Developing Zynq Software with Xilinx SDK

Lab 8

SDK Project Management

November 2013
Version 03

Lab 8 Overview

You should not share or archive your workspace simply by zipping it up and sending it off. Workspaces are just a container of software projects, and your preferences are user- and location-specific. If you simply copy the workspace to a different location in the system, it is not guaranteed to work.

Software projects, including board support packages, and software applications that you create in your workspace, **can** be shared with other team members or archived into a source control system. To do this, share or archive a collection of source files and SDK metadata files in the project directory. This lab will show you the steps how to do this.

Lab 8 Objectives

When you have completed Lab 8, you will know how to:

- Create a complete project archive
- Create a new, duplicate project by importing your archive
- Create a new application, and import the sources for that application

Experiment 1: Create a Complete SDK Project Archive

This experiment will show you the proper method for archiving a project for sharing. It is important to note that this is a multi-step process.

I. Archive the project sources in an archive file

II. Archive the Run/Debug configurations in an archive directory, if desired

III. Archive debug breakpoints, if desired

Experiment 1 General Instruction:

Export the SDK projects to a single-file archive, including the hardware platform, BSP, and all applications.

Experiment 1 Step-by-Step Instructions:

The version of Eclipse used in SDK 2013.3 does not always properly refresh the projects in the workspace to accurately reflect the status of that project. Before archiving, we will force a manual refresh to make sure everything in the workspace is properly synchronized with SDK.

1. Select the **Hello_Zynq** project in *Project Explorer*.

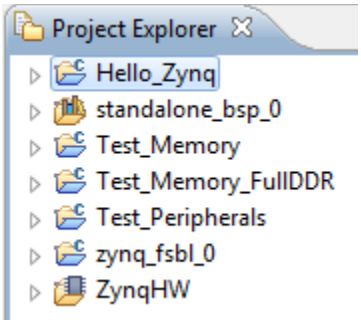

Figure 1 – Select All Projects

2. Select **File → Refresh** or press F5 to force a refresh.

3. Repeat Steps 1 and 2 for the other six projects to refresh them all. It would have been easier to select them all and refresh them all at once, but that doesn't work in this version of SDK.

4. One cross-check is to check that all your boot images are visible. In Project Explorer, browse to **Test_Peripherals → bootimage** and see if the three files all show up.

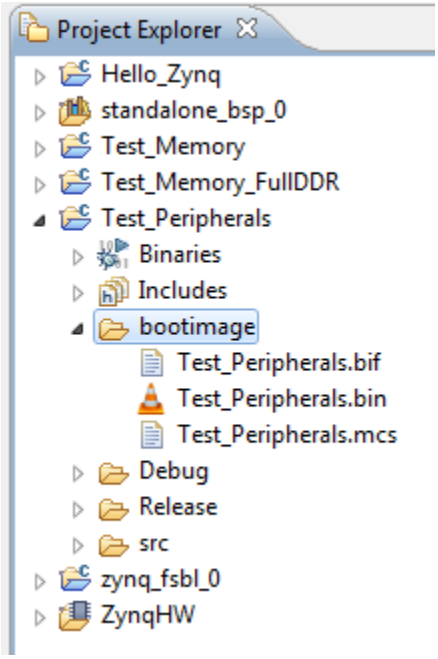

Figure 2 – Check that the bootimage Folder Lists BIF/BIN/MCS

5. In SDK, select **File → Export**.

6. Select **General → Archive File**. Click **Next >**.

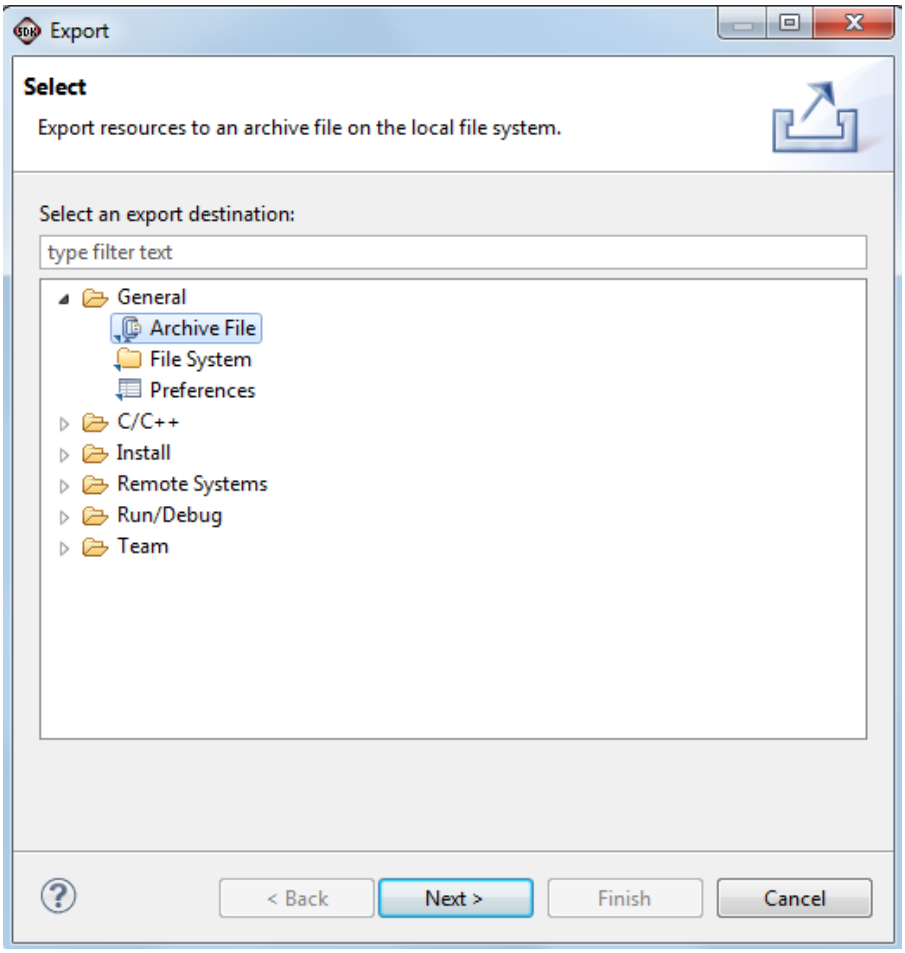

Figure 3 – Export Archive File

7. Select all the checkboxes or click the **Select All** button. Click **Browse** and name the archive 'Lab08_project_export.zip' in the `ZynqSW\2013_3` Speedway directory. Click **Save**. Click **Finish**.

