Test Plan

RISC-V Secure Hardware Video Output

Ahmad Alothaimin, Hector Soto, Jack Chen,
Ross Wegter, Ryan Nand
March 22, 2021
Rev. 0.1.0

Table of Contents

Table of Contents **2**

1.0 Introduction **3**
- 1.1 Purpose 3
- 1.2 Types of testing 3
- 1.3 Conduct of Systems Tests 3
- 1.4 Recording of Results, Witnessing, and Authorities 3
- 1.5 Limitations 4

2.0 References **4**
- 2.1 Design Documentation 4
- 2.2 Other Documents 4

3.0 Objectives and Resources **4**
- 3.1 Unit/Module Test 4
  - 3.1.1 Display IP Block Test 5
  - 3.1.2 12-bit PMOD to DVI Board Test 5
  - 3.1.3 Linux Driver Test 5
- 3.2 Integration Test 6
  - 3.2.1 Test Attachment of Display IP Block to Main GFE Vivado Project 6
  - 3.2.2 Test Driver and Display IP Block Interoperability 7
- 3.3 Functional Test 7
  - 3.3.1 Display Image Test 7
  - 3.3.2 Display Altering Image Test 8
- 3.4 Acceptance Test 8

4.0 Personnel **8**
- 4.1 Roles 8

5.0 Detailed Test Cases **9**
- 5.1 Display IP Block Test 9
- 5.2 DVI PMOD Functional Test 10

1.0 Introduction

1.1 Purpose

This document describes the plan for testing the RISC-V secure hardware video output project. This test plan documentation will illustrate and explain the guidelines for testing, as well as describe the equipment, personnel, or other resources needed to perform the tests. In the end, this document should provide an estimate of the test efforts and list the deliverable elements of the test activities.

1.2 Types of testing

- Unit/Module Testing: Consists of a family of test cases establishing that the individual module performs some functionality correctly and according to specifications.

- Integration Testing: Checks to see that multiple modules within the system operate correctly together. Identifies the different paths in the system and ensures that each interface has been realized. The test cases are traceable to both high and low level system designs and deciphers if the system meets the performance specifications.

- Functional Testing: Consists of test cases to determine if the module within the system or the system itself performs the most basic functionality, excluding parametric variables.

- Acceptance Testing: Consists of test cases that determines if the system is meeting the clients requirements. These test cases must adhere to the functional and performance specifications.

1.3 Conduct of Systems Tests

RISC-V secure hardware video output requires thorough testing for all of the test cases listed above. The test cases must be conducted as stated by the procedure for each respective test. All steps must be followed and completed in appropriate order according to the procedure of the test cases. If there are obstacles within the specific test case preventing the completion of the testing, this obstacle must be documented with any additional information explaining the situation. These obstacles must then be reviewed by appropriate personnel determining the next steps.

1.4 Recording of Results, Witnessing, and Authorities

The testing results must be abstract, unambiguous, traceable, and verifiable. All results must be documented and verified by the respective tester within the test case. Any failures or errors within the test cases illustrated above will be reviewed by the proper personnel and the system

design will then be investigated for revision. This may lead to further testing which will then be reflected in the test cases.

1.5 Limitations

Remote work on this project has proved challenging and has some associated limitations. The majority of output measurements will be qualitative and may come at the expense of informally acquiring results by means of questionnaires to a person local to the development board. For example, a potential setup may have a camera pointed towards a monitor connected to the remote development board to visually inspect video output. However, this camera may send video capture in black and white and thus limit our measurements of color. In turn, we would rely on other means for obtaining test results such as a report from a person local to the remote development board.

2.0 References

2.1 Design Documentation

- Product Design Specification Rev. 0.2.0
- GFE Rel4.0 System Description Aug 9, 2019

2.2 Other Documents

- GFE Aug 9, 2019

3.0 Objectives and Resources

In the following section, the tests are relisted with their objectives and the bare minimum resources required to perform each of the tests.

3.1 Unit/Module Test

All unit testing requires the following personnel and equipment. These tests are the ensure each individual component works properly.

Personnel: A unit tester with an understanding equivalent to an Associates' degree in Computer Engineer/Science.

