Intro:-

ELM stands for Embedded Linux Mezzanine, which is a small device based on Ultrascale+ ZYNQ FPGA. The main purpose of ELM is to serve as an on board control interface for ATCA modules. These test suites are being developed to automate/semi-automate the bring up tests of the ELM.
This manual includes the ZYNQ-DDR interface testing steps from HW configuration to SDK project building and actual testing on ELM. The DDR tests have to be done by a software program which already exists. The testing software should write various data patterns into the memory and then read back, thus we can check if the signal conditions of the PS-DDR interface are good enough or not. The DDR4 memory on the board is MT40A1G16KNR-075:E is a main RAM for the CPU having a total range of 4GB.

Requirements-

- Vivado Design Suite (V2019.1) + SDK installed
- Board connected to the system with JTAG and USB-UART connection.

Launching Vivado

----Steps to launch vivado on madorsky-d2---

- run this script, shown relative to the home directory:

```
>>> ~/bin/viv191.sh
```

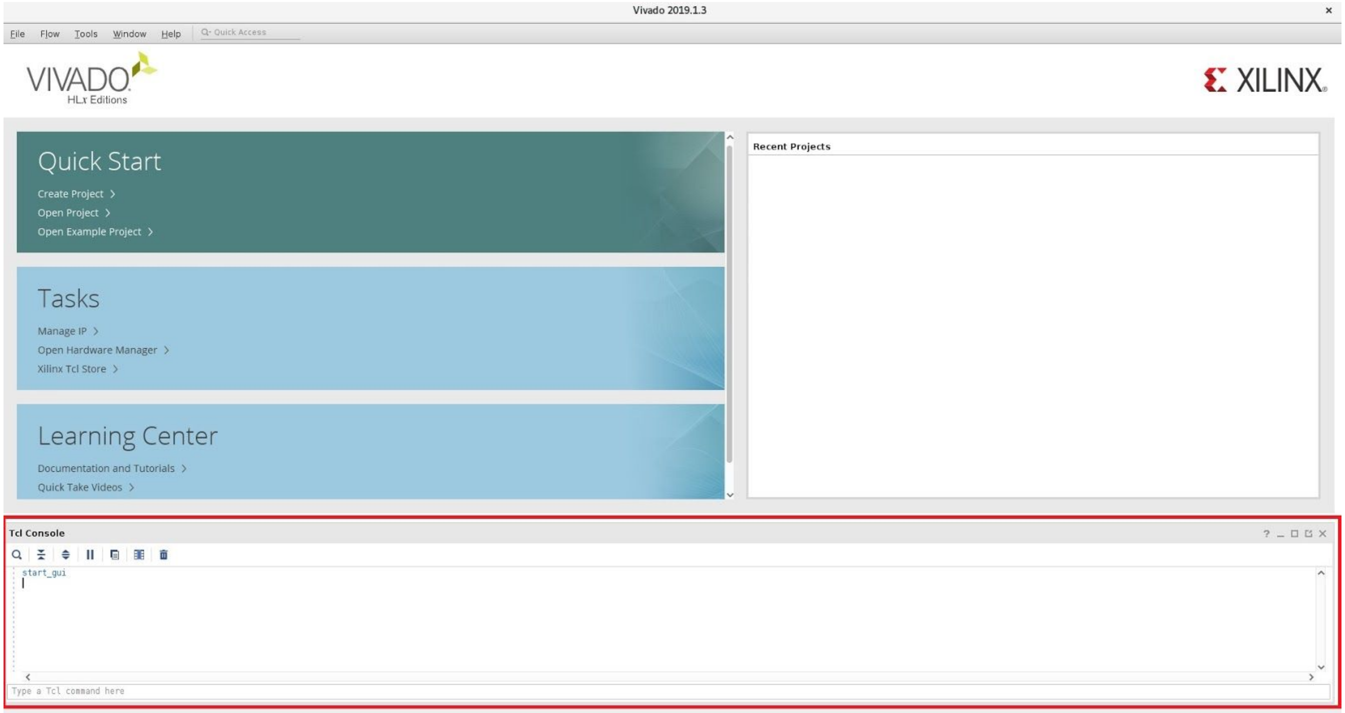
Hardware (HW) configuration in Vivado-

1. PS-DDR interface in vivado-

The HW configuration is done using Vivado Design Automation and IP integrator for simplicity. To automate the process of HW configuration, a tcl script is developed.

