ECE 544: FreeRTOS system using Board Files

Vivado 2017.3

Revision 0.1

By

Sai Bodanki

(Edited by Roy Kravitz)

Disclaimer:

I have tried to be as accurate as possible and have taken into account the mistakes I made when I was building my first embedded system in my course work. I have always felt that learning by mistakes is a good way to learn. So don’t worry if you can’t get it to work in the first try, make mistakes and learn. If you are not able to get it to work by yourself, always go to the d2l discussion page or the Xilinx discussion forums. If you find any problems please communicate within yourself to solve it as much as possible. This document is mainly based on the previous versions provided by Prof. Roy Kravitz.

Revision History:

07-Feb-2018	SB	0.0	Created this file for ECE 544 Project #2
09-Feb-2018	RK	0.1	Edited instructions based on my experiences

Free RTOS System:

A FreeRTOS system needs at-least the following Hardware support:

- Microblaze or Zynq, mdm, AXI bus support, clock and reset generator, etc.
- Memory - DDR SDRAM + On-Chip BRAM (128Mb + 128KB)
- Interrupt Controller
- Timer

We are going to build our Project #2 system using board files instead of n4fpga.v and n4ddrfpga.xdc. This is a different way to create IP Integrator-based projects in Vivado but is convenient in cases like ECE 544 where the design is based on a 3rd party FPGA development board.

What are Board files?

After installing Vivado, the default installation directory on your drive will contain a folder called `board_files`. If Vivado is installed on the `C:` drive (recommended) the board_files folder can be found here: `C:\Xilinx\Vivado\2017.3\data\boards`.

By default, this folder contains XML files for different FPGA boards supported and sold by Xilinx. These XML files define the different interfaces on the board (interfaces such as Slide Switches, Push Buttons, LEDs, USB-UART, DDR Memory, Ethernet etc.) and assign them to Xilinx IP blocks (ex: the buttons, switches, and LEDs on the Nexys4 DDR are assigned to AXI_GPIO blocks.

FPGA development boards vendors like Digilent provide Board Files for their development boards to make the task of creating an embedded system with the IP Integrator easier. When you create a new project in Vivado using the board file for your development board you save quite a bit of work:

- No need to add pin constraints for any of the peripherals supported in the Board file. You would need to add constraints for 3rd party and your own customized IP (ex: you would have to add a constraints file if you included Nexys4IO in your design).
- IP is customized (ex: DDR SDRAM) to the recommended configuration for the development board by default.
- The correct FPGA component (incl. the speed grade) is selected by default.
- A top-level wrapper file (similar to n4fpga.v) can be generated automatically by Vivado.
- The desired features can be included (or not) into the embedded system.

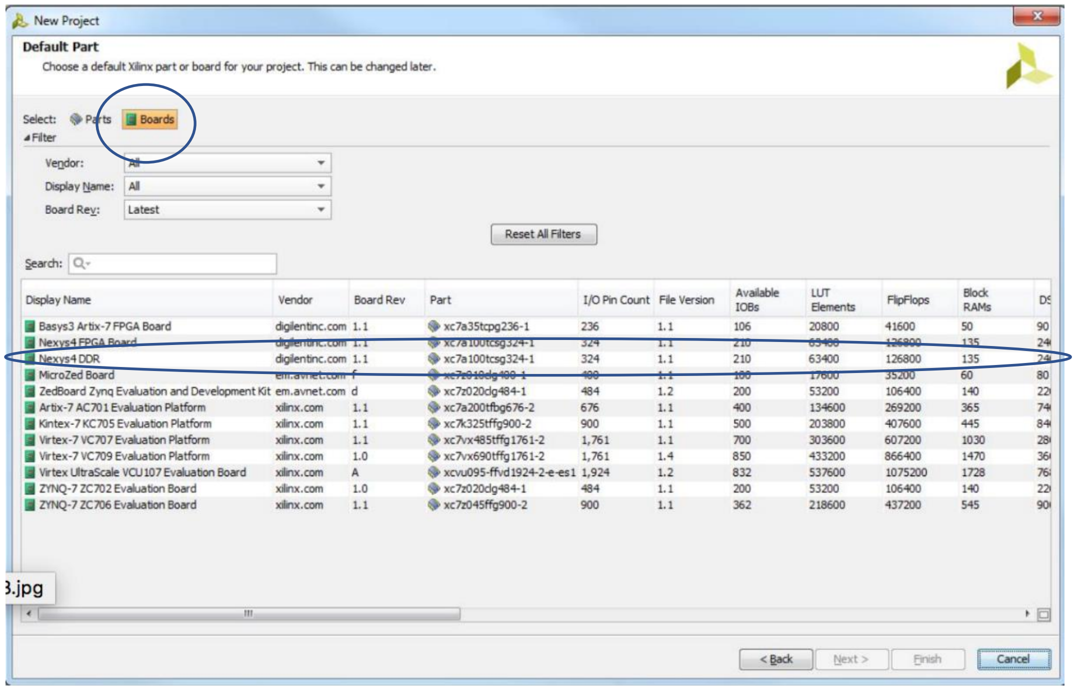
Creating a Vivado project with a board file means you can focus your effort on applying the embedded system, not creating the embedded system from scratch. If you are working on a custom FPGA-based board (not a development platform) you have to start from scratch, pretty much the same way we did Project #1 (We did, however, give you a head start on your Project #1 system by providing a functional top level file and constraint file).