Equipment:

- An PC or Virtual Machine running Debian 10 "Buster".
- Installation of Vivado 2019.1 with a valid license.
- A clone of the `ssith/gfe` repository on the `feature/video-output` branch.
    - All files relevant to video output are located in [DIRECTORY NAME]/

3.1.1 Display IP Block Test

Verify that the behavior of the IP block generates a valid VGA output signal.

Test equipment:

- Local FPGA board and provided cable for programming.
    - Display IP block was developed on an Arty A7-35T.
- Constraints file (.xdc) for local FPGA board.
- The `[DIRECTORY NAME]/display_test/display_test.xpr` vivado project.
- VGA cable and a monitor that accepts VGA.

Objectives:

- Determine if the IP block sends out a valid VGA signal
- Examine if test pattern is displayed out on a monitor

3.1.2 12-bit PMOD to DVI Board Test

Verify that that 12-bit PMOD to DVI board accepts a valid VGA signal and generates a valid DVI-D or HDMI signal.

Test equipment:

- Local FPGA board and provided cable for programming
    - Display IP block was developed on an Arty A7-35T
- Constraints file (.xdc) for local FPGA board.

- The `[DIRECTORY NAME]/display_test/display_test.xpr` vivado project.
- Icebreaker 12-bit PMOD to DVI board.
- HDMI cable and a monitor that accepts HDMI.

Objectives:
- Determine that the 12-bit to PMOD to DVI board properly converts an HDMI signal.
- Examine test pattern displayed on monitor and spot differences, if any, from 3.1.1 Display IP Block Test

3.1.3 Linux Driver Test

Verify that the driver operates correctly by having it manipulate a DMA framebuffer and memory mapped IO, even if there's no physical hardware to display said framebuffer or receive IO signals.

Test equipment:
- Virtual hardware Display IP block (software that interacts with the driver as if it were the hardware of the Display IP block).

Objectives:
- Verify driver has no errors when mounted or unmounted from the kernel.
- Verify that the driver can manipulate the framebuffer when given commands from user space.
- Verify the driver can send and receive signals via its allocated IO ports.

3.2 Integration Test

All the following integration tests require the following personnel and equipment.

Personnel: An integration tester with an understanding equivalent to an Associates' degree in Computer Engineer/Science. It is recommended that this person also has a fine understanding of Vivado.

Equipment:
- An PC or Virtual Machine running Debian 10 "Buster".
- Installation of Vivado 2019.1 with a valid license.
- A clone of the `ssith/gfe` repository on the `feature/video-output` branch.
    - All files relevant to video output are located in [DIRECTORY NAME]/
- Remote access to the VCU118 FPGA development board.

3.2.1 Test Attachment of Display IP Block to Main GFE Vivado Project

Integrate the Display IP Block unit onto the main GFE project, fix errors that will occur during bitstream generation.

Test Equipment:

- Vivado project located in `[DIRECTORY NAME]/display_test/[INTEGRATE VIDEO BLOCK NAME].xpr`

Objectives:

- Verify that the bitstream of the GFE Vivado project with the Display IP block is generated without errors.

3.2.2 Test Driver and Display IP Block Interoperability

Integrate driver into current Linux build so that it drives the Display IP block. Fix any IO errors that may occur either due to timing or mismatched addresses. The Display IP block should have no issue with reading from the framebuffer so long as IO is operating correctly.

Test Equipment:

- Vivado project located in `[DIRECTORY NAME]/display_test/[DRIVER and DISPLAY BLOCK NAME].xpr`
- Driver for video output.
- Userspace program to draw something on the screen.
- Local FPGA (optional).

Objectives:

- Verify that the bitstream generated without errors.
- Examine the behavior of the driver with the userspace program. Writing and running new userspace programs may be done to verify integration of the driver and display IP.

3.3 Functional Test

All the following functional tests require the following personnel and equipment.

Personnel: An functional tester with an understanding equivalent to an Associates' degree in Computer Engineer/Science. It is recommended that this person also has a good understanding of Vivado.

Equipment:

- An PC or Virtual Machine running Debian 10 "Buster".
- Installation of Vivado 2019.1 with a valid license.
- A clone of the `ssith/gfe` repository on the `feature/video-output` branch.
    - All files relevant to video output are located in [DIRECTORY NAME]/
- Remote access to the VCU118 FPGA development board.

3.3.1 Display Image Test

Upload the bitstream that displays an image onto the VCU118.