Running the TCL script-

1. After launching the Vivado,There are 2 options available to run the tcl script
    a. Using tcl console - Highlighted in Red below



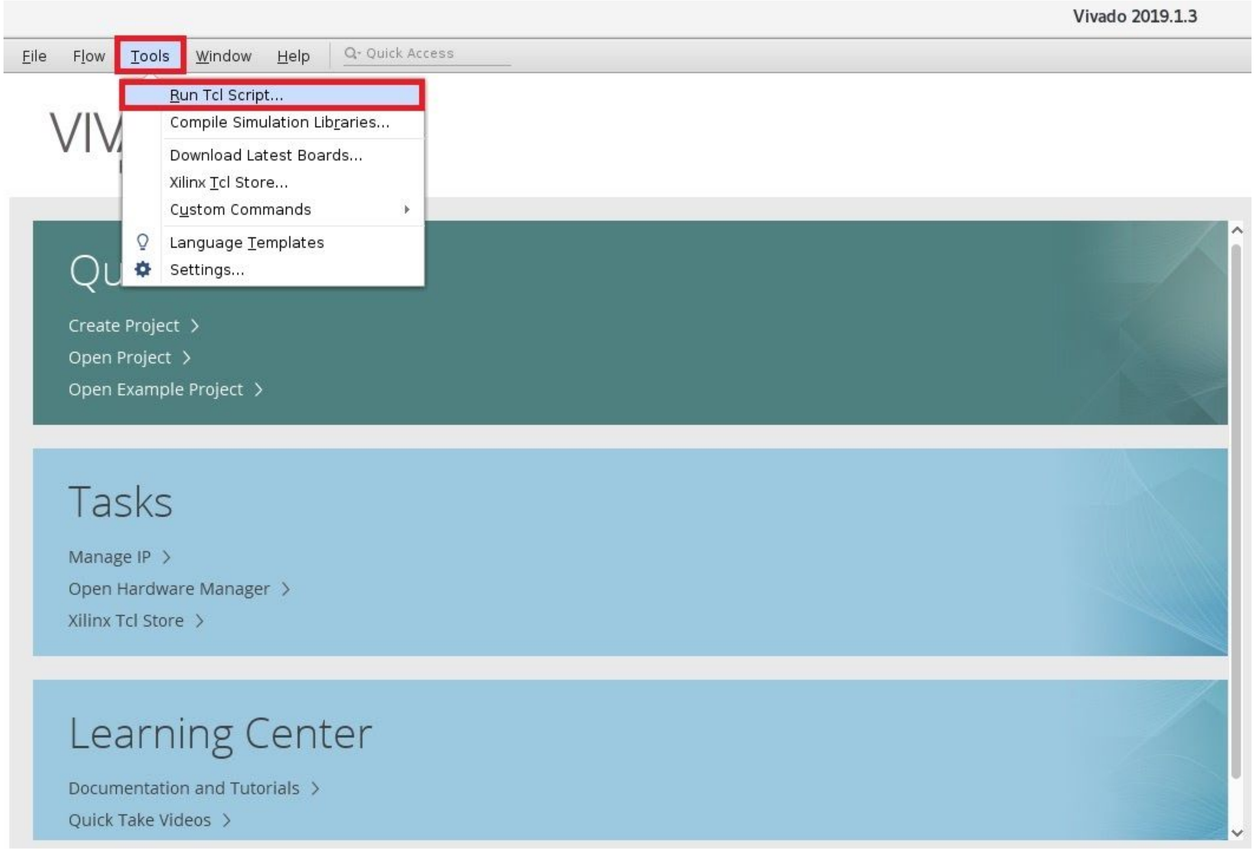
```
start_gui
```

Run the following commands-

```
> cd <your project dir>
> source psddr.tcl
```

    b. In Vivado GUI, Use option Tools-> Run Tcl Script

Here, the Tcl file is sourced by using GUI instead of using command line.



