PCIe Test Platform Application Guide

(V1.0)

(22.11.2021)

Shenzhen Pango Microsystems Co., Ltd.

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.0	22.11.2021	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
PCIe	Peripheral component interconnect
DMA	Direct Memory Access
Bar	Base Address Registers

Table of Contents

Figures

Chapter 1 Overview

1.1 Introduction

The FPGA device, supporting PCIe interface, is responsible for DMA module and PC communication for PCIe interface data interaction, with the driver code required. As a result, it is necessary to develop PCIe driver in the Linux environment. And it is also needed to develop a human-computer interaction interface, call the driver code and PCIe board to perform DMA read and write tests, PIO read and write tests, configuration space read and write tests, Tandem loading tests, as well as PCIe DMA performance tests.

The Windows-based PCIe test tool driver is available for a fee (30-day trial period), by developing PCIe driver in the Linux environment, it can save costs dramatically. Self-development can also provide customers with a complete set of PCIe DMA verification test solutions.

Currently, PCIe test methods are relatively simple to be implemented, requiring the use of PCI Scope tool to check the link status. And self-developed DMA test tool can be used by means of manual step-by-step operation, so as to make a manual comparison to confirm whether the verification results are normal, therefore, the test efficiency is very low, investing a lot of labour costs.

Developing an automated test platform not only enhances the test efficiency and reduces labour costs, but also benefits product demonstrations and promotional applications.

Chapter 2 Test Interface Function Description

GTK (GIMP Toolkit) is used as the test interface, which is an open-source graphical interface toolkit that is widely used in the Linux environment. The following section explains the components of the test interface in detail. The main interface is shown in the following figure:

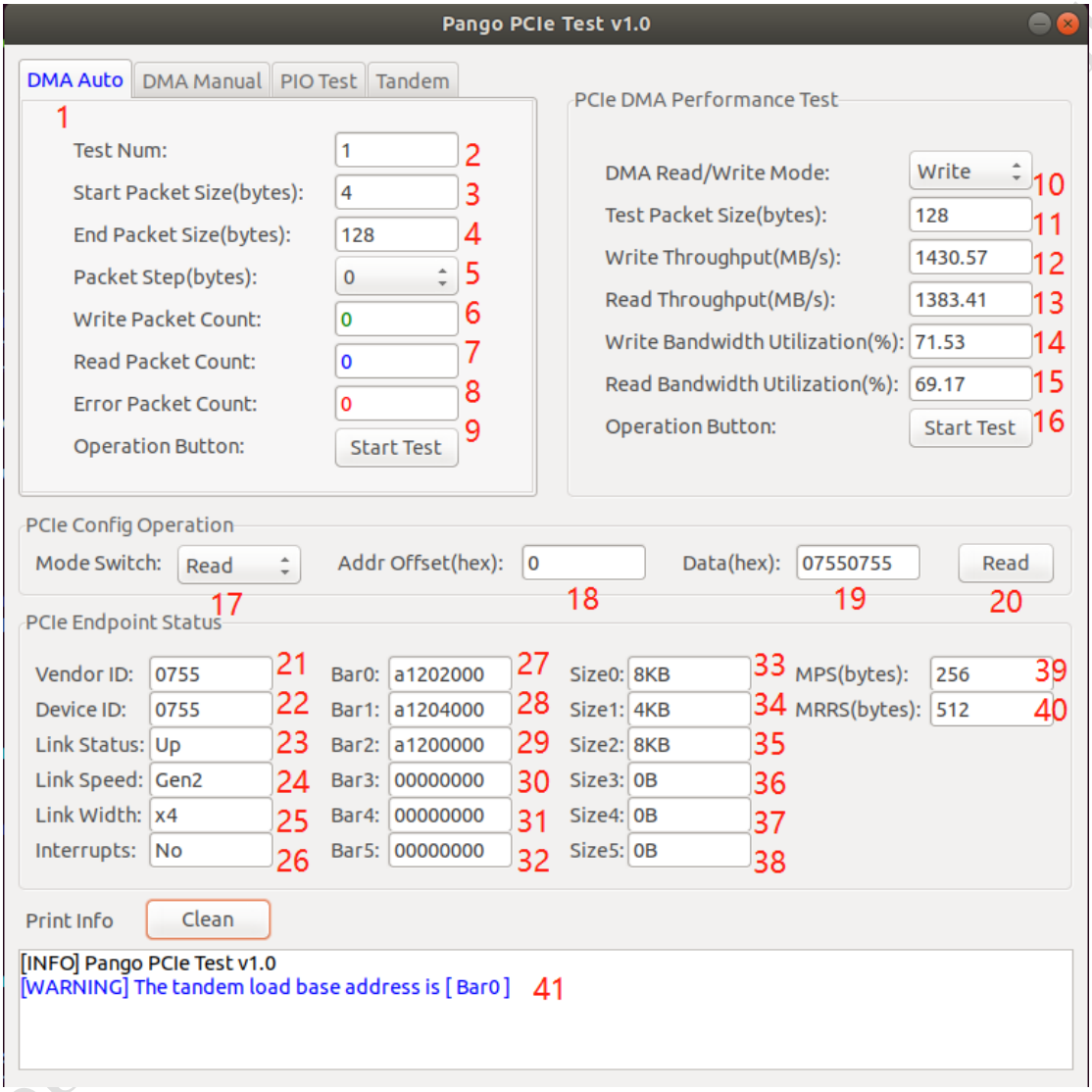


Figure 2-1 Main Test Interface

2.1 DMA Auto Interface

No. 1-9 in the main interface represent the DMA Auto test, as shown in the interface below:

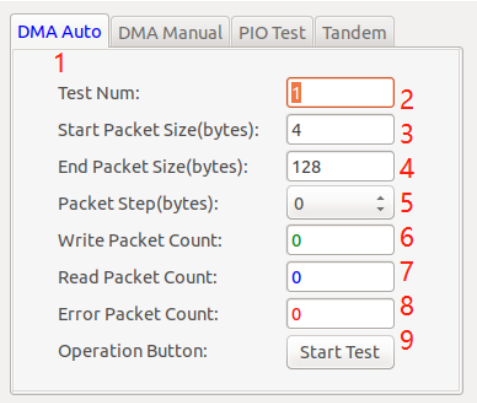


Figure 2-2 DMA Auto Interface

The meanings represented by the numbers are as follows:

No. 1-DMA Auto: DMA automatic test selection interface.

No. 2-Test Num: DMA automatic test count. When the test count is 0, it represents a loop test until the test is manually stopped.

No. 3-Start Packet Size (bytes): The size of the start packet, i.e.: the data length for the first DMA read/write test, in bytes (minimum 4 bytes).

No. 4-End Packet Size (bytes): The size of the end packet, i.e.: when the packet step value is not equal to 0, the maximum length of the data packet accumulated by the step value (not exceeding 4096 bytes).

No. 5-Packet Step (bytes): Packet step value selection, i.e.: After one DMA read/write operation is completed, the length of the next read/write data packet will be the length of the previous data packet plus the step value, and the length of the final data packet will not exceed End Packet Size. The optional step values are: 0 ,4 ,8 ,16 ,32 ,64 ,128 ,256 ,512 ,1024, as shown in the figure below:

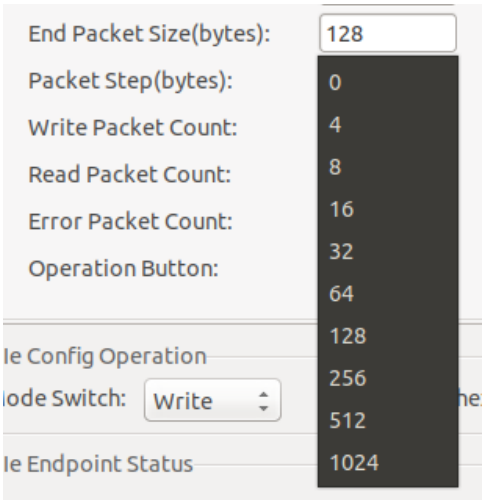


Figure 2-3 Packet Step Selection

No. 6-Write Packet Count: DMA write count.

No. 7-Read Packet Count: DMA read count.

No. 8-Error Packet Count: DMA read/write error count.

No. 9-Operation Button: DMA Auto Test Start Button. Clicking the [Start] button will bring up the test file selection interface, as shown in the figure below:

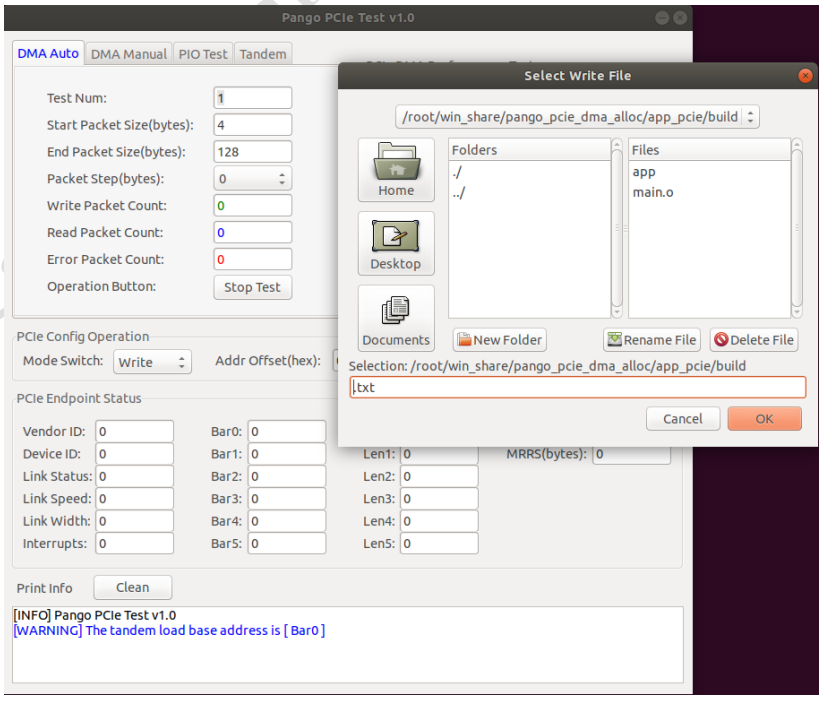
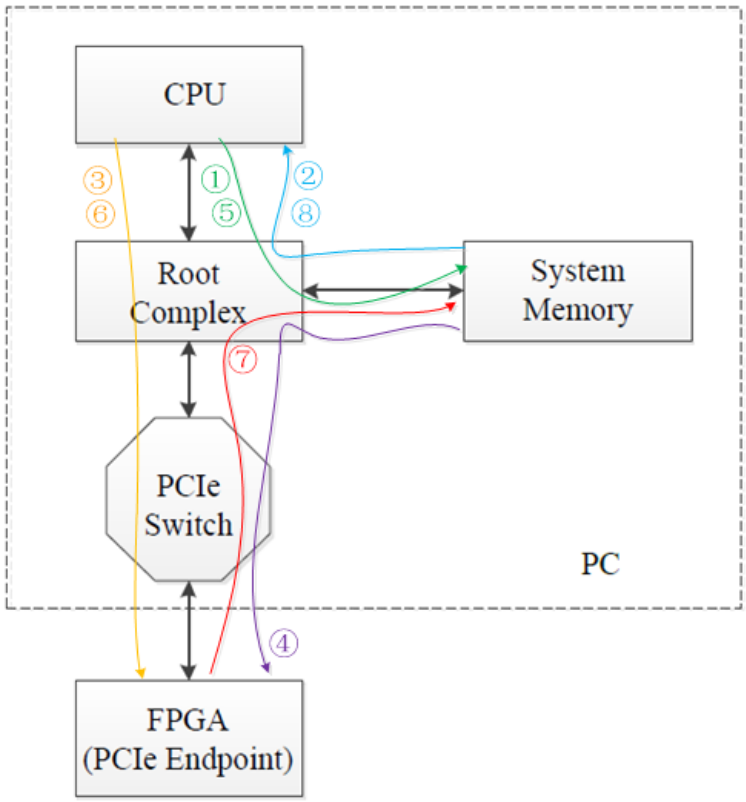


Figure 2-4 Test File Selection Interface

The test file for DMA Auto test needs to be defined by the user, with the contents filled. The number of bytes in the file is preferably greater than the End Packet Size, otherwise, it will fill the insufficient bytes with ASCII codes from 0 to 9. The filled data is only cached in DDR, and will not be written to the test file. A test file is provided in the app_pcie folder with the file name of test.txt , which users can directly use for testing. DMA Auto test process is shown in the following figure:



①:DDR write, CPU writes the data file to DDR.
②:CPU reads the data from DDR to confirm if the written and the output data of DDR are consistent.
③:CPU issues a command to FPGA to trigger FPGA to perform DMA read operation.
④:FPGA performs DMA read operation.
⑤:CPU clears DDR data.
⑥:CPU issues a command to FPGA to trigger FPGA to perform DMA write operation.
⑦:FPGA performs DMA write operation.
⑧:DDR read operation, CPU reads DDR data and compares it with the data written in Step 1.

Figure 2-5 DMA Auto Test Process

2.2 Performance Test Interface

No. 10-16 in the main interface represent the PCIe performance test interface, as shown in the interface below:

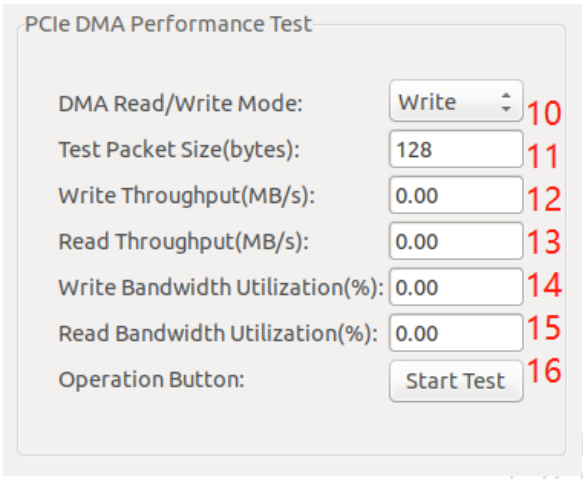


Figure 2-6 Performance Test Interface

The meanings represented by the numbers are as follows:

No. 10-DMA Write Mode: DMA write mode selection, including: Write, Read and W&R modes, as shown in the following figure:

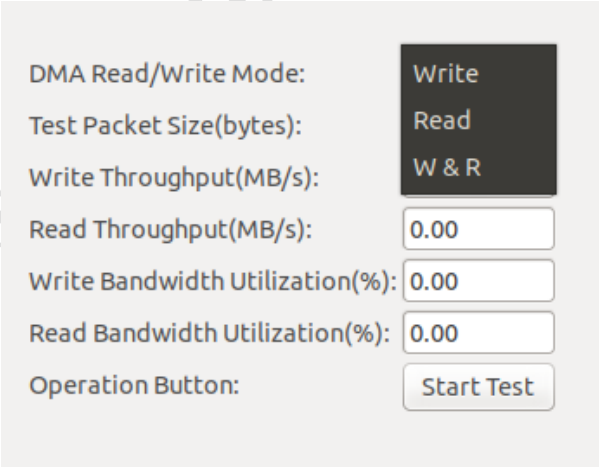


Figure 2-7 DMA Read/Write Selection Interface

No. 11-Test Packet Size (bytes): the data packet length of DMA single transmission, the default length is 128 bytes, in the FPGA hardware engineering of Performance Test, it is 128 bytes. Keep the default value.