Test Equipment:

- Vivado project located in `[DIRECTORY NAME]/display_test/[DISPLAY IMAGE NAME].xpr`

Objectives:

- Verify that the bitstream generated without errors.

3.3.2 Display Altering Image Test

Have a user space program utilize the driver to have changing images on the screen. This can be as simple as having a square bounce around the screen or fully displaying a terminal.

Test Equipment:

- Vivado project located in `[DIRECTORY NAME]/display_test/[DRIVER and DISPLAY BLOCK NAME].xpr`
- Driver for video output.
- Userspace program to draw something on the screen.

Objectives:

- Verify that images drawn on the screen can change in real time.

3.4 Acceptance Test

No formal acceptance tests are listed, unless desired by all stakeholders. Examples of such tests would be examining the power or layout usage of our display block, or listing the memory address regions that our DMA requires.

4.0 Personnel

4.1 Roles

Human Resources		
Role	**Minimum Resources Recommended**	**Responsibilities/Comments**
Test Manager	1 Person	Oversees tests and manages testing resources
Unit Tester	1 Person with understanding equivalent to an Associates' degree in Computer Engineer/Science	Will ensure individual units and blocks of design are functional.
Integration Tester	1 Person with understanding equivalent to an Associates' degree in Computer Engineer/Science.	Will upload bitstream to remote FPGA and test system for proper integration and/or functionality.

5.0 Detailed Test Cases

5.1 Display IP Block Test

Test Writer: Jack Chen						
Test Case Name:		Display IP Block Test		**Test ID #:**		1.1
Description:		Verify that the behavior of the IP block generates a valid VGA output signal.		**Type:**		
Tester Information						
Name of Tester:				**Date:**		
Hardware Ver:		VCU 118 rev 2 Local FPGA (Nexys A7 described)		**Time:**		
Setup:		FPGA powered on, FPGA's VGA output connected to a powered monitor with VGA cable. Selected Vivado project in `gfe/display_test` has the correct target for this chosen FPGA.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Check target part on Vivado	Target part matches FPGA model				
2	Generate bitstream	Bitstream generates with no errors				
3	Active high asserted	Monitor shows that a signal is detected				
4	Moving lines test pattern	Coloured lines scan vertically and horizontally without distortion on the monitor				
5	Color gradient test pattern	A gradient of available colors are displayed as columns				
6	Bouncing square test pattern	A white square moves across the monitor, bouncing on the edges				
Overall Test Results:						

5.2 DVI PMOD Functional Test

Test Writer: Ross Wegter

Test Case Name:	DVI PMOD Functions	**Test ID #:**	1.2
Description:	Measuring qualitative output of DVI PMOD board by variable changes to input vs expected results.	**Type:**	Automated Black Box Unit Test

Tester Information

Name of Tester:		**Date:**	
Hardware Ver:	VCU 118 rev 2 12bit DVI PMOD V1.1a	**Time:**	
Setup:	VCU118 powered on and physically/remotely accessible with 12bit DVI PMOD board connected via PMOD connectors. 12bit DVI PMOD board needs to be connected to a powered monitor via HDMI cable. Test program will be programmed in FPGA fabric as a modified Test Pattern Generator module.		

	Action	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Generate Bitstream of GFE with Test.	Bitstream generated in Vivado successfully.				
2	Program VCU118	pyprogram_fpga.py programs generated bitstream onto VCU118 without error.				
3	Set active signal high for 5s then low for 5s.	Monitor will detect signal and display nothing for five seconds then monitor will detect no signal for 5s.				
4	Output Red, then Green, and then Blue in sequence across all pixels.	A full screen of red for a period of 3s, followed by a full screen of Green for 3s, and then a full screen of Blue for 3s.				
5	Output Red, Green, and Blue concurrently(White)	A full screen of White for 3s.				
6	Output RGB Divisions from 16 to 0 by 2 across screen sequentially.	A full screen of white for a period of 3s followed by half RGB values for 3s continuing this sequence until RGBs of 2 after which RGB values of 0 (Black) will follow for 3s.				

7	Display test pattern and set timing out of Horizontal Sync	Test pattern is horizontally distorted.				
8	Display test pattern and set timing out of Vertical Sync	Test pattern is distorted with incorrect vertical alignment.				
Overall Test Results:						