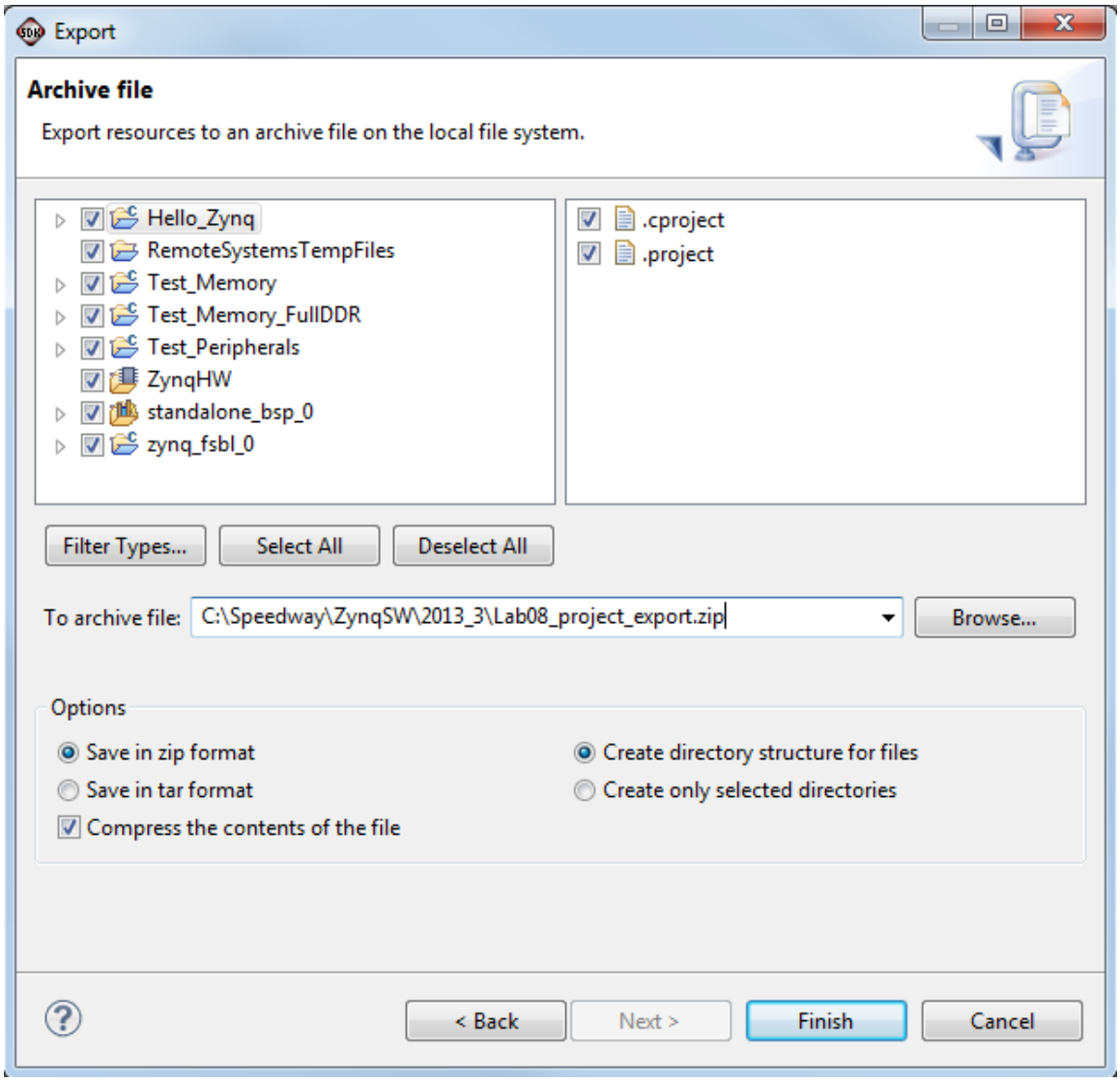

Figure 4 – Export All

Next, we'll store the Run and Debug Configurations that we previously created. This is an optional step.

8. Select **File → Export**. Select **Run/Debug → Launch Configurations**, then click **Next >**.

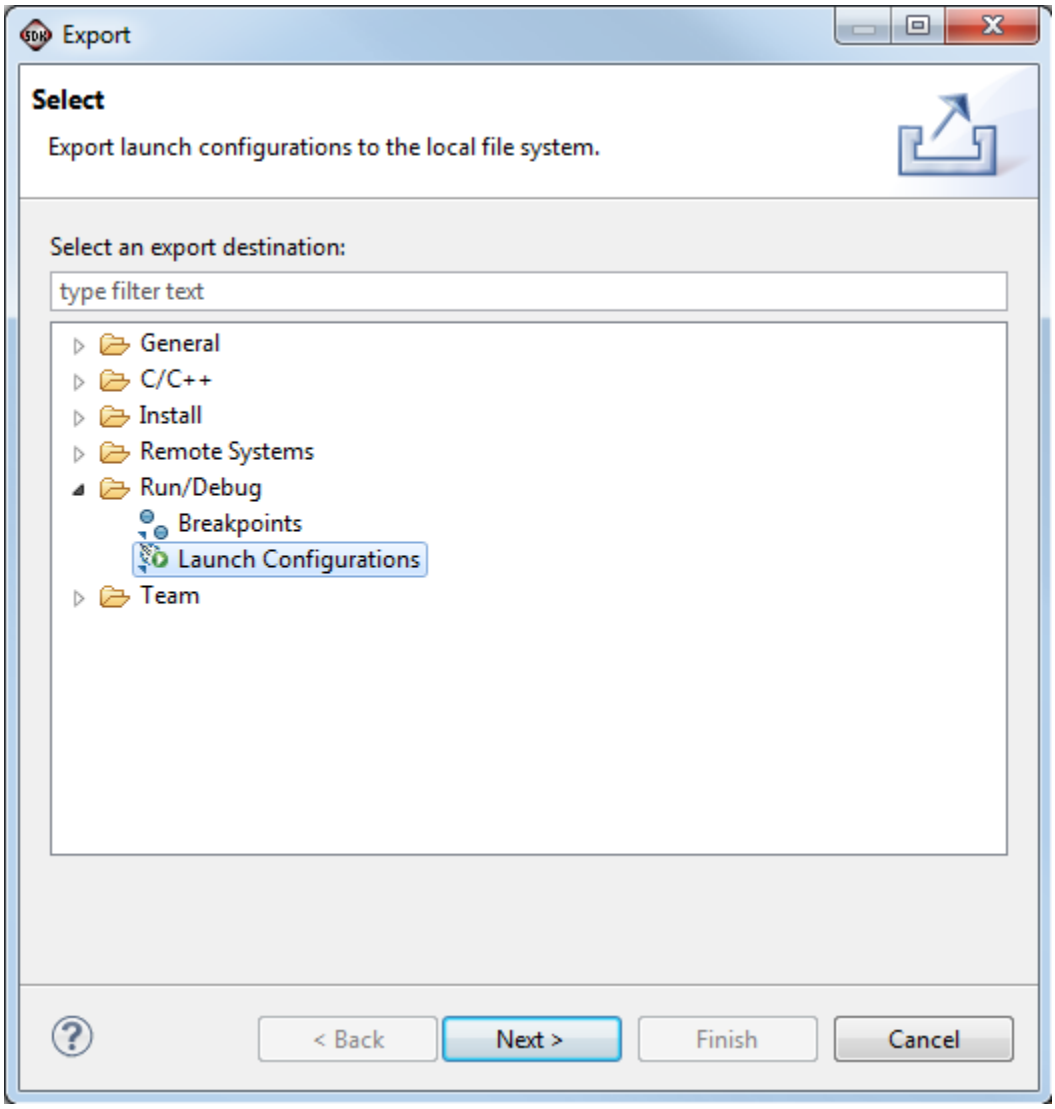

Figure 5 – Export Run/Debug Launch Configurations

9. Click the **Select All** button. Browse to `C:/Speedway/ZynqSW/2013_3/`. Click **Make New Folder**, and name it `Lab08_config_export`. Click **Finish**.