No. 12-Write Throughput (MB/s):: DMA write bandwidth.

No. 13-Read Throughput (MB/s):: DMA read Bandwidth.

No. 14-Write Bandwidth Utilization (%): DMA write bandwidth utilization.

No. 15-Read Bandwidth Utilization (%): DMA read bandwidth utilization.

No. 16-Operation Button: Performance test start button, after the button is pressed, the test will be conducted every 1 s until it is manually stopped.

2.3 Config Operation Interface

No. 17-20 in the main interface represent the PCIe configuration register interface, as shown in the interface below:

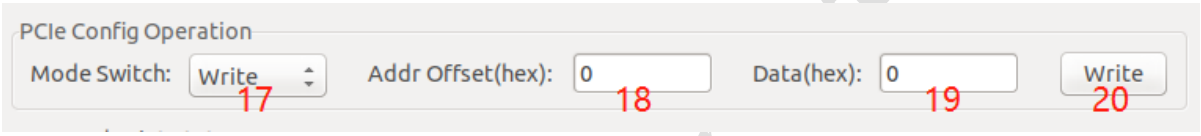


Figure 2-8 Config Operation Interface

No. 17 - Mode Switch: configuration register read/write operation switch (Write/Read), as shown in the figure below:

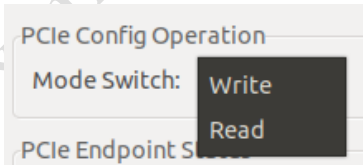


Figure 2-9 Read/Write Switch Selection

No. 18-Addr Offset (hex): read/write offset address, hexadecimal, within configurable address range.

No. 19-Data (hex): For write operations, it indicates the data to be written; for read operations, it indicates the data read from the register. No. 20-Write/Read: Depending on the selection of Mode Switch, different button operations are displayed.

The list of configuration space registers is shown below, and it can be read or written according to the offset addresses shown below:

PCI Express Configuration Space				
31	16	15	0	Address
Device ID		Vendor ID		000h
Status		Command		004h
Class Code			RevID	008h
BIST	Header	Lat Timer	Cache Ln	00Ch
Base Address Register 0				010h
Base Address Register 1				014h
Base Address Register 2				018h
Base Address Register 3				01Ch
Base Address Register 4				020h
Base Address Register 5				024h
Cardbus CIS Pointer				028h
Subsystem ID		Subsystem Vendor ID		02Ch
Expansion ROM Base Address				030h
Reserved			CapPtr	034h
Reserved				038h

Max Lat	Min Gnt	Intr Pin	Intr Line	03Ch
Reserved				040h
				044h
				048h
				04Ch
MSI Control		Next Cap Pointer	Cap ID	050h
Message Address [31:0]				054h
Message Address [63:32]				058h
Reserved		Message Data		05Ch
Mask Bits				060h
Pending Bits				064h
Reserved				068h
Reserved				06Ch
PE Cap		Next Cap Pointer	Cap ID	070h

Device Capabilities			074h
Device Status	Device Control		078h
Link Capabilities			07Ch
Link Status	Link Control		080h
Reserved			084h
			088h
			08Ch
			090h
Device Capabilities 2			094h
Device Status 2	Device Control 2		098h
Link Capabilities 2			09Ch
Link Status 2	Link Control 2		0A0h
Reserved			0A4h
			0A8h
			0ACh
Message Control	Next Pointer	Capability ID	0B0h
Table Offset		Table BIR	0B4h
PBA Offset		PBA BIR	0B8h
Reserved			…
Advanced Error Capabilities and Control Register			118h

Figure 2-10 Configuration Space Register List

2.4 Endpoint Status Interface

No. 21-40 in the main interface represent the PCIe status information interface, as shown in the interface below:

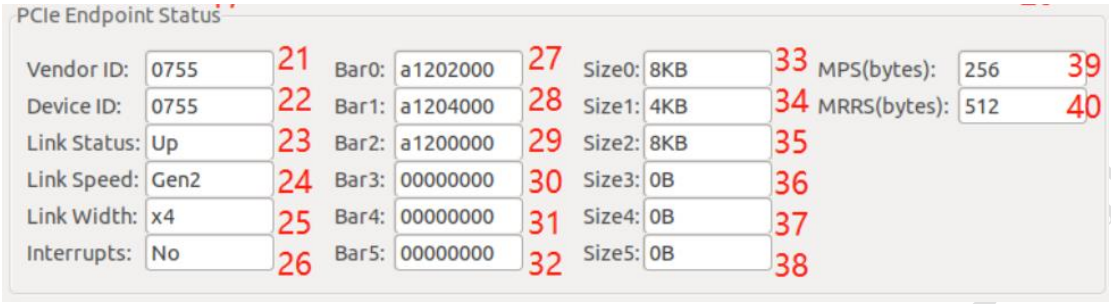


Figure 2-11 Endpoint Status Interface

No. 21-Vendor ID: Vendor ID.

