Cycle Model Studio
Windows Visual Studio C++ 2013 Integration Application Note

Copyright © 2016 ARM Limited. All rights reserved.

Release Information

The following changes have been made to this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Issue</th>
<th>Confidentiality</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2016</td>
<td>A</td>
<td>Non-Confidential</td>
<td>Restamp release with 9.0.0</td>
</tr>
</tbody>
</table>

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of ARM Limited (“ARM”). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any patents.

This document is provided “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, ARM makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

Non-Confidential Proprietary Notice

This document may include technical inaccuracies or typographical errors.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version shall prevail.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to ARM’s customers is not intended to create or refer to any partnership relationship with any other company. ARM may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any signed written agreement specifically covering this document with ARM, then the signed written agreement prevails over and supersedes the conflicting provisions of these terms.

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited or its affiliates in the EU and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. You must follow the ARM trademark usage guidelines http://www.arm.com/about/trademarks/guidelines/index.php.

Copyright © ARM Limited or its affiliates. All rights reserved.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

In this document, where the term ARM is used to refer to the company it means “ARM or any of its subsidiaries as appropriate”.

© 2016 ARM Limited. All Rights Reserved

Non-Confidential
Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Product Status

The information in this document is final, that is for a developed product.

Web Address

http://www.arm.com
Abstract

This application note describes how to integrate a Cycle Model with the Microsoft Visual Studio C++ 2013 Windows environment. This process includes setting up the Windows environment, compiling the Cycle Model on Linux, and integrating with Visual Studio C++.

Integration Process

The basic integration procedure includes the following steps:

1. Setting up the environment on Windows.
2. Cross compiling the Windows Cycle Model on Linux.
3. Integrating with Visual Studio C++.

Note that this procedure has been verified on Windows 7 (64 bit).

Before you begin, make sure that your environment is set up correctly on Linux, and that you can compile a Cycle Model for use on Linux.

Step 1 – Windows Environment Setup

First, you must have read access to an ARM Cycle Models release directory on Windows. You can mount the release directory on to your PC, copy the required directories/files to your PC, or install Cycle Models Studio software on your Windows machine (see the Cycle Model Studio Installation Guide). If you mount or copy the release data from Linux, make sure the release installation included the Windows data. If you install the Windows release, make sure it matches the release version installed on Linux.

At a minimum, you need the following directories from CARBON_HOME:

- include
- lib
- makefiles
- Win

In addition, you must set the following environment variables appropriately (see the Cycle Model Studio Installation Guide for additional details):

- CARBON_HOME
- ARMLMD_LICENSE_FILE

You can set environment variables in Windows by selecting:

Start > Control Panel > System > Advanced > Environment Variables
The following dialog displays:

Add the following to your Windows PATH environment variable, based on your CARBON_HOME variable.

%CARBON_HOME%\bin;%CARBON_HOME%\Win\lib;
%CARBON_HOME%\Win\lib\win\shared

Depending on your environment, you may need to explicitly expand %CARBON_HOME%.

Note: For Linux users, on Windows, the %VARIABLE_NAME% notation is identical to shell variable accessing via $VARIABLE_NAME.
Step 2 – Creating the Cycle Model

To compile a Cycle Model for Windows on a Linux machine, specify the .lib file extension for the Cycle Model name (in the format lib<design>.lib) using the –o Cycle Model compiler option. Note that you need to replace <design> with the name of the design.

- Using Cycle Model Studio GUI, specify the file name in the Basic Options section of the Compiler Properties.
- Using the Cycle Model compiler (cbuild) command line, enter:

  cbuild -o lib<design>.lib ...

  instead of -o lib<design>.a, which generates a Cycle Model for Linux.

The Cycle Model Studio or Cycle Model compiler outputs a number of files, including the following, which are required for compiling/running the executable on Windows:

- lib<design>.lib – Windows static object library
- lib<design>.h – include file for interfacing to the Cycle Model

All files listed above must be copied to (or be accessible to) your Windows environment.

By default, DB files are embedded in the “libdesign.lib” file. If embedded, they are NOT required. However, if the Cycle Model is compiled with –noFullDB, then one of DB files is required. Additionally, the Visual Studio Project Property called “Working Directory” must be set to the directory where these files reside.

- lib<design>.symtab.db – Cycle Model database with information about all internal signals
- lib<design>.io.db – Cycle Model database with only a subset of the signals—only the IOs and additional signals marked as observable or depositable

Step 3 – Visual C++ Integration

To integrate the Cycle Model, you must either create a new Visual C++ project or make changes to your existing project. The following are the steps to open and set up a new project:

   - Select File > New Project and create a project with the following settings:
     o Project Types: Visual C++ Win32
     o Name: <Enter Project Name>
     o Location: <Enter Project Dir Path>
     o Template: Win32 Console Application
In the “Win32 Application Wizard”, click on **Next** and make sure:
  
  - the ‘Console Application’ box is selected under Application Type, and
  - the ‘Empty project’ box is selected under ‘Additional options’

- **Click Finish.**

2. Add C++ files to the project. This can be accomplished by dragging and dropping the files into the Visual Studio “Solution Explorer.” For the Cycle Model, you must add the header file `lib<design>.h`. For the test bench, you must add your test bench C/C++ file.
3. Open the Project Property Page (Project > Properties), and set the following properties:
   - Set “Configuration:” to “All Configurations”.
   - From the “C/C++” tab, select the “General” properties and set “Additional Include Directories” to:
     "$(CARBON_HOME)\include"; <directory containing libdesign.h>
   - From the Linker tab,
     i. Select the “Input” properties and in the “Additional Dependencies” box add:
        lib<design>.lib libcarbon5.lib libgcc.lib
     ii. Select the “General” properties and in the “Additional Library Directories” box add:
        "$(CARBON_HOME)\Win\lib";
        "$(CARBON_HOME)\Win\lib\wxshared";
        "$(CARBON_HOME)\Win\bin\lib\gcc\x86_64-w64-mingw32\4.7.2";
        <directory containing the Cycle Model, the libdesign.lib file>
   - Click Apply.
   - Open “Configuration Manager…”
     o Add a new platform by selecting <New...> from the pull-down menu:
4. Build the Project.
   - Set the configuration to either “Debug” or “Release”.
   - Select Build > Build Solution.

5. Run the Simulation.
- Set the configuration to either “Debug” or “Release”.
- Select **Debug -> Start**.

6. **Debug Simulation Startup Issues.**

If the simulation reports an exception error at the beginning, there may be a problem finding the ARM license or finding the database file. To determine the cause, perform the following steps:

- Set a breakpoint on the exit function:
  - Open the New Breakpoint window by selecting **Debug > New Breakpoint** (or **Ctrl+B**).
  - In the Function field, type “exit”.
  - Click **OK**.

- Start the simulation by selecting **Debug > Start** (or **F5**).
- Examine the error message(s) displayed in the Command Prompt window that was open when the simulation started.

If it is a license issue, you may see the following error message:

```
```

In this case, examine the ARMLMD_LICENSE_FILE environment variable to verify that it is set to a valid license.