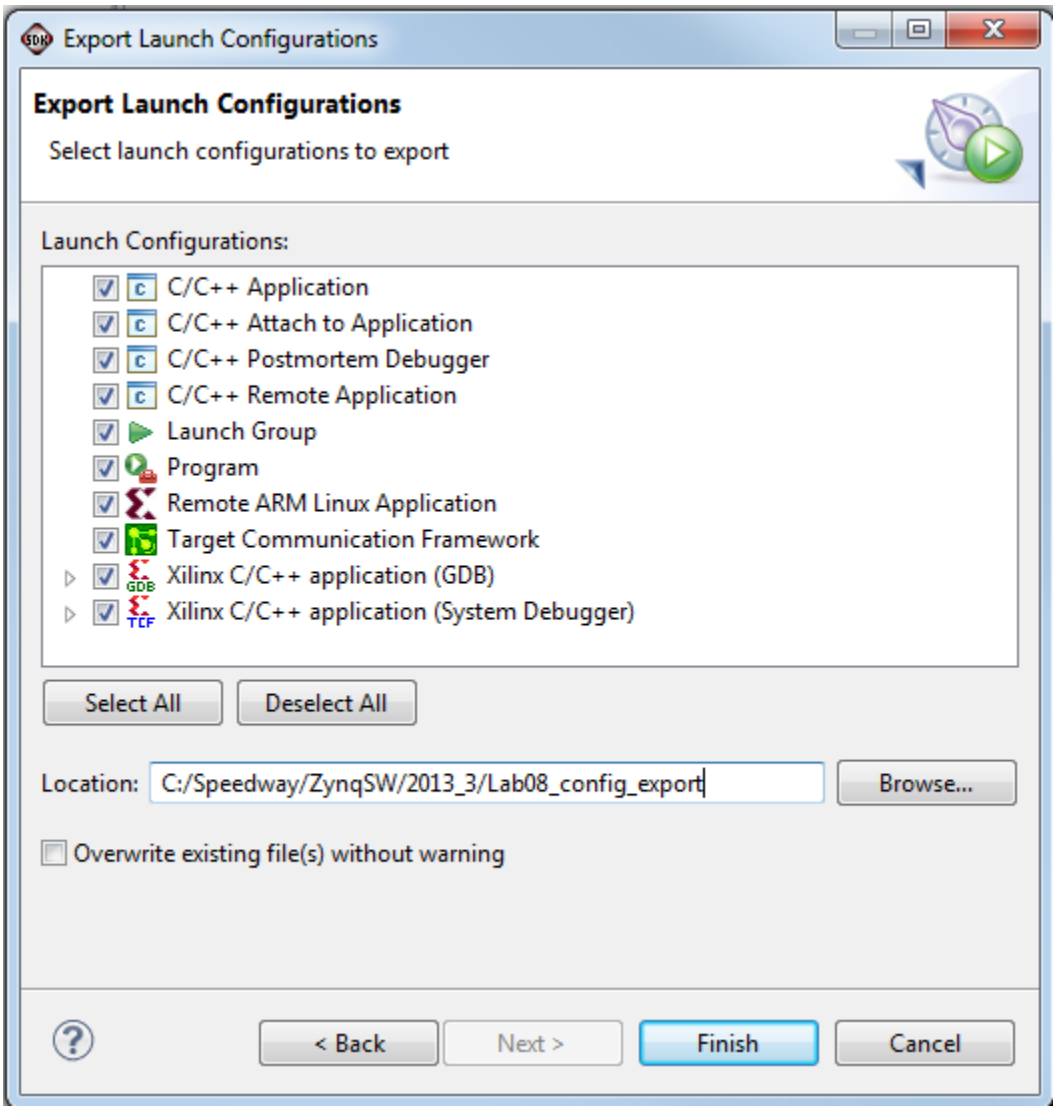

Figure 6 – Export Launch Configurations

Although our Debug exercise did not include a lot of breakpoint setting, you can imagine that some engineers invest a lot of time in setting up a debug environment. If they wanted to transfer that environment, including breakpoints, those must be exported explicitly.

10. Select **File → Export**. Select **Run/Debug → Breakpoints**, then click **Next >**.

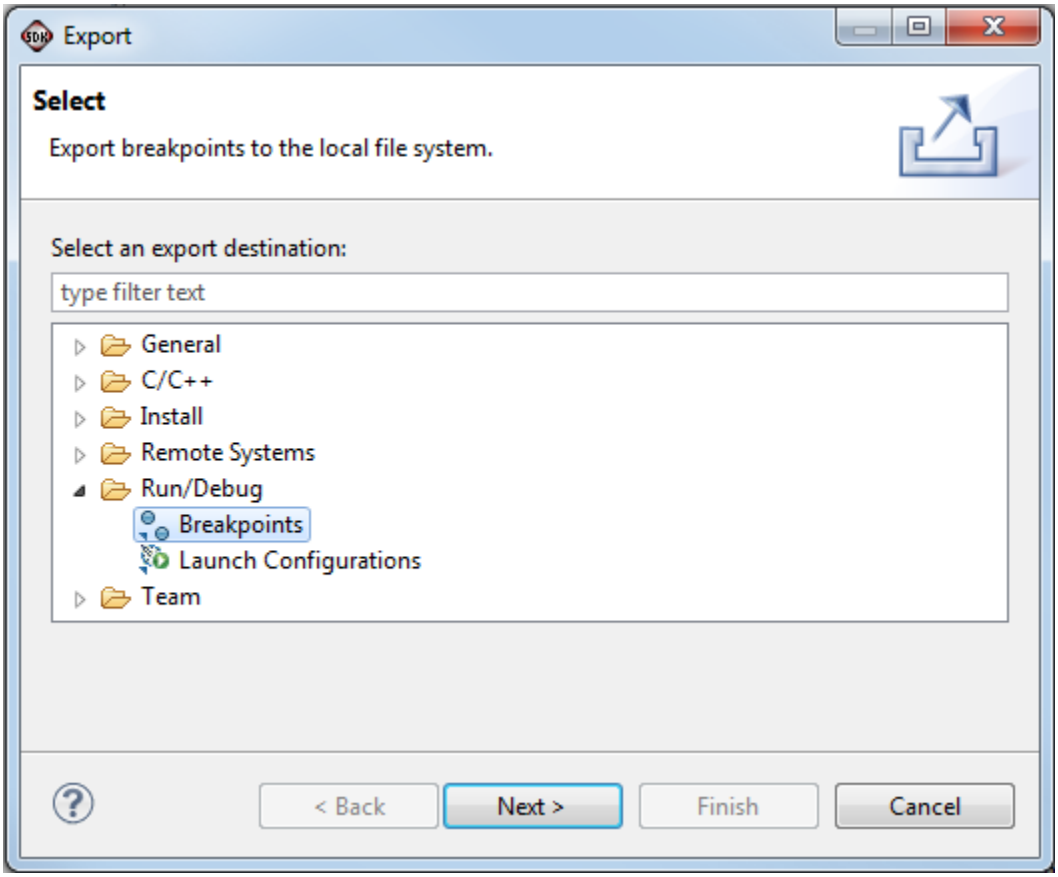

Figure 7 – Export Breakpoints

11. Click the **Select All** button. Browse to `C:/Speedway/ZynqSW/2013_3/`. Name the file `Lab08_breakpoint_export.bkpt`. Click **Finish**.