No. 22-Device ID: Device ID.

No. 23-Link Status: Device link status, Up (Link creation successful) and Down (Link creation failed). No. 24-Link Speed: Device link speed.

No. 25-Link Width: Device link width.

No. 26-Interrupts: Whether interrupts are supported (not supported by default).

No. 27-Bar0: Bar0 base address assigned to EP.

No. 28-Bar1: Bar1 base address assigned to EP.

No. 29-Bar2: Bar2 base address assigned to EP.

No. 30-Bar3: Bar3 base address assigned to EP.

No. 31-Bar4: Bar4 base address assigned to EP.

No. 32-Bar5: Bar5 base address assigned to EP.

No. 33-Size0: Space size of Bar0.

No. 34-Size1: Space size of Bar1.

No. 35-Size2: Space size of Bar2.

No. 36-Size3: Space size of Bar3.

No. 37-Size4: Space size of Bar4.

No. 38-Size5: Space size of Bar5.

No. 39-MPS(bytes): Max Payload Size.

No. 40-MRRS(bytes): Max Read Request Size.

2.5 Print Info Interface

Some necessary information can be printed in Print Info interface, as shown in the figure below:

Figure 2-12 Print Info Interface

Most of the print information during the test will be printed on the console through the printf function. The [Clean] button has no actual function and is reserved for use.

2.6 DMA Manual Interface

No. 42-51 in the main interface represent the DMA Manual Test Interface, as shown in the interface below:

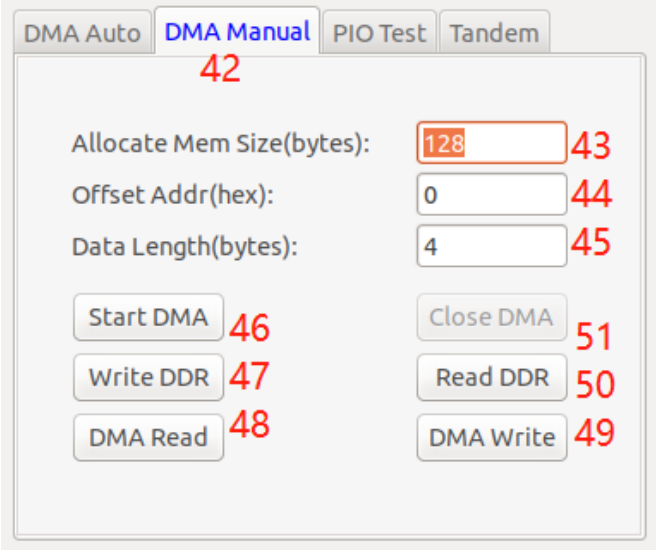


Figure 2-13 DMA Manual Interface

No. 42-DMA Manual: DMA manual test selection interface button.

No. 43-Allocate Mem Size (bytes): DDR cache size to be requested, in bytes (minimum 4 bytes).
No. 44 - Offset Addr (hex): Test base address offset.

No. 45-Data Length (bytes): Length of a data packet in DMA testing, in bytes (minimum 4 bytes).

No. 46-Start DMA: Start DMA manual test button, if it is pressed, a test file selection interface will pop up, as shown in the figure below:

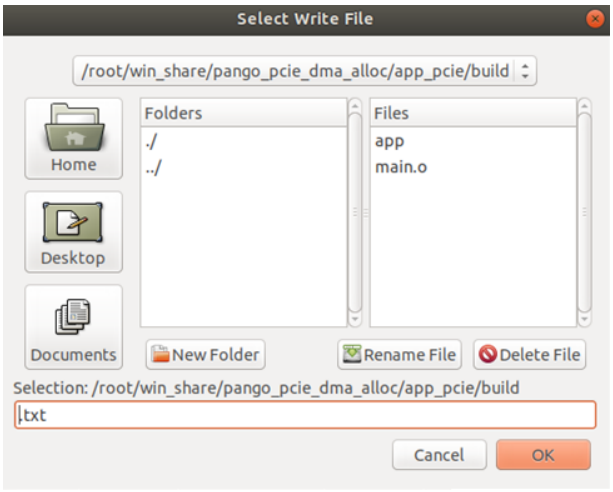


Figure 2-14 Test File Selection Interface

After successfully opening the text, the [Start DMA] button will turn grey and become inoperable, while the [Close DMA] button will turn from grey to black and become operable.

