APU-2 Tool User Guide

UG-10301-01-08

Revision History

Version	Details of Change	Author	Date
01	Initial Release	W. Hulme	Mar. 31, 2013
02	Update for APU2 Tools R2.1	W. Hulme	Apr. 17, 2013
03	Update for APU2 Tools R2.2	W. Hulme	July 3, 2013
04	Section 2.3:Added more information in for installation steps. Figure 3-2: updated to use APEX-642 naming. Section 4.2: Corrected install path naming	W. Hulme	Sept. 20, 2013
05	Update for APU2 Tools 13R1.6	W. Hulme	Mar. 28, 2014
06	Update to reflect Synopsys installation flow	AO	Jan 15, 2015
07	Update to reflect Synopsys installation updates on package APU-2 IPP J-2014.09-SP1-1	AO	Feb 3, 2015
08	Updated formatting	AO	Feb 17, 2015

Table of Contents

Table of Figures

1 Introduction

This document is intended to provide a general overview of the APU-2 tool set, including the installation and various tools provided with the development environment.

1.1 Terminology

Term	Definition
Kernel	Algorithmic/Functional unit that executes on APU or other device.
Buffer	A block of scalar or vector data memory.
SIMD processor	A processor architecture which exploits multiple data streams against a single instruction stream to perform operations which may be naturally parallelized.

1.2 Acronyms

Acronym	Definition
ACP	Array Control Processor
APU	Array Processing Unit
CMEM	Computational Memory
CPU	Central Processing Unit
CU	Computational Unit
DMA	Direct Memory Access
DMEM	Data Memory
IDE	Integrated Development Environment
IMEM	Instruction Memory
ISS	Instruction Set Simulator
MIMD	Multiple Instructions Multiple Data
OpenCV	Open Source Computer Vision
RISC	Reduced Instruction Set Computer
SIMD	Single Instruction Multiple Data

1.3 References

The following Synopsys IP Programmer for APU2 Documents are referenced or included in this document:

- **Install Manual**: “install-manual-<version>.pdf”
- **Chess User Manual**: “chess_user-manual-<version>.pdf”

2 Overview

The APU-2 tools are used to prototype, benchmark and develop kernel code for the CogniVue APU-2 hardware. The Synopsys ChessDE tool simulates the APU-2 instruction set, and can be run on either Windows or Linux-based PCs.

2.1 Prerequisites

See Section 2 of the **Install Manual**.

2.2 Installation Instructions

Follow the instructions in section 3 of the **Install Manual**

Synopsys software is downloaded, installed and maintained via the Synopsys SolvNet Online Support portal.

2.2.1 Additional notes for Install Manual Section 3.3.1 "Setting up the license server "

Typically the license file needs to be modified to reflect the server host name, and port number used to broadcast the license. The VENDOR daemon line needs to be modified to point to the location of snpslmd.exe. This is where you installed the SCL package.

Windows example:

C:\Your_*SCL_InstallDir*\SCL\11.9\win32\bin\snpslmd

License file would look like:

```
>>>>
#-----------------------------------------------------------------------
-----------------------------------------------------------------------
--------------------
# Synopsys License file created by CVDKIF 3.16.2 on January 05, 2015 at
15:00:42 for daemon snpslmd
SERVER CGV_lic_server 60EB69ED0D88 27020
VENDOR snpslmd C:\Your_SCL_InstallDir\SCL\11.9\win32\bin\snpslmd
<<<<
```

NB: It is preferable that the installation, and all associated source files, exist in directories **which do not have spaces in the path**. This is for both command line and IDE users.

You can configure the license server as a Windows service so you do not need to restart it after every reboot.

On Windows to start lmtools (Start → All Programs → Synopsys →SCL<version> → lmtools).

Under "Service/License File" Tab

- select "Configuration using Services"

Under "Config Services" Tab,

- define a Service Name
- define path to the lmgrd.exe file (for example: *C:\Synopsys\SCL***<version>***win32\bin\lmgrd.exe*)
- path to the license file (for example *C:\Synopsys\target.lic*)

2.3 Post-Install Configuration

The IP Programmer for APU2 distribution only contains the bare processor model. Once the IP Programmer for APU2 has been installed, a number software archives must be generated. For each archive, there is a ChessDE project file. You can build the software archives by opening the project files in ChessDE, or on the command line with the command 'chessmk <prx file>'.