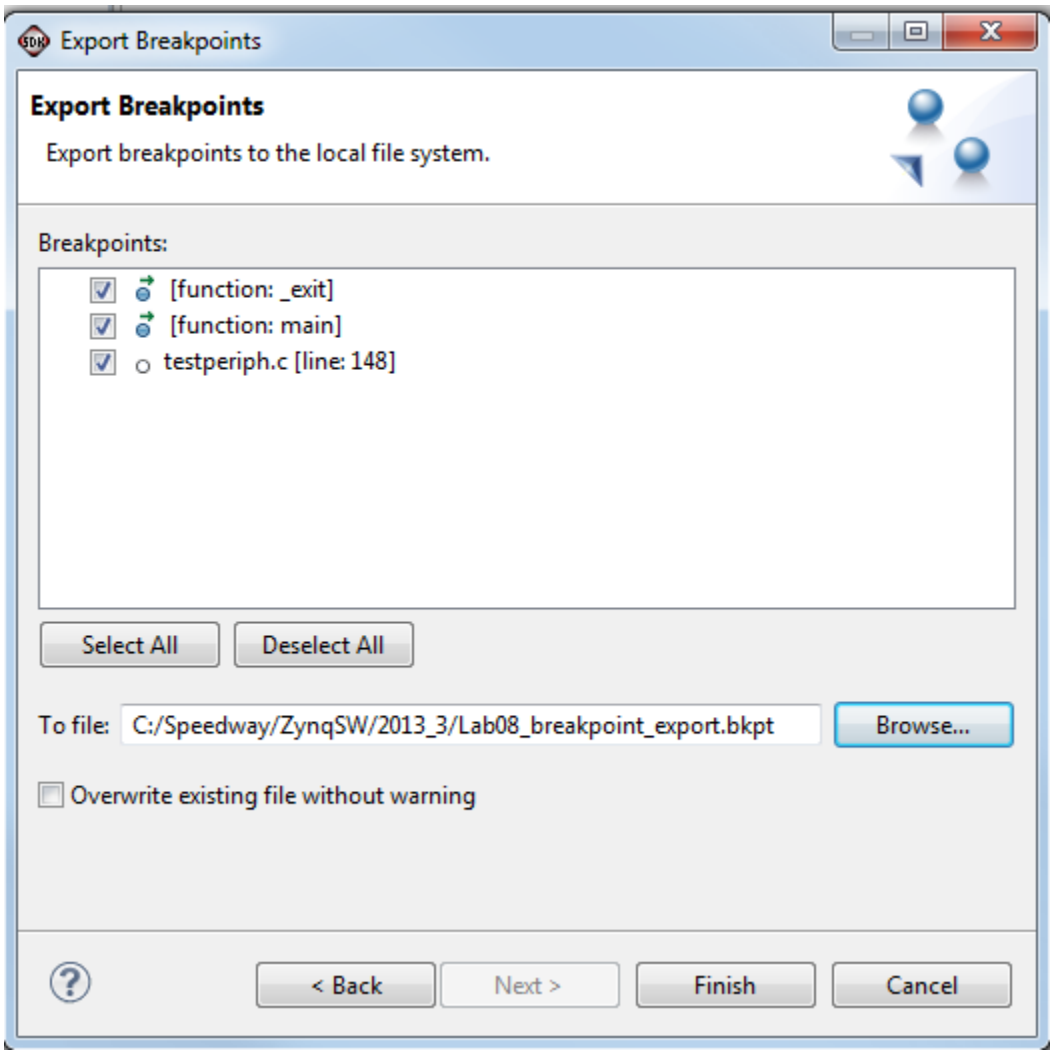

Figure 8 – Export Breakpoints

12. One final, manual step is required. Your project currently does not make use of any repositories, but if it did, these are not captured in any of the Exports that you have already created. A specific File → Export menu option does not exist to capture the repositories. However, this information is contained in an underlying file. You should save this file as well as any repositories referenced. Browse in Windows Explorer from your WorkSpace to:

```
.metadata\.plugins\org.eclipse.core.runtime\.settings
```

13. Create a folder called **Lab08_repositories_export**. Make a copy of file **`com.xilinx.sdk.tools.prefs`** (file **`com.xilinx.sdk.sw.prefs`**

doesn't exist yet) in this folder. Incidentally, these files also contain the customizations that you made when creating the new linker scripts for your applications. Restoring these file will also allow the tools to remember your Linker Script Generator settings.

14. Browse to `C:/Speedway/ZynqSW/2013_3/` in Windows Explorer to view what was created. Note the two files and two directories that were created. You can archive all four of these items together, along with any repositories you were using, to share with a colleague to recreate your SDK workspace.

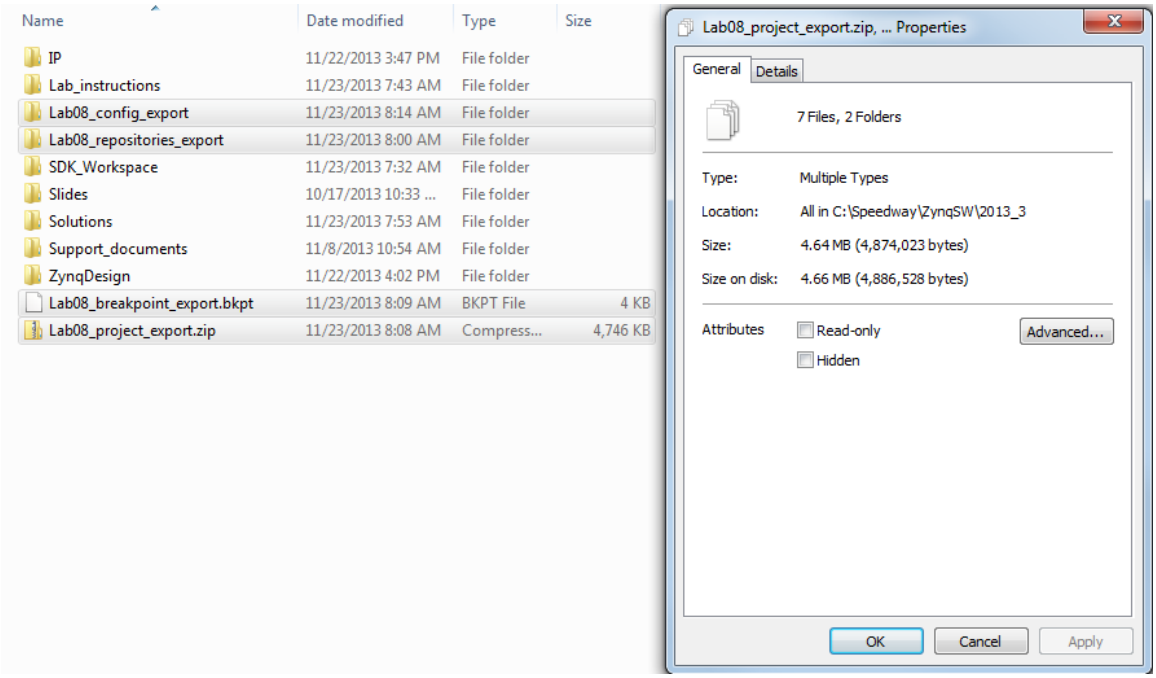

Figure 9 – Archive Files/Directory Created

Question:

Answer the following question:

- *What is the advantage of exporting your Workspace items as opposed to simply zipping the workspace?*

_____

_____

_____

Experiment 2: Import a Shared Project Archive

Now that you've created an archive, we'll next learn how to import an archive. We will assume that you now are the recipient of the archive files and directory created in Experiment 1. What steps are necessary to duplicate the workspace in a different location?

Experiment 2 General Instruction:

Create a new workspace and import all the previous projects from the lab8_export.zip archive.

