integer multiplication. We will then integrate the integer multiplication peripheral into a microprocessor system and develop software to interact with the peripheral using XPS. This lab serves as a simple hardware/software co-design example.

System Overview

The microprocessor system you will build in this lab is depicted in Figure 1. In place of the GPIO modules, utilized in the last lab, is a multiplication block, which represents the integer multiplication peripheral you will create. The UART in Figure 1 will be used to connect the XUP board to the workstation computer which will display the output of software executing on the MicroBlaze processor. The software you will develop in this lab will provide proof of operation for your multiplication peripheral.

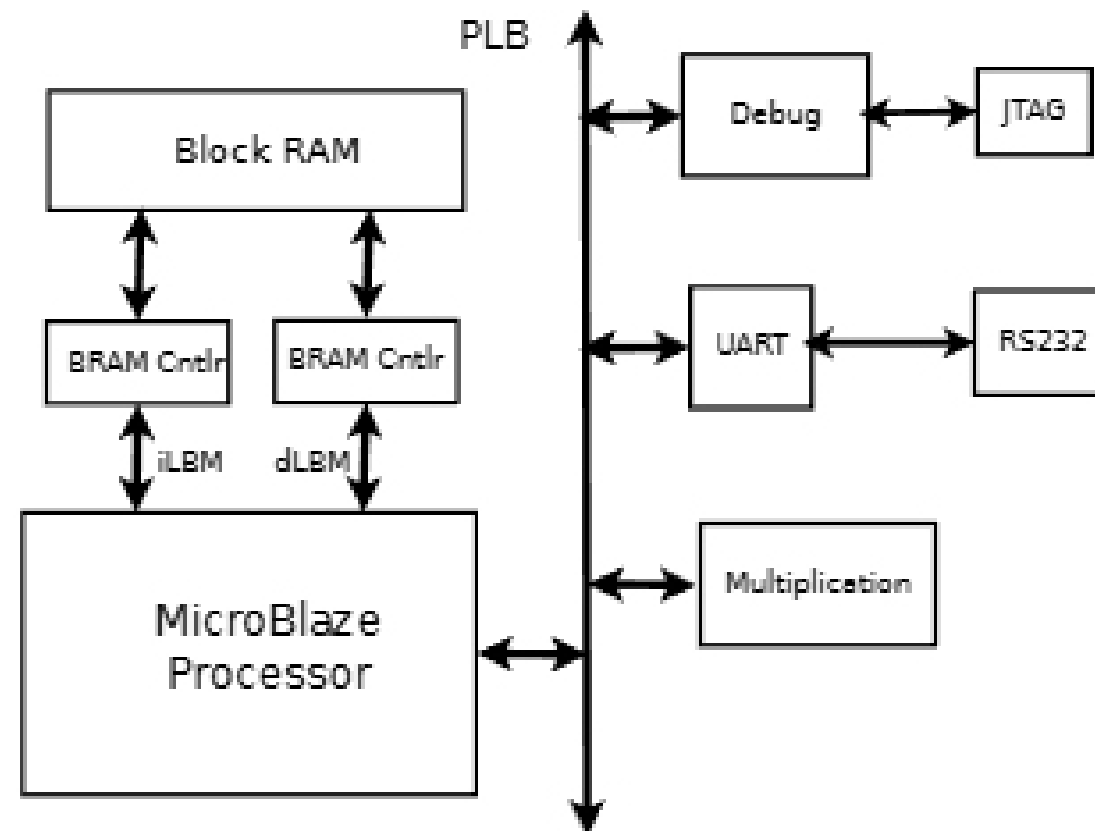


Figure 1: MicroBlaze System Diagram

Procedure

1. To begin, create the base system shown in Figure 1 using a similar configuration as last lab. We do not have the multiplication peripheral yet but will add that after we create it.
 - (a) Create a folder for Lab 3 and open XPS as outlined in Lab 2.
 - (b) Use the Base System Builder (BSB) to create a system similar to that which you built in Lab 2. Ensure a UART is the only peripheral your system contains (i.e. no Ethernet MAC, PCIExpress, etc.).
 - (c) After the BSB completes, navigate to the 'System Assembly View' in XPS. Then click on the 'Addresses' tab. Change the size field on both the dlmb_cntrl and the ilmb_cntrl to 128K. Hit 'Generate Addresses' to regenerate the address ranges of the other components on the bus.
2. At this point, we should have everything in Figure 1 except for the multiplication peripheral. The next

few steps will guide you through the process of creating a custom IP peripheral. For today's lab, our custom peripheral will have three software accessible 32-bit registers and a hardware multiplication block. Two of the software registers will be read and write accessible by the microprocessor and will hold the multiplicand and multiplier. The third register will be read accessible by the microprocessor and write accessible by the multiplication logic within the peripheral. The third register will hold the product of the multiplication.