Installing the Digilent Board Files

Download and extract this Zip file: https://github.com/Digilent/vivado-boards/archive/master.zip

This zip file will contain a folder called new/board_files. Save this in your user documents folder. Copy/Paste the entire board_files folder to the board_files folder inside the Vivado Installation directory and merge them both.

Once you have done this you should be able to see an updated list of board files including Nexys4 DDR and other boards supported by Digilent. You may have to restart Vivado to do this.



Vendor: All
Display Name: All
Board Rev: Latest

Display Name	Vendor	Board Rev	Part	I/O Pin Count	File Version	Available IOBs	LUT Elements	FlipFlops	Block RAMs	DS
Basys3 Artix-7 FPGA Board	digilentinc.com	1.1	xc7a35tcpg236-1	236	1.1	106	20800	41600	50	90
Nexys4 FPGA Board	digilentinc.com	1.1	xc7a100tcsg324-1	324	1.1	210	63400	126800	135	24
Nexys4 DDR	digilentinc.com	1.1	xc7a100tcsg324-1	324	1.1	210	63400	126800	135	24
MicroZed Board	em.avnet.com	f	xc7z010clg400-1	400	1.1	100	17600	35200	60	80
ZedBoard Zynq Evaluation and Development Kit	em.avnet.com	d	xc7z020clg484-1	484	1.2	200	53200	106400	140	22
Artix-7 AC701 Evaluation Platform	xilinx.com	1.1	xc7a200tfbg676-2	676	1.1	400	134600	269200	365	74
Kintex-7 KC705 Evaluation Platform	xilinx.com	1.1	xc7k325tffg900-2	900	1.1	500	203800	407600	445	84
Virtex-7 VC707 Evaluation Platform	xilinx.com	1.1	xc7vx485tffg1761-2	1,761	1.1	700	303600	607200	1030	28
Virtex-7 VC709 Evaluation Platform	xilinx.com	1.0	xc7vx690tffg1761-2	1,761	1.4	850	433200	866400	1470	36
Virtex UltraScale VCU107 Evaluation Board	xilinx.com	A	xcvu095-ffvd1924-2-e-es1	1,924	1.2	832	537600	1075200	1728	76
ZYNQ-7 ZC702 Evaluation Board	xilinx.com	1.0	xc7z020clg484-1	484	1.1	200	53200	106400	140	22
ZYNQ-7 ZC706 Evaluation Board	xilinx.com	1.1	xc7z045ffg900-2	900	1.1	362	218600	437200	545	90