The test file for DMA Manual test needs to be defined by the user, with the contents filled. The number of bytes in the file is preferably greater than the Allocate Mem Size, otherwise, it will fill the insufficient bytes with ASCII codes from 0 to 9. The filled data is only cached in DDR, and will not be written to the test file.

No. 47-Write DDR: Write data to the DDR cache.

No. 48-DMA Read: Perform DMA read operation.

No. 49-Close DMA: DMA write operation.

No. 50-Read DDR: Read data to the DDR cache.

No. 51-DMA Write: Close DMA manual test button. After closing the manual test, the [Close DMA] button will turn grey and become inoperable, while the [Start DMA] button will turn from grey to black.The correct DMA manual test operation process No. 46-51: Start DMA → Write DDR →DMA Read →DMA Write →Read DDR →Close DMA.

2.7 PIO Test Interface

No. 52-59 represent the PIO test interface, as shown in the figure below:

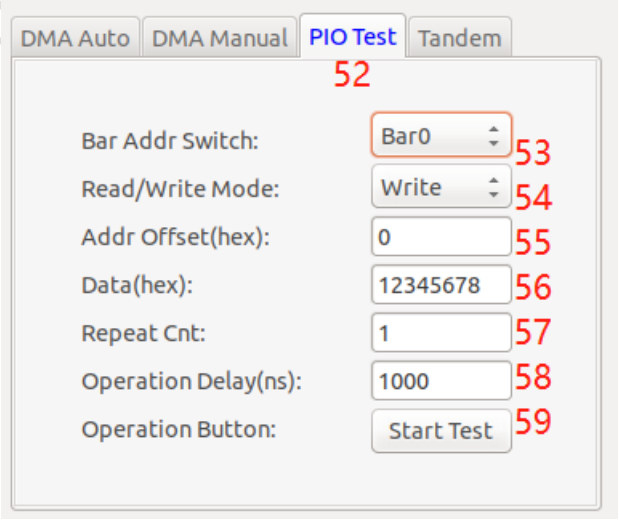


Figure 2-15 PIO Test Interface

No. 52-PIO Test: PIO test interface selection button.

No. 53-Bar Addr Switch: Bar base address selection interface, including Bar0, Bar1, Bar2, Bar3, Bar4, Bar5,as shown in the following figure.

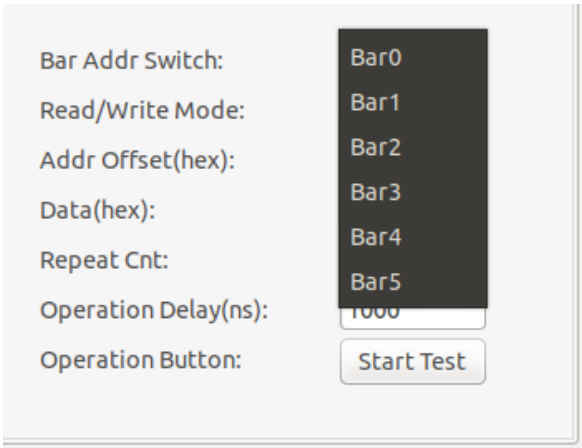


Figure 2-16 Bar Selection

No. 54-Read/Write Mode: Read/Write operation switch options, including Write, Read, W&R, W&R Step. W&R Step indicates that when performing read/write operations again, the address will increment by 4 each time, but will not exceed the Bar space size.

No. 55-Addr Offset (hex): Bar read/write operation offset address.

No. 56-Data (hex): Indicates the data to be written during write operations (for multiple write operations, data will increment by 1 from the initial value); indicates the data to be read during read operations.

No.57-Repeat Cnt: Number of operations, if it is 0, it indicates the read and write loop continues until the operation is manually stopped.

No. 58-Operation Delay (ns): Operation delay for one (read, write, or write then read) operation, measured in ns, it is 1000 ns by default.

No. 59-Operation Button: PIO test start button. After configuring the above parameters, click the [Start Test] button.

2.8 Tandem Interface

No. 60-62 in the main interface represent the Tandem loading interface, as shown in the figure below:

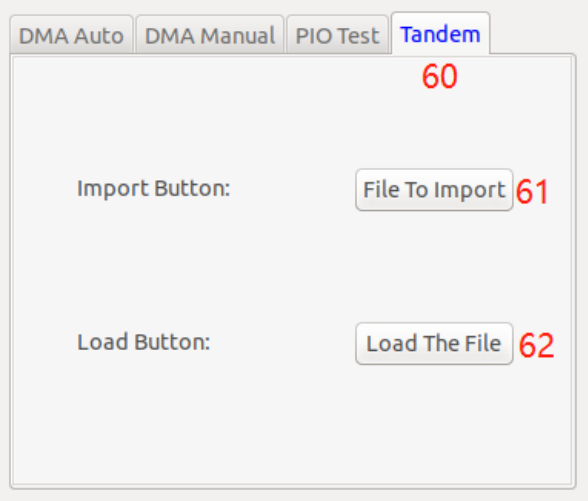


Figure 2-17 Tandem Interface

No. 60-Tandem: Tandem loading interface selection button.

