Atlys BSB Support Files for PLB-based Designs

DIGILENT®
www.digilentinc.com

Revision: October 30, 2012

Overview

This package will integrate board support for the Atlys Spartan-6 FPGA Development Board into Xilinx EDK tools. It includes board definition files for creating PLB-based MicroBlaze embedded designs in the Base System Builder (BSB). It also includes cores for custom peripherals such as the Digilent USB-EPP interface, the 16MB Quad SPI Flash Memory and the AC'97 Audio codec. With these files the BSB can be used to create Platform Studio projects initialized with cores that are properly configured to control the on-board peripherals. The currently supported cores are outlined in Table 1.

TABLE 1. BSB SUPPORTED PERIPHERALS

Peripheral	Peripheral name in BSB	Core name(s)	Notes
128MB DDR (cached)	MCB_DDR2	mpmc	
8 User Switches	DIP_Switches_8Bits	xps_gpio	--
5 User Push Buttons	Push_Buttons_5Bits	xps_gpio	--
8 LED outputs	LEDs_8Bits	xps_gpio	--
UART	RS232_Uart_1	xps_uartlite/xps_uart16550	--
PS2 Keyboard or Mouse interface through USB	PS2_Keyboard_Mouse	xps_ps2	BSB does not connect automatically the interrupt outputs. It is recommended that after the BSB is generated to connect manually the Interrupt outputs to an interrupt controller
10/100/1000 Mbps Ethernet PHY	Soft_TEMAC	xps_ll_temac	Requires a license, otherwise will stop functioning on the board after a period of time; for faster data handling it is recommended to use both DMA and interrupt
10/100 Mbps Ethernet PHY	Ethernet_MAC	xps_ethernetlite	Exclusive to Soft_TEMAC
16 MB SPI PCM	SPI_FLASH	xps_spi	1X mode only
16MB SPI PCM in Quad mode	Digilent_quad_spi_if	quad_spi_if	Custom core; supports 1X, 2X and 4X modes; exclusive to SPI_FLASH
Digilent USB-EPP interface	Digilent_Usb_Epp	d_usb_epp_dstm	Custom core; DSTM transfer mode support in future release
AC'97 Audio Codec	Digilent_ac97_if	ac97_if	Supports multiple sample rates

For additional information on using these cores, please refer to their PDF datasheets

Using the BSB Support Files

Note: For EDK versions older than 14.3, the screenshots below may differ. However, the relevance of the data entered into the text boxes and combo boxes remains the same.

Start Xilinx Platform Studio and create a new project using Base System Builder by selecting "Create New Project Using Base System Builder". The "Create New XPS Project Using BSB Wizard" will appear, as shown in Figure 1 below.
In this window:

1. Make sure that PLB system is selected

2. Click on the "Browse" button beside the "Project File" and select a folder where the system.xmp project file will be located
   Notes:
   a. It is **NOT** recommended to use a path which contains spaces, such as "My Documents" for the project folder, because a folder like this might affect the functionality of the EDK and SDK tools, from which many are linux-based.
   b. It is recommended to use an empty folder. The whole EDK hardware project will be stored in the same folder, therefore it is easier to archive and copy the whole project. Later, if preferred, SDK can also be set to use as project workspace a subfolder in the project root folder.
   c. By default, the project description file is "system.xmp". However, it can be given any name if the file extension is kept and obviously, does not contain spaces

3. Click on the "Browse" button beside the "Set Project Peripheral Repository Search Path" box and browse to the path containing the .\lib subfolder from the BSB PLB Support Files folder, then press OK. The BSB window should look like in figure below:
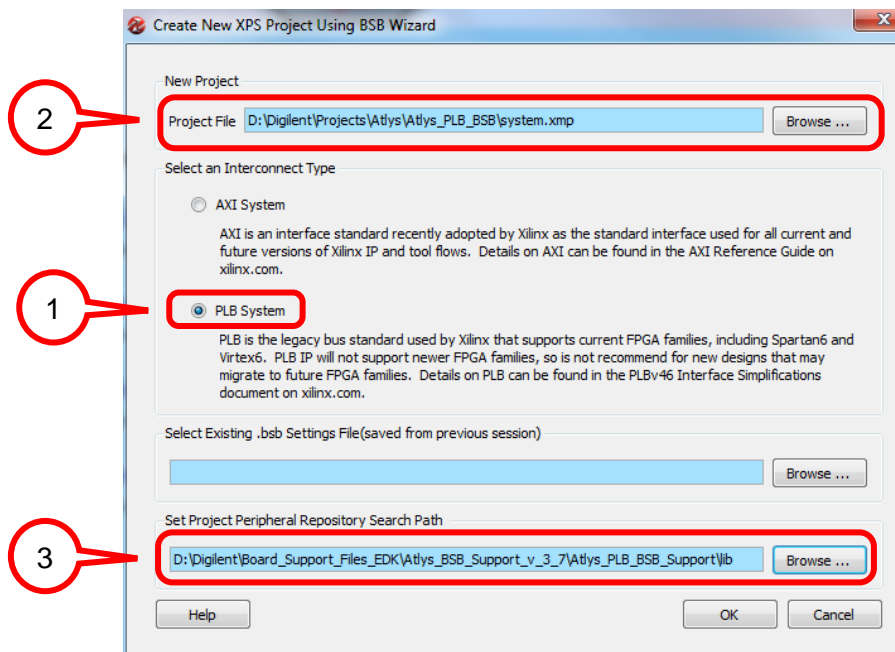