Experiment 2 Step-by-Step Instructions:

1. Select **File → Switch Workspace → Other**.
2. Call it 'New_Workspace' then click **OK**.

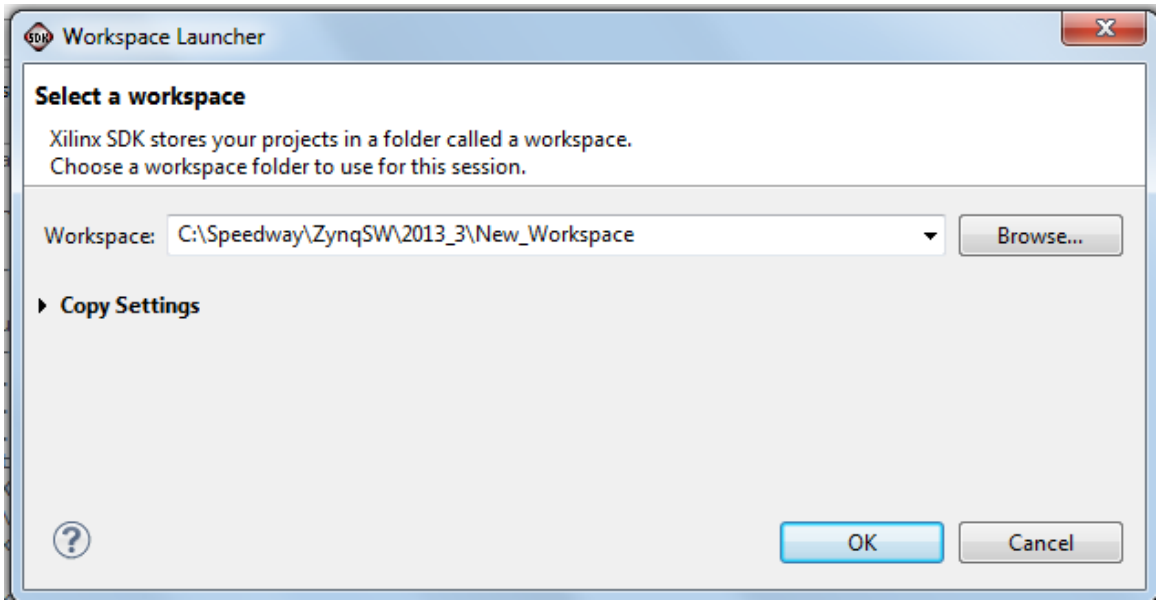

Figure 10 – New Workspace Created

3. Close the *Welcome* screen, if necessary. You should observe that you have a new workspace with no projects – no hardware platform, no BSP, and no applications.

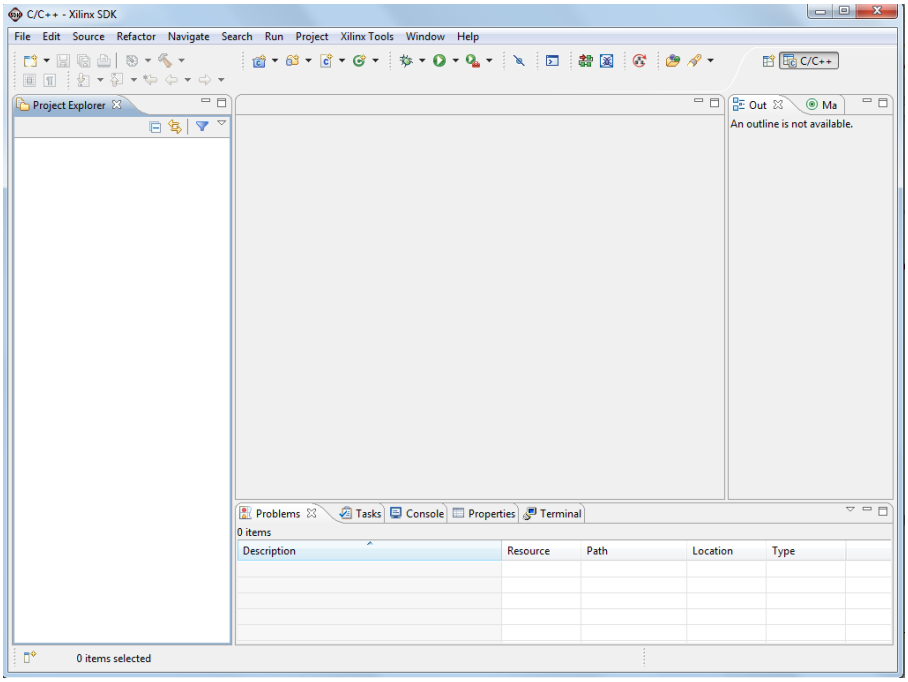

Figure 11 – New Workspace is Empty

4. Select **File → Import**. Select **General → Existing Projects into Workspace**, then click **Next >**.

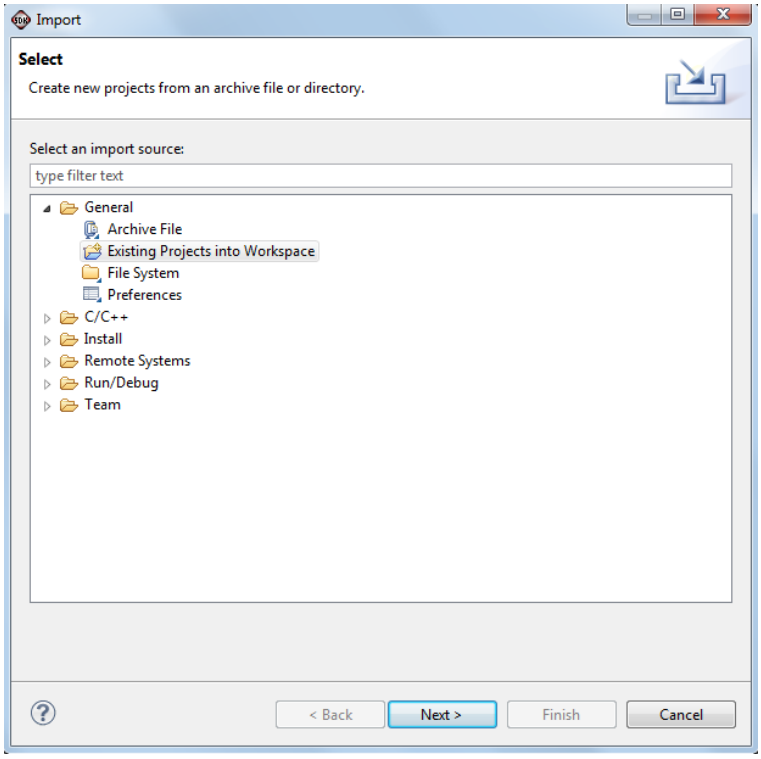

Figure 12 – Import from Archive File

5. Select the radio button for **Select archive file**. Browse, select, and open `Lab08_project_export.zip`. Click the **Select All** button. Click **Finish**.