No. 61-Import Button: Bitstream file selection button, if it is pressed, the file selection interface will pop up, as shown in the figure below:

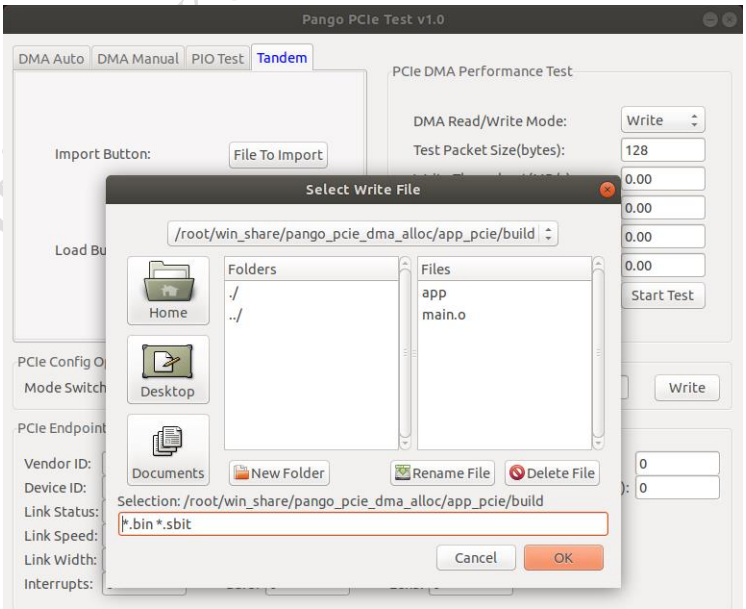


Figure 2-18 Tandem Bitstream Loading Selection Interface

The loading bitstream is in PG2L100H_PCIe_Tandem folder, please refer to the "Quick Loading Instructions" document for specific operations. No. 62 - Load Button: Bitstream loading button.

Tandem loading process: click the [File To Import] button first to import the bitstream file to be loaded, then click the [Load The File] button to start loading the bitstream file.

Chapter 3 Run Test Interface

Enter the project folder, it contains two folders and one file, namely: app_pcie and driver folders as well as run.sh script file.

3.1 Supporting Hardware

Hardware test environment: PG2L100htest board (P04I100KF01_A2)

PDS version: Pango Design Suite 2021.1-SP6.2

First, download FPGA bitstream from pango_pcie_dma_alloc/pcie_test_rtl directory. The hardware project file names for different test items are as follows:

DMA Auto, DMA Manual, PIO Test: PG2L100H_PCIe_DMA (the example is generated by using IP)

Tandem: PG2L100H_PCIe_Tandem (contains three bitstream files and quick loading instructions)

Performance test: PG2L100H_PCIe_performance

Config Operation can be performed without relying on hardware projects.

3.2 Operating Environment

The package is installed on a Linux system (not a virtual machine), with the GTK package also installed. If a GTK compilation error occurs, please search for GTK installation tutorials on Baidu (GTK version: GTK+-2.0).

This test code has been successfully tested on Fedora 16 and Ubuntu 16.04.

3.3 Run Script

The provided disc is Fedora 16, the method to open the console is different from Ubuntu 16.04. They will be introduce separately:

Step 1: After system startup, copy the pango_pcie_dma_alloc folder to the system desktop (to the Desktopfolder) first

Step 2: Start the console

Fedora 16: Move the mouse to the top left corner of Applications, and select "System Tools", then "Terminal Ubuntu 16.04": "Ctrl+Alt+T"

After entering the console, the operations are identical, taking Ubuntu 16.04 as an example.

Step 3: Entering administrator privileges

Fedora 16: Type su in the console, click the "Enter" button to enter directly without a password

Ubuntu 16.04: Type su in the console, click the "Enter" button, and enter the root privilege password, as shown in the figure below:

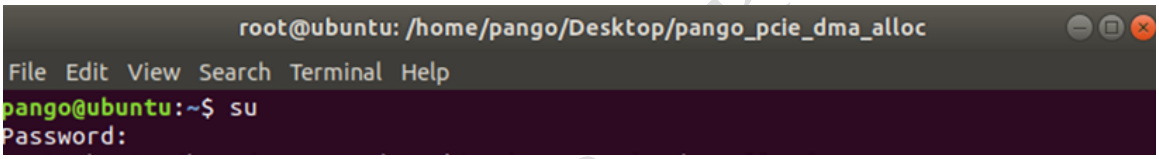


Figure 3-1 Entering Administrator Privileges

Step 4: Enter the "pango_pcie_dma_alloc" folder ("cd Desktop/pango_pcie_dma_alloc/"), then click the "Enter" button as shown in the figure below:

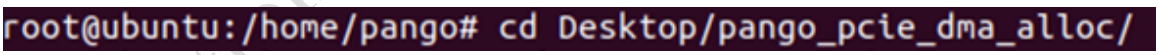


Figure 3-2 Entering pango_pcie_dma_alloc Folder

Step 5: Modify "run.sh" script file permissions (chmod –R 777 run.sh), and click the "Enter" button as shown in the figure below:

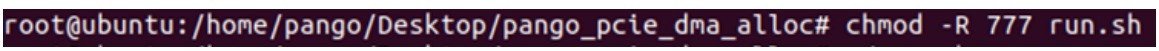


Figure 3-3 Modifying run.sh Script File Permissions

Step 6: Run the "run.sh" script (./run.sh), and click the "Enter" button as shown in the figure below:

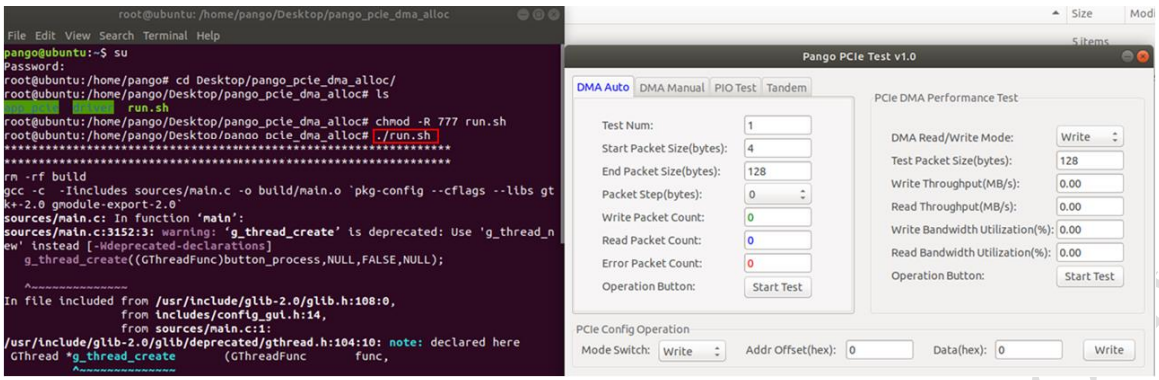


Figure 3-4 Script Execution File

If the current console user is not an administrator, a prompt will appear when the script is run, as shown in the figure below:

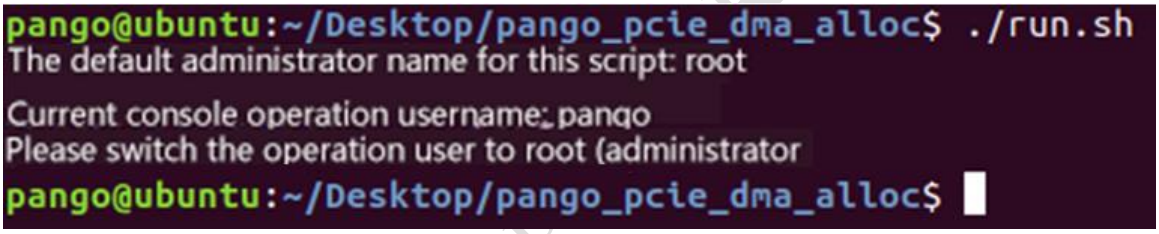


Figure 3-5 Script Prompt Information

The default administrator name for the script is root. If the user's Linux system administrator name is not root, it is necessary to modify the root_name variable of the run.sh script file, as shown in the figure below:

Figure 3-6 root_name Variable

Change root_name='root' to root_name='xxx', xxx is the user's Linux system administrator name.

3.4 Closing Debug Information

When running in the test interface, there is a lot of print information: the debug information is shown in green, the regular print information is shown in blue, the warning information is shown in yellow, and the error information is shown in red. To close the debug information, it is required to comment out the DEBUG macro definition in the app_pcie/includes/config_gui.h file, as shown in line 20 in the figure below, then re-run the run.sh script.

3.5 Version Compatibility

In the lower versions of GTK, there is an incompatibility in component size with the higher versions.

3.6 Modify ID

During PCIe driver loading, the vendor ID and the device ID must match the vendor ID and the device ID in the FPGA hardware project,which can be modified in driver/id_config.h, as shown in the figure below:

```
1: #ifndef _ID_CONFIG_H_
2: #define _ID_CONFIG_H_
3:
4: #define PCI_PANGO_DEFAULT_VENDOR_ID     0x0755
5: #define PCI_PANGO_DEFAULT_DEVICE_ID     0x0755
6:
7: #endif // _ID_CONFIG_H_
```

Figure 3-7 ID Macro Definition

Chapter 4 Attachment

The figure below lists all error message descriptions for easy self-checking:

Figure 4-1 Error Information Self-check

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its series products at any time without prior notice.
4. The information contained in this document is intended to assist users in resolving application-related issues. While we strive for accuracy, we cannot guarantee that the document is entirely free from flaws. Should any functional abnormalities and performance degradation arise due to deviation from the prescribed procedures outlined herein, our company will neither be held liable nor concede that such issues stem from product deficiencies. The solutions presented in this document are just one of the feasible options and cannot cover all application scenarios. Consequently, if users encounter functional abnormalities or performance degradation despite adhering to the prescribed procedures outlined herein, we cannot assure that such issues are indicative of product deficiencies