Batch and shell files are provided for Windows and Linux user for this post-install build. To develop any APU kernel, you need to build the APU-2 model first on your local machine, as follows:

2.3.1 Windows

1. Go to **YourInstallDIR**\ipp-APU2**<version>**\win32\designs\APU2
2. Copy the provided “apu2_ipp_compile_VERSION.bat” file into the folder in step 1 and click on it to build the model and the library.
3. When the model has been successfully built, create an environment variable on your machine: (Steps may vary depending on OS version)

   i. Right-click the "My Computer" icon (“This PC” in Win8)
   ii. Choose "Properties" from the context menu
   iii. Click the "Advanced" tab
   iv. Click the "Environment Variables" button
   v. Under the "User variables" section, click new
   vi. Variable: APU2_DIR
   Value: **YourInstallDIR**\ipp-APU2**<version>**\win32\designs\APU2\lib\base
   vii. Click OK

2.3.2 Linux

1. Go to **YourInstallDIR**/ipp-apu2/**<version>**/linux

   *$ cd **YourInstallDIR**/ipp-apu2/**<version>**/linux*
2. Source the environment setup script

   *$ source **YourInstallDIR** /ipp_apu2/<version>/linux/chess_env_LNa64.sh*
3. Set $GCC_HOME environment variable to point to g++ compiler (ex. */ usr/bin/g++)*

   *$ export GCC_HOME= **path_to_g++_binary**/g++*
4. Copy provided “apu2_compile.sh” to **YourInstallDIR**/ipp-apu2/<version>/linux/designs/APU2
5. Source the file in the previous step

   $ source apu2_compile.sh

3 APU Architecture

3.1 Overview

The APU has both Scalar and SIMD processing capability. Scalar processing is performed in the Array Control Processor (ACP) unit, and Vector processing is performed in the Vector Processor Unit. The ACP is a 32-bit RISC based processor implementation. The Vector Unit is made up of thirty-two 16-bit computational units (CU's) each of which is attached to a x16bit CMEM memory space.

Both processing units operate from the same instruction memory (IMEM) either serially or concurrently depending on the instruction word. Each processor operates in an independent data memory space: the ACP in DMEM, and the Vector unit in CMEM.