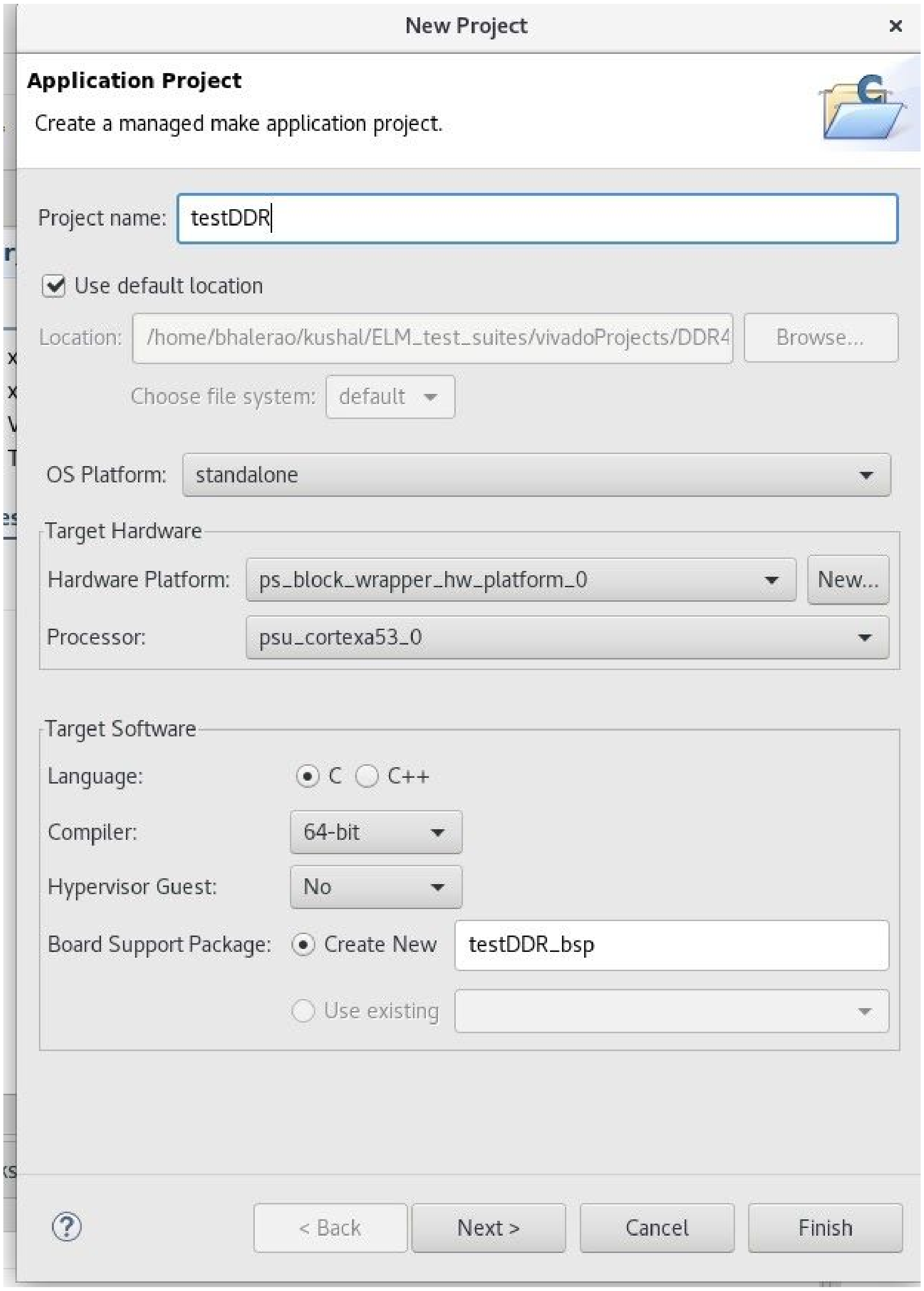
Both approaches do the same job.

The TCL script does the following tasks-

- Checks if the script is running the correct version (2019.1).
- Creates the new vivado project at current working directory (where the tcl is sourced) with the device as- xczu4cg-sfvc784-1-e
- Creates the block design in IP integrator with name 'ps_block'
- Configures ZYNQ UL+ IP with required parameters.
    - DDR
    - Clocks
    - PSU banks(0-3) IO standards
    - UART (MIO 5...9)
- Validates and saves the design
- Generate the HDL wrapper in target language and add it to the source file as a top module.
- Generates the Output products
- Export the hardware files to SDK
- Launch the Xlinx's Software Development Kit (SDK)

Xilinx’s Software Development Kit (SDK)

1. Once the SDK is launched, in the project explorer tab, select the ps_block_wrapper_hw_platform_0 and File->New->Application Project. In the new project window, give the project name , and follow the settings below and click next.



New Project

Application Project

Create a managed make application project.