Figure 1. BSB window with specifying the Peripheral Repositor Search Path

Note: Obviously, the steps above can be made in any order until you don't click the OK button of the "Create New XPS Project Using BSB Wizard" window.

Click OK. You should now be able to select the Digilent Spartan-6 Atlys as your development board further in the Board Selection window.

Using Interrupt for the Digilent Usb-Epp interface

EPP requests for the USB-EPP interface come from the USB port. If there is no answer in 100mS, the PC application will signal a timeout. Therefore, it is recommended handling EPP requests with an interrupt service routine instead of continuously polling the interface status.

The demo applications include examples for using the USB-EPP interface in both polling and interrupt mode.

In order to use interrupt service routines, the interrupt request signal for the Digilent USB-EPP has to be connected to either an interrupt controller or the Microblaze processor interrupt input.

If the "Use Interrupt" option is selected for any core in BSB, then the Base System Builder will add an interrupt controller to the system.

For example, to select the "Use Interrupt" option for the Digilent_Usb_Epp peripheral in BSB, in the "Peripheral Configuration" window click on the "Digilent_Usb_Epp" peripheral and select "Use Interrupt" as shown in Figure 2 below.
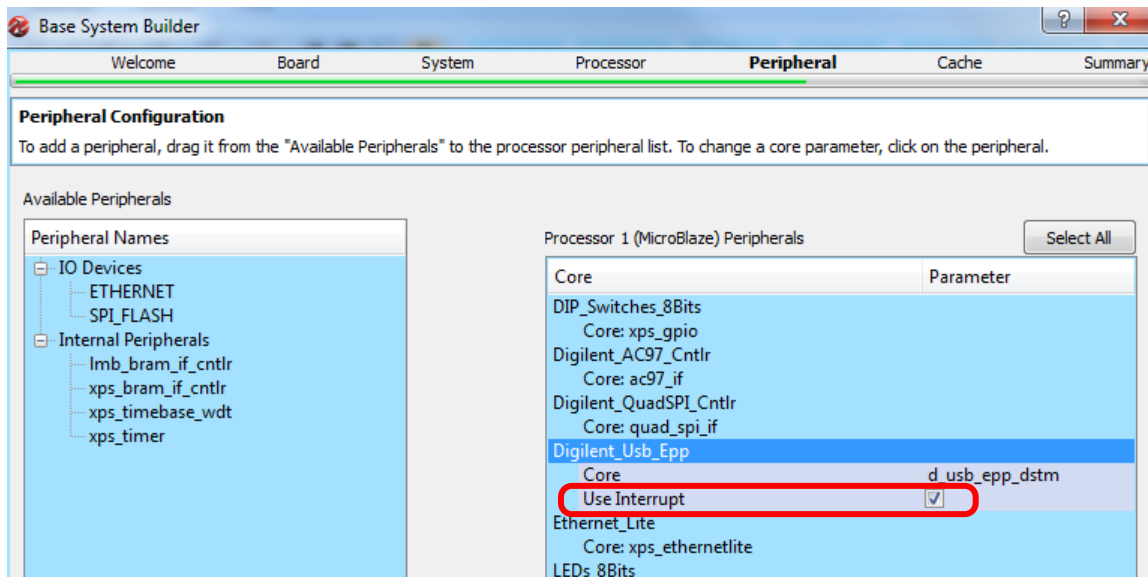

Figure 2. Connecting the interrupt output for the Digilent_Usb_Epp interface in Base System Builder.