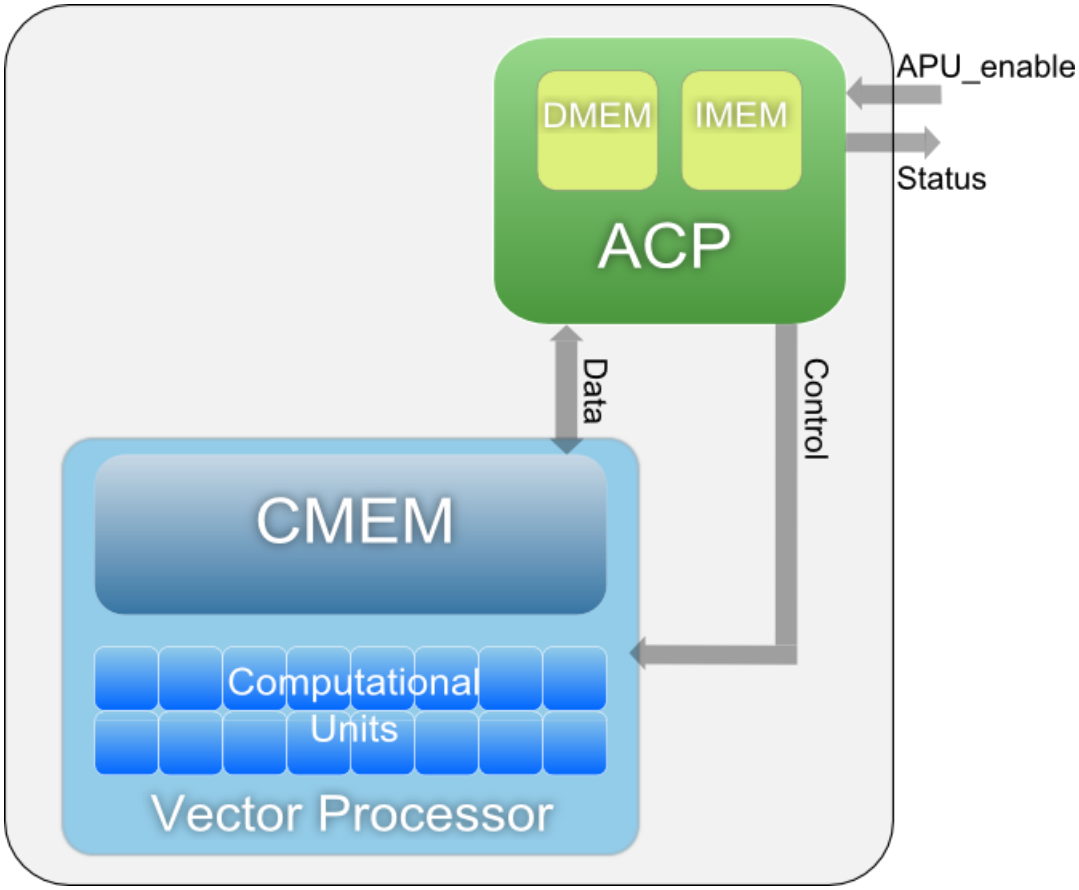

Figure 3-1 – Processor Architecture

3.1 Features

APEX642 is comprised of two second generation Array Processing Unit (APU-2) cores configurable as single SIMD engine with 64 16bit Computational Units (CU), or configurable as two core engines with 32 16bit CUs each.

The vector units are shared between multiple APUs. This way, the vector size can be configured with the appropriate control and data connectivity.

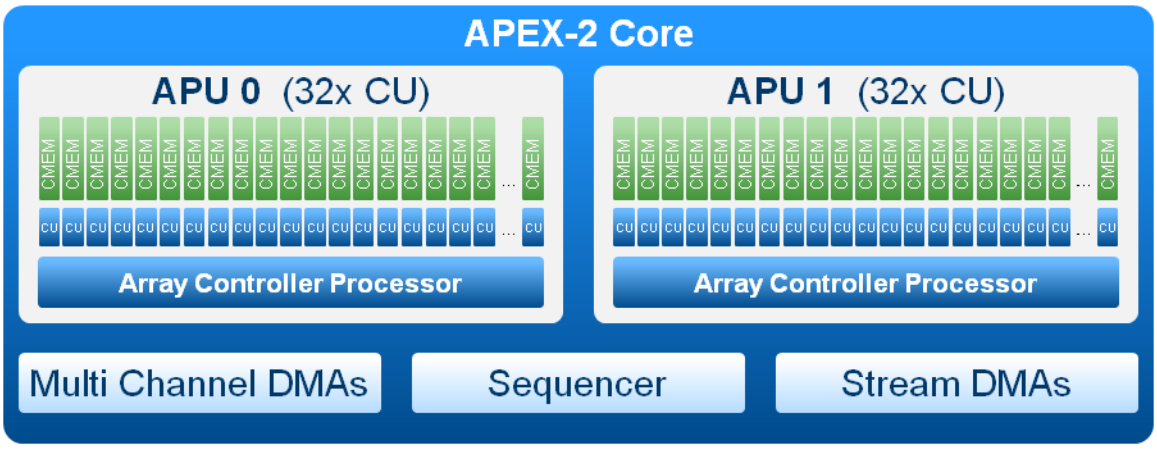

Figure 3-2 – APEX642 Multi-APU Architecture

Features of the APU include:

- Instruction Memory (IMEM)
- Scalar Data Memory (DMEM)
- CU Data Memory (CMEM)
- Scalar unit
- Vector unit

4 APU-2 Chess Development Environment

4.1 Overview

The APU-2 Development Environment is the development interface used to compile, run, debug and profile applications with the APU2 Instruction Set Simulator. The development environment may be used in either the Windows or Linux environments. See section 2.1 for more details.

The Chess Instruction Set Simulator runs a cycle accurate simulation of the APU2 processing architecture. APU-2 kernels which are developed and tested in this environment can be executed on the APU-2 hardware.

4.2 Using the IDE

The Chess Development Environment may be run under:

4.2.1 Windows

Select

Start → All Programs → Synopsys → IP Programmer for APU2 → <version>→ ChessDE

4.2.2 Linux

Executing the following (shell setup is only needed if it is not part of your environment setup by default)

*$ source **YourInstallDIR** /ipp_apu2/<version>/linux/chess_env_LNa64.sh*

$ chessde &

For all other information on the APU-2 Chess Development environment you can refer to the **Chess User Manual** and other manuals in

YourInstallDIR\ipp-APU2**<version>**\win32\doc\manuals directory.