Nexys2 BSB Support Files for PLB-based Designs

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Revision: September 9, 2011

Overview

This package will integrate board support for the Nexys2 Spartan-3E FPGA Development Board into Xilinx EDK tools, both for the 500 die and 1200 die version. 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 and the 4-digit Seven-Segment Interface. 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
User Reset			BTN0 is automatically assigned as the system reset input
16 MB PSRAM	Micron_RAM	xps_mch_emc	Any combination of these memories can be selected; a memory bus multiplexer resolves the shared buses;
16-MB Parallel PCM	INTEL_FLASH	xps_mch_emc	
8 User Switches	Switches_8Bit	xps_gpio	--
4 User Push Buttons	Push_Buttons_4Bit	xps_gpio	BTN0 is automatically assigned as the system reset input
8 LED outputs	LEDs_8Bit	xps_gpio	--
UART	RS232_Uart_1	xps_uartlite/xps_uart16550	--
Digilent Usb-Epp interface	Digilent_Usb_Epp	d_usb_epp_dstm	Custom core; DSTM transfer mode support in future release
Seven-Segment Decoder	Ssg_Decoder_0	Ssg_decoder	Custom core; supports individual digit blanking and blinking, autoblank, variable refresh rate and blink rate and individual segment data

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

Using the BSB Support Files

Use the BSB Support Files for PLB as a Project Peripheral Repository Search Path such as in the example below:

1. Start Platform Studio and create a new project in Base System Builder. Choose PLB system in the “Create New XPS Project Using BSB Wizard” window.
2. 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 Support Files folder, then press OK. The BSB window should look like in the figure below:

Copyright Digilent, Inc. All rights reserved. Other product and company names mentioned may be trademarks of their respective owners.