Steps to Build the System

Step	Screen	Action	Explanation/Comments
1	Vivado Home	Create a new Project	You know the Drill
2	Part Select	Select the Boards tab and pick the Nexys4 DDR as the target platform. Click *Next* and *Finish*	
3	Create New Block Design	Click on Flow Navigator/IP INTEGRATOR/Create Block Design. In the BLOCK DESIGN PANE containing tabs for Sources, Design, Signals, and Board Click on *Board*. You will see a scrolling list of all of the peripherals supported by the Nexys4 DDR board file.	.
4	Click Clock folder→systemclock	Double Click on System Clock and then Click OK to accept the defaults. Right Click on the clock wizard block in the Schematic pane and select *Customize IP*. A Clocking Wizard appears. Pick the configurations based on previous configurations (project 1). Add a 200MHz (for DDR SDRAM) and a 50MHz clock (for the PmodOLEDrgb SPI clock to the 100MHz system clock. Deselect the *reset* input to the clock wizard.	
5		Click *OK* to accept the remaining default parameters.	NA
6	Add IP	Add a Microblaze IP with the following configuration: • Local Mem -128KB • Cache -32KB • Debug and UART • Interrupt Controller	You can further customize the IP block to add hardware floating point and perhaps a a few hardware breakpoints.
7	BLOCK DESIGN/Board	Double Click on External Memory/DDR2 SDRAM and accept the default	

		(`mig_ddr_interface`) by pressing *OK*. Run Connection Automation. Customize the MIG block and set the following configuration: • Click on *Next* until you see A memory type parameter. Select DDR2 SDRAM and click *Next* until you see a Change period window. Set the period to 5000 (200MhZ) • Click *Next* until you see a Validate button. Click *Validate* (You should see Current Pinout is valid) • Click *Next* and *Agree*	
8	Run Connection Automation	You should see a window where you can select `sys_clk_i` of `MIG_7_Series` Select your 200MHz clock as the source. Click on *OK*.	
9	IP catalog	Add an AXI Timer with one Timer Enabled. Connect its interrupt to the Concat block	This will be the "tick" for the FreeRTOS system
10	IP Catalog	At a FIT Timer. Can generate 4Khz interrupts. Run Connection Automation	Can be changes according to Project need
11	BLOCK DESIGN Board tab.	Add a PmodOLEDrgb and a PmodENC to the Jx ports you want to connect them to. Use a PmodGPIO for the PmodHB3. Run connection automation.	A list of all the available Pmods from Digilent will pop up when you select one of the Pmod connectors on the board.
12	BLOCK DESIGN Board tab.	Add LEDs, pushbuttons, DIP switches. Customize the GPIO connected to the pushbuttons and DIP switches to include interrupts. Route the interrupt signal from the GPIO to the Concat block.	
13	BLOCK DESIGN Board tab	Add a UARTLite. Configure the UARTLite for 11500-N-8-1	
14	IP Catalog	Add an AXI_Timebase_WDT. Route the interrupt to the Concat block and route the `WDT_Reset` signal to `aux_reset_in` on the Processor Reset block. Configure the	

		WDT timeout to 2-3 seconds. Run Connection Automation	
14	BLOCK DESIGN Schematic	Complete the connections in your embedded system by routing any additional interrupts, making the port signals external, etc.	This will vary depending on how you designed the embedded system. Check your schematic to make sure you have correctly connected all of the signals.

Note: Generating the EN (PWM) signal could affect the connections to the PmodGPIO interface used for the PmodHB3. For example, while the PmodGPIO can be used to generate the DIR (Direction) signal to the PmodHB3 and to return the value of the Hall Sensor, you will have to connect your PWM signal (no matter how you create it) to the PmodHB3, as well. One way we can think of is to co-opt (take over) the GPIO I/O buffer that goes to the EN pin on the PmodHB3 for your PWM signal. This could be done in the embedded system wrapper that is produced by the IP Integrator.

Once you are sure you have added all of the peripheral your application needs and have connected the signals properly check that all of the peripheral devices have gotten assigned to unique addresses and that the DDR2 SDRAM has a valid address range. In the Address Editor you see the Instruction has MIG and BRAM. Anything more than these can be excluded from the segment. (Right click and Select Exclude). Generate Output products (this will take a long time)

- You can now Save your design and go to sources.
- Right Click on your design and select create HDL wrapper. Select *Copy generated wrapper to allow user edits* so that you can make any edits required for your implementation.
- Let Vivado pick the top level module and select *OK*.
- BINGO!! Your system is now ready to build.