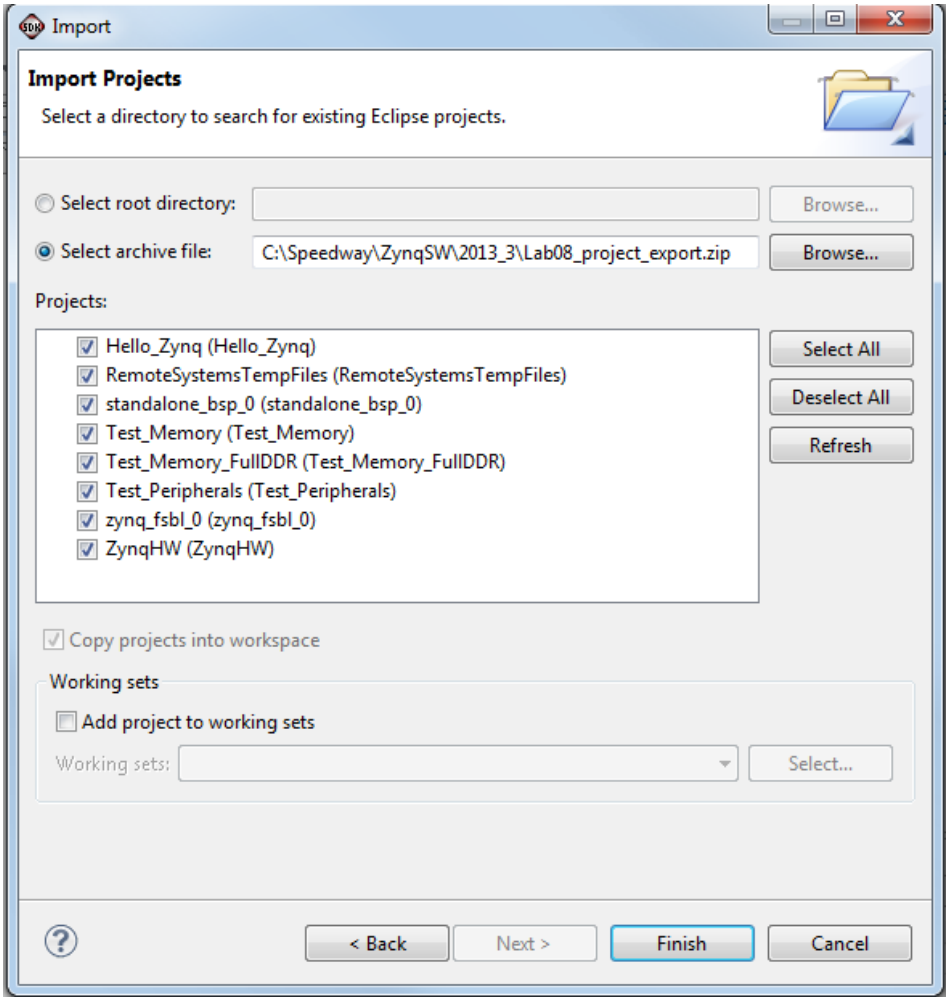

Figure 13 – Import from Archive File

6. You should now see your Projects restored in *Project Explorer*, as shown below.

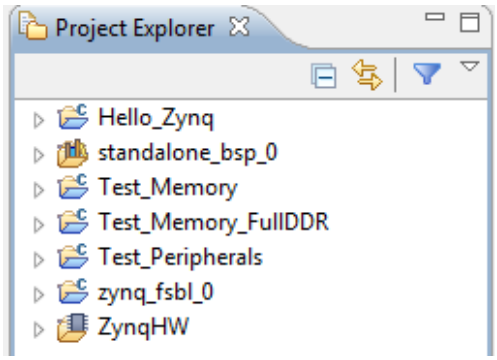

Figure 14 – Projects Restored

7. All of the projects should import and then automatically build. You may notice an ERROR that briefly displays. This is a result of importing all the projects at once, with the software applications and BSP being imported before the hardware platform because they got imported in alphabetical order. We could have avoided this by importing the ZynqHW hardware platform first, and then the rest. However, this ERROR is benign and self-corrects as soon as the hardware platform is imported.

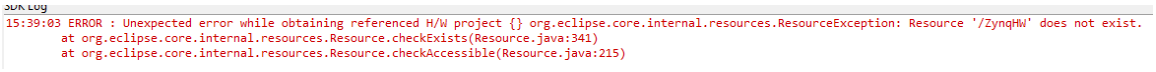

Figure 15 – Error from Importing ZynqHW Last

8. Select **File → Import**. Select **Run/Debug → Launch Configurations**, then click **Next >**.

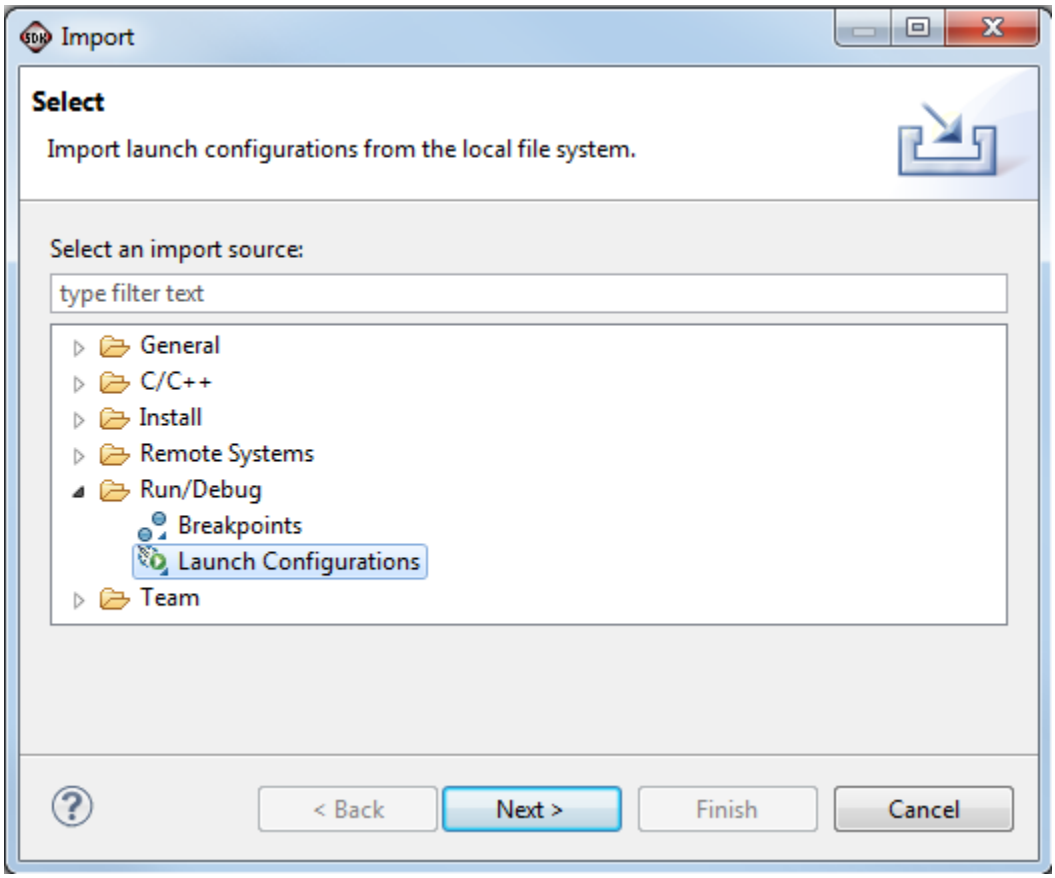

Figure 16 – Import Launch Configurations

9. Browse to `C:/Speedway/ZynqSW/2013_3/Lab08_config_export` and click **OK**. Select the **Lab08_config_export** checkbox. Verify that the four configurations on the right also have their checkboxes checked. Click **Finish**.

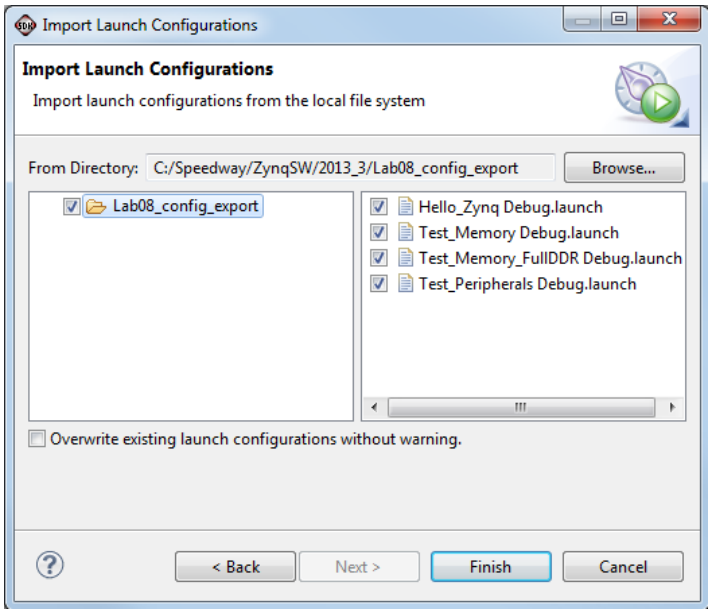

Figure 17 – Import Launch Configurations

10. Verify the Configurations were imported by selecting **Run → Debug Configurations**. Then expand both **Xilinx C/C++ application** items (GDB and System Debugger). You should see four Debug configurations (three under GDB and one under System Debugger). Click **Close**.