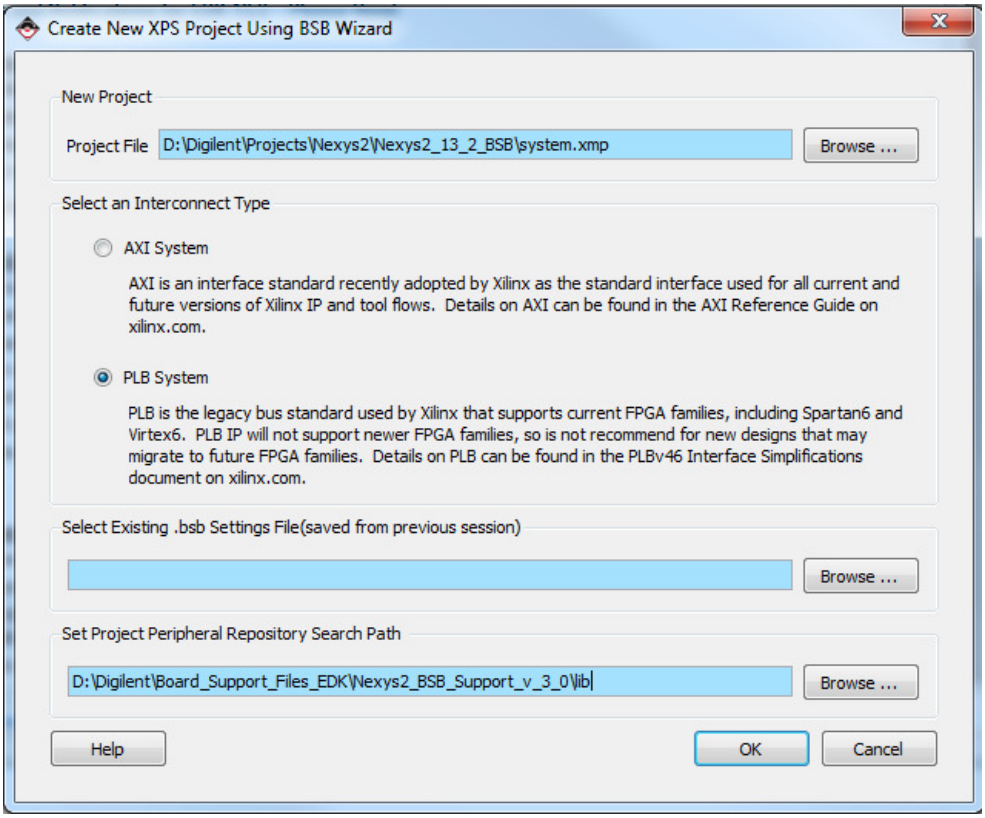

Figure 1. BSB window with specifying the Peripheral Repository Search Path

Click OK. You should now be able to select the Digilent Nexys3 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 are coming from the USB port. If there is no answer in about 100mS, the PC application will signal a timeout. Therefore it is recommended to handle the Epp requests with an interrupt service routine instead of continuously polling the interface status. 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.

This can be simply done in BSB in the “Peripheral Configuration” window. Click on the “Digilent_Usb_Epp” peripheral and select “Use Interrupt” like is shown in Figure 2 below.

Copyright Digilent, Inc. All rights reserved. Other product and company names mentioned may be trademarks of their respective owners.

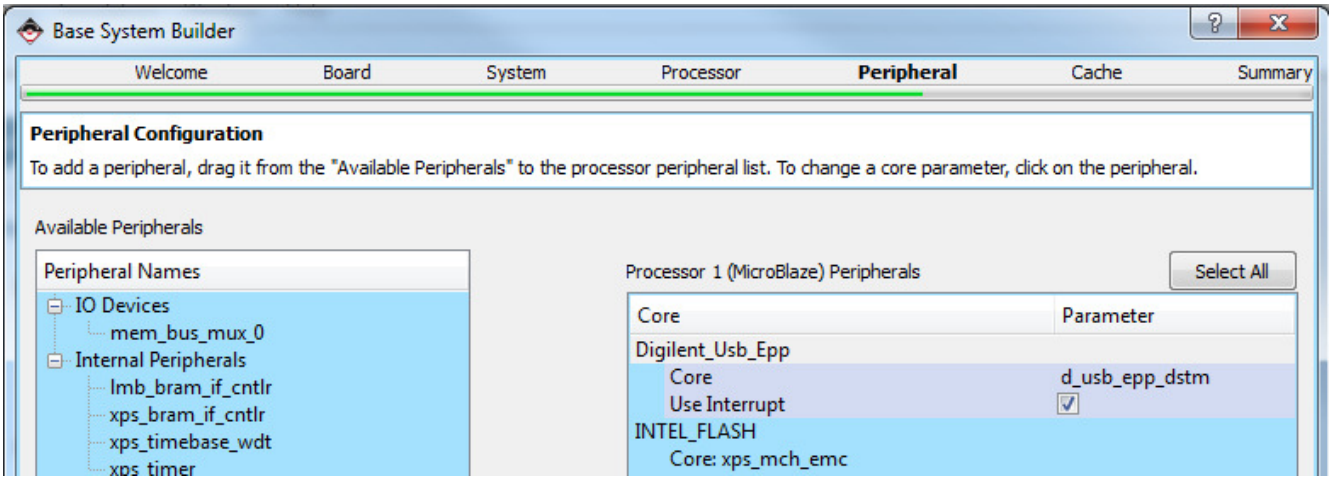

Figure 2. Connecting the interrupt output for the Digilent_Usb_Epp interface in Base System Builder.

In this case BSB will automatically add an interrupt controller to the system and connect the Usb_Epp interface interrupt output to the interrupt controller. The interrupt signal priority depends on what other interrupt signals are used in the system.

Note that it is also recommended to use interrupts for the PS2-HID interface if using a mouse, due to the increased data rate of the mouse. BSB does support yet automatically connecting the PS2 interrupt signals, but these signals can be manually connected after the Base System is generated, in the System Assembly view's Ports tab.

Notes about the mem_bus_mux core:

At the "Peripheral Configuration" window, in the "Available Peripherals" (left-side) pane appears a peripheral names mem_bus_mux_0, with the core named mem_bus_mux. This core is used to split and multiplex the address and data buses for the three memories present in the system.
The core is automatically added to the Base System and it **does not have to be added by the user.** Adding the core to the system does not affect the system functionality.

Copyright Digilent, Inc. All rights reserved. Other product and company names mentioned may be trademarks of their respective owners.