



# **APU2 Customer Engagement Package Bug Report Template**

**UG-10147-00-01**

## **Copyright**

Copyright © 2013 CogniVue Corporation ("CogniVue") All rights reserved.

This document contains information which is proprietary to CogniVue and may be used for non-commercial purposes within your organization in support of CogniVue's products. No other use or transmission of all or any part of this document is permitted without written permission from CogniVue, and must include all copyright and other proprietary notices. Use or transmission of all or any part of this document in violation of any applicable Canadian or other legislation is hereby expressly prohibited.

User obtains no rights in the information or in any product, process, technology or trademark which it includes or describes, and is expressly prohibited from modifying the information or creating derivative works without the express written consent of CogniVue.

## **Disclaimer**

CogniVue assumes no responsibility for the accuracy or completeness of the information presented which is subject to change without notice. In no event will CogniVue be liable for any direct, indirect, special, incidental or consequential damages, including lost profits, lost business or lost data, resulting from the use of or reliance upon the information, whether or not CogniVue has been advised of the possibility of such damages.

Mention of non-CogniVue products or services is for information purposes only and constitutes neither an endorsement nor a recommendation.

## Revision History

Version	Details of Change	Author	Date
01	Initial release		Aug 27, 2013

# Table of Contents

1	General Document Information .....	6
1.1	Acronyms .....	6
1.2	References.....	6
2	APU2 Tools Bug Report .....	7
2.1	Please ensure the following steps before submitting the bug report: .....	7
2.2	Bug report template:.....	7

## Table of Figures

No table of figures entries found.

# Table of Tables

Table 1-1: Acronyms .....	6
---------------------------	---

# 1 General Document Information

This document presents a bug report template for reporting APU tool and its sw package related bugs to CogniVue developers.

It is essential that the report describes the issue effectively so that time and effort is not unnecessarily wasted in trying to understand and reproduce the issue. This template attempts on collecting all the relevant information for this purpose.

## 1.1 Acronyms

A section on Acronyms may be included as shown below.

Acronym	Definition
APU	Array Processing Unit
ISS	Instruction Set Simulator

**Table 1-1: Acronyms**

## 1.2 References

- [1] "APU-2\_Engagement\_Package\_Release\_Notes", CogniVue document UG-10301-03
- [2] "APU-2\_Tool\_User\_Guide ", CogniVue document UG-10301-01

## 2 APU2 Tools Bug Report

### 2.1 Please ensure the following steps before submitting the bug report:

1. If Windows installation, post-installation is also required, refer to Section 2.3 of [2];
2. Same issue has been observed on more than one machine;
3. Check the section of “known issues” in [1];

### 2.2 Bug report template:

1. Reporter and email address;
2. Issue severity;
3. Test environment:
  - a. PC environment:
    - i. OS (Windows XP, Windows 7 or Linux, 32 or 64-bit, service pack number);
    - ii. C++ compiler: g++ or visual studio and their versions;
  - b. APU tool version;
  - c. APU sw library version if available;
  - d. Test platform where the issue appears: ISS or FPGA;
4. Issue type:
  - a. Compiler error/warning;
  - b. APU instruction/intrinsic bug;
  - c. Tool IDE;
5. Issue description:
  - a. Repeatability and steps to reproduce the issue ;
  - b. Descriptions of expected behaviors;
  - c. Compiler error/warning messages, if any;
6. Issue test case:

Test cases are required for issue type a & b above;

Accompanying test\_\*\*\*.cpp that triggers a “failure” case should include the following:

```

////////////////////////////////////
/// Bug description:
/// Instruction
/// Data type
/// etc.
////////////////////////////////////

```

```
////////////////////////////////////
/// APU implementation
////////////////////////////////////

int apu_***()
{
}

////////////////////////////////////
/// Reference implementation if available
/// otherwise, load the pre-calculated golden result into reference buffer
////////////////////////////////////

int ref_***()
{
}

////////////////////////////////////
/// Test main
////////////////////////////////////

int test_apu_***()
{
    int rval = 0;
    // Call apu kernel
    apu_***();

    // Call reference implementation;
    // Or load the pre-calculated golden result into reference buffer
    ref_***();

    // Validation step is required
    // Compare results and print pass/fail
    rval = compare_results (apu_dst, ref_dst);
    if (rval == 0)
    {
    }
}
```