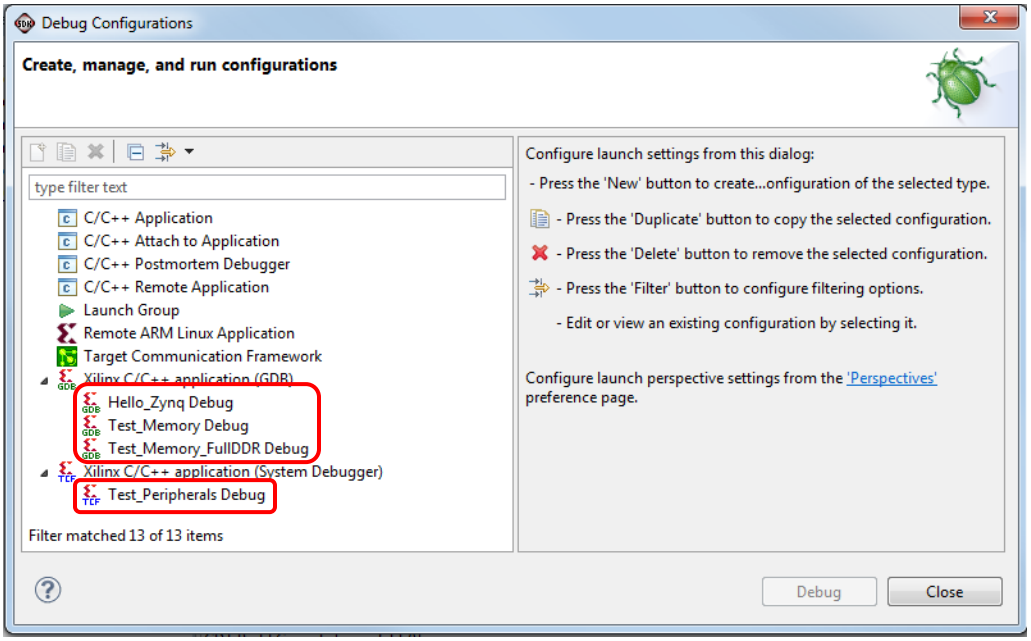

Figure 18 – Run/Debug Configurations Restored

Next, we'll restore the breakpoints.

11. Select **File → Import**. Select **Run/Debug → Breakpoints**, then click **Next >**.

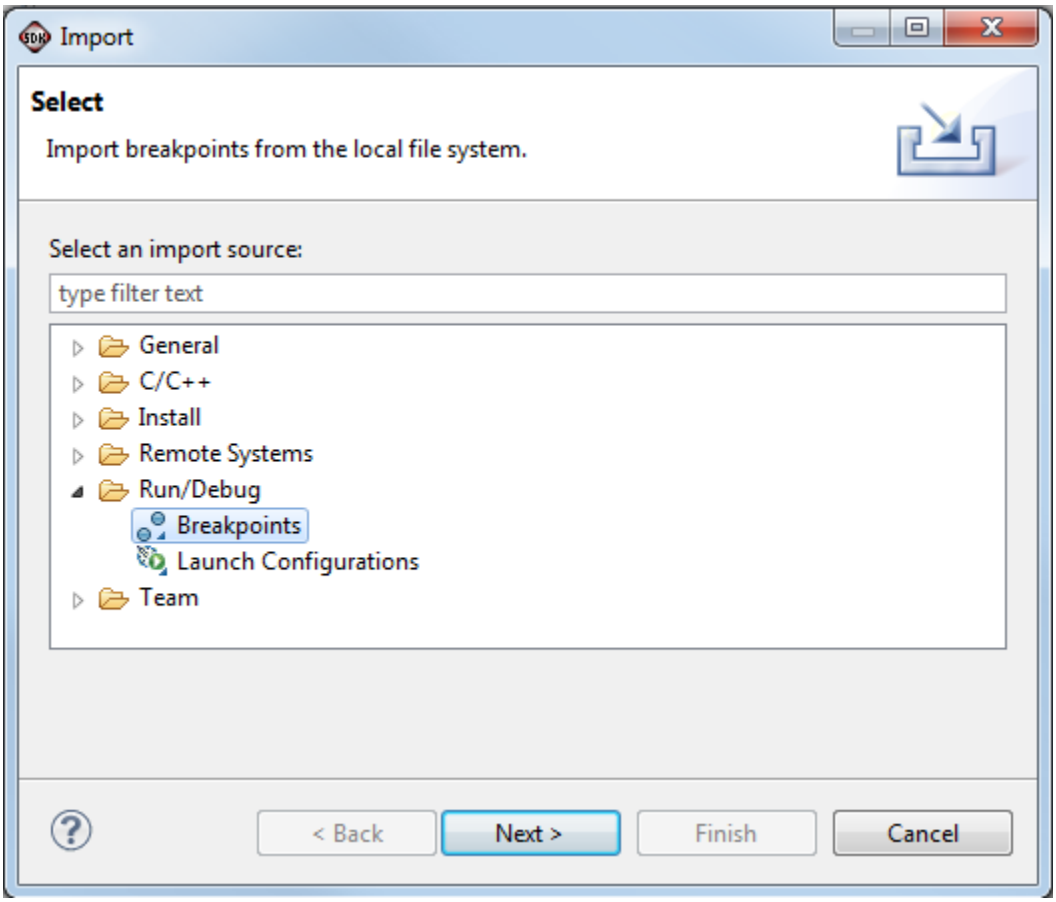

Figure 19 – Import Breakpoints

12. Browse to the `C:\Speedway\ZynqSW\2013_3\` folder, select `Lab08_breakpoint_export.bkpt`, and click **Open**. Click **Finish**.