- (a) From the XPS main menu, select **Hardware**→**Create or Import Peripheral**. The 'Create and Import Peripheral Wizard' should come up (Figure 2). Hit 'Next' to continue.

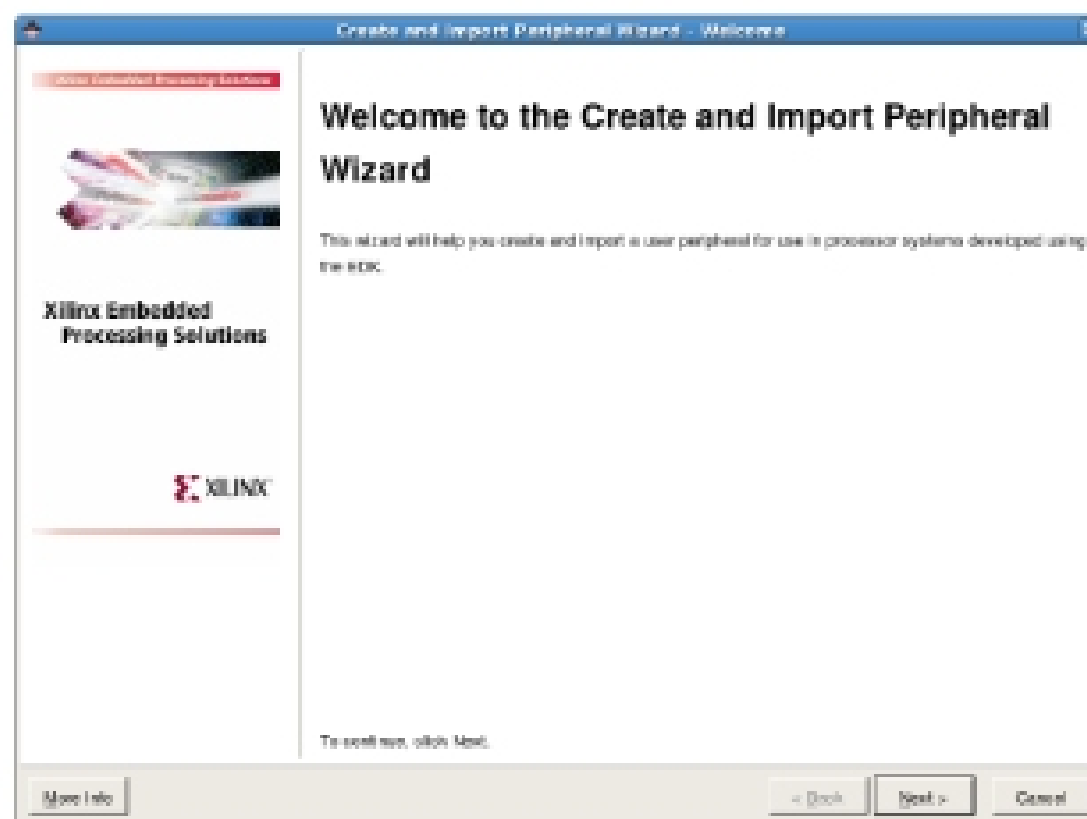


Figure 2: Create and Import Peripheral Wizard - Welcome

- (b) The next window allows us to either create templates for a new peripheral or import an existing peripheral. Ensure 'Create templates for a new peripheral' is selected and leave 'Load an existing .cip settings file' unchecked. Press 'Next' to continue.
- (c) The window that follows gives us the option to store our peripheral design files in a location other than our current XPS project directory. Leave everything as default and hit 'Next' to continue.
- (d) A window will appear prompting you to assign a name and version number to your peripheral (Figure 3). Name the peripheral 'multiply' and leave the version number as default (i.e. 1.00.a). You can enter a short description of your peripheral if you would like in the 'Description' window.