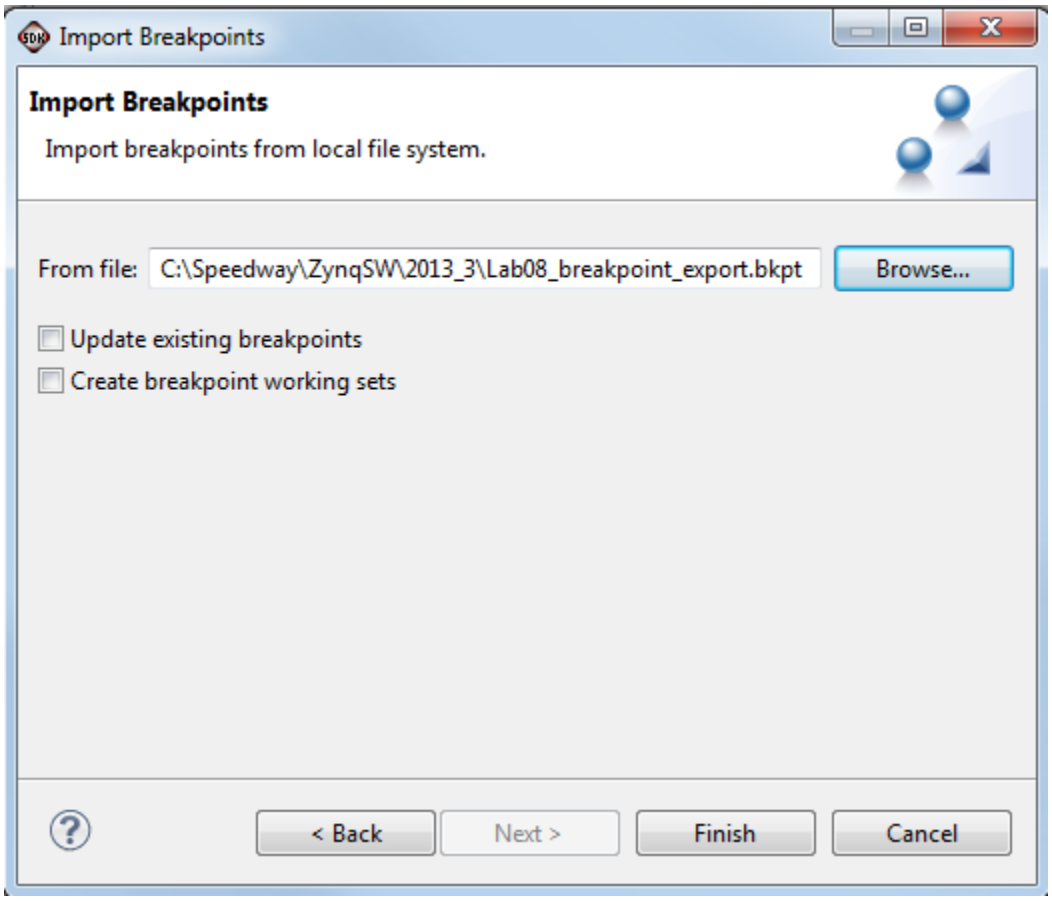

Figure 20 – Import Breakpoints

13. Verify the Breakpoints were imported by selecting **Window → Open Perspective → Debug**. Click on the *Breakpoints* tab. You should see three breakpoints.

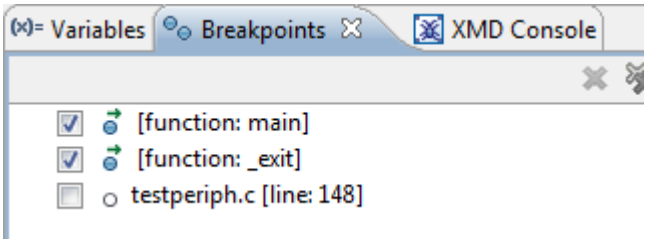

Figure 21 – Run/Debug Configurations Restored

14. Close the Debug perspective by right-clicking on it (in the upper right-hand corner) and selecting **Close**.

Next, we'll restore the project preferences, which would include the repositories (if we had them) as well as our previous Linker Script Generator settings.

15. Since we're restoring a settings file for SDK that gets read at launch, SDK cannot be open during this restore. Close SDK.

16. Browse in Windows Explorer to the `Lab08_repositories_export` folder. Copy and paste the **`com.xilinx.sdk.tools.prefs`** file to the following folder in the **New_Workspace**:

    `.metadata\.plugins\org.eclipse.core.runtime\.settings`

17. If you had received this archive from a colleague and if the archive included repositories, it would also include file **`com.xilinx.sdk.sw.prefs`**. You would need to copy it as well and then edit the **`com.xilinx.sdk.sw.prefs`** file. You would need to find all repository entries in that file, make sure you have copies or access to those repositories, and then update the repository entries to point to your own copies of the repositories.

18. Launch SDK and open the **New_Workspace**.

19. You can verify that these new settings have taken affect by reviewing the **Generate Linker Script** settings for Hello_Zynq. If the sections all point to ps7_ram_0_S_AXI_BASEADDR, then you have successfully restored those preferences. If it points to DDR (the default with no preference file), then it was not successful.

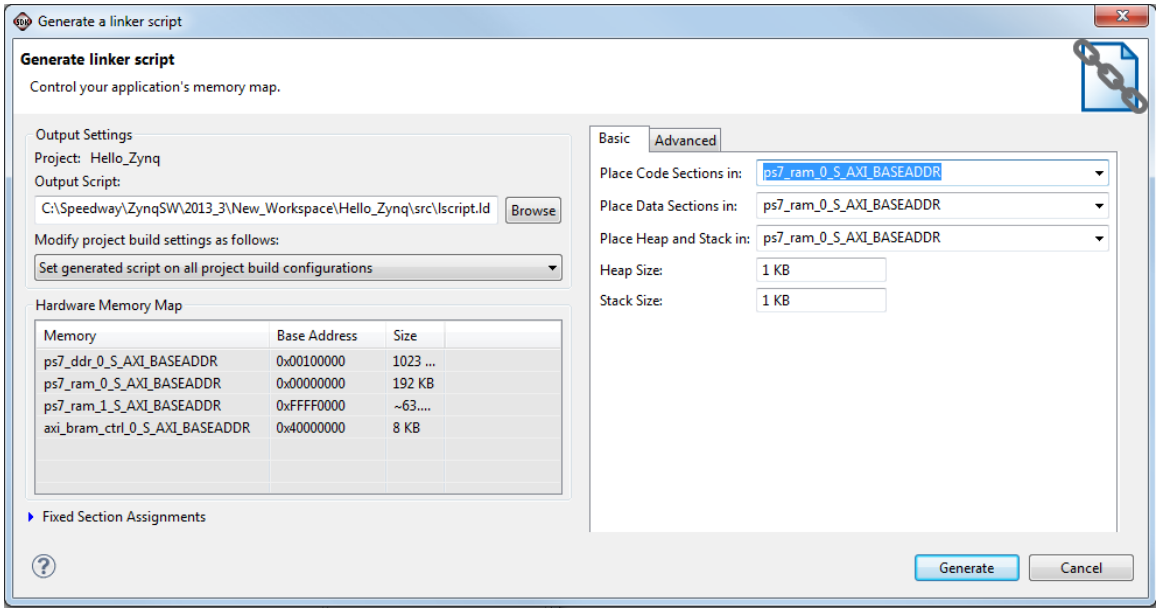

Figure 22 – Preferences Restored as seen in Generate Linker Script

20. You now have a duplicate of your previous workspace. Set ZedBoard back to Cascaded JTAG MODE and connect the JTAG and USB-UART cables. Download the PL Bitstream  . Then download Hello_Zynq to test.

21. 

Figure 23 – PLL Used, JTAG Boot, Cascaded JTAG: MIO[6:2] = 00000

22. If you have more time, go to the Exploring Further section.

23. To avoid confusion, select **File → Switch Workspace** and go back to **SDK_Workspace**.

Exploring Further

If you have more time and would like to investigate more…

- Run the test applications in your regenerated workspace.

This concludes Lab 8.

Revision History

Date	Version	Revision
12 Nov 13	01	Initial release
23 Nov 13	02	Revisions after pilot
01 May 14	03	ZedBoard.org Training Course Release

Answers

Experiment 1

- *What is the advantage of exporting your Workspace items as opposed to simply zipping the workspace?*

If you zip and share your SDK workspace, there is a very good chance that it will not work when opened again. The SDK workspace is full of absolute paths, so unless the recipient unzips the SDK workspace to the exact same location, it won't fully work. It might appear to work initially, but it is likely not going to build properly.

If you export the workspace, it is fully transportable.