Project name: testDDR

☑ Use default location

Location: /home/bhalerao/kushal/ELM_test_suites/vivadoProjects/DDR4

Choose file system: default

OS Platform: standalone

Target Hardware

Hardware Platform: ps_block_wrapper_hw_platform_0

Processor: psu_cortexa53_0

Target Software

Language: ● C ○ C++

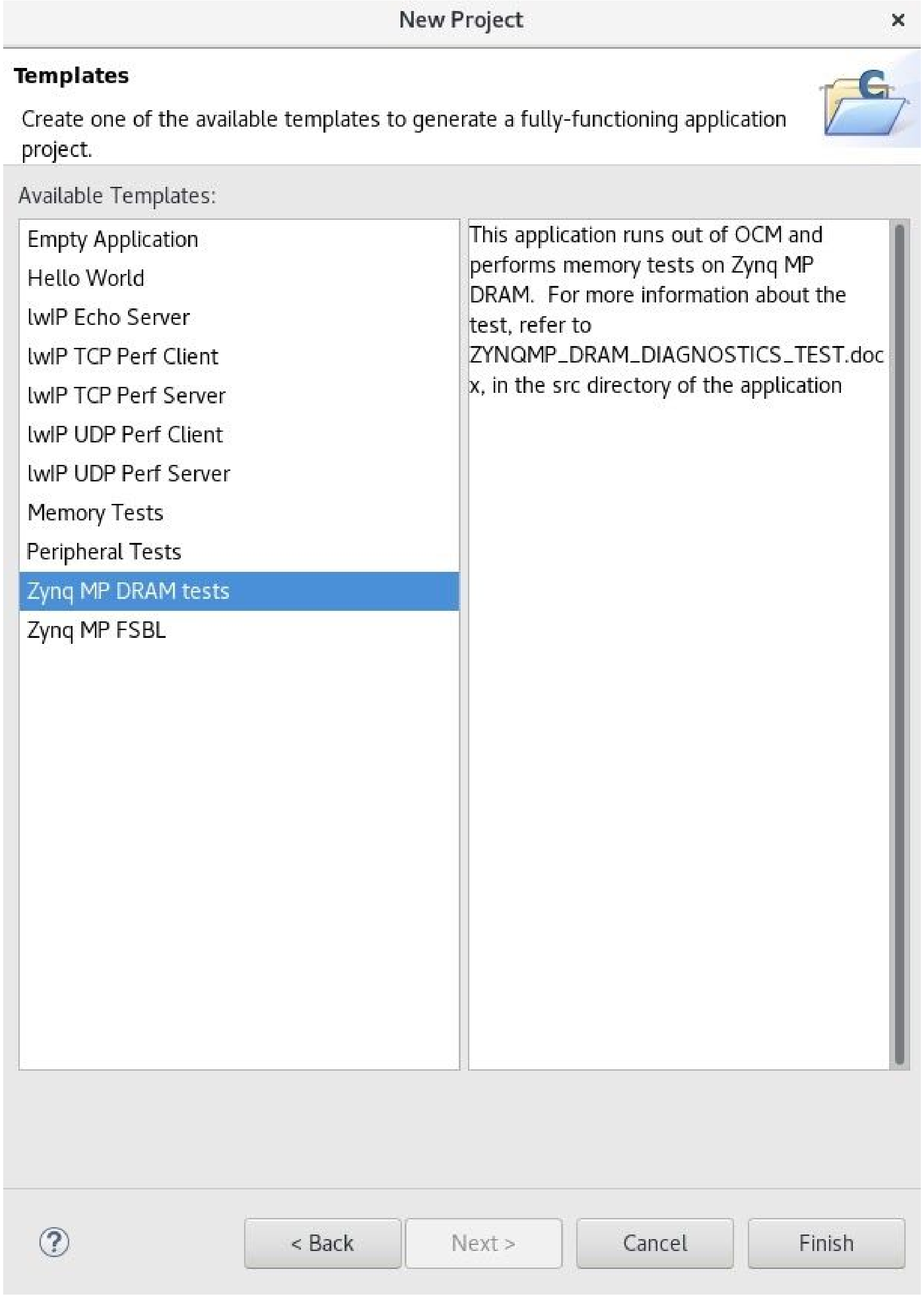
Compiler: 64-bit

Hypervisor Guest: No

Board Support Package: ● Create New testDDR_bsp

○ Use existing

2. Next window would show the example templates available from Xilinx. The application template used for this testing is ‘ZYNQ DRAM tests’. Click Finish.



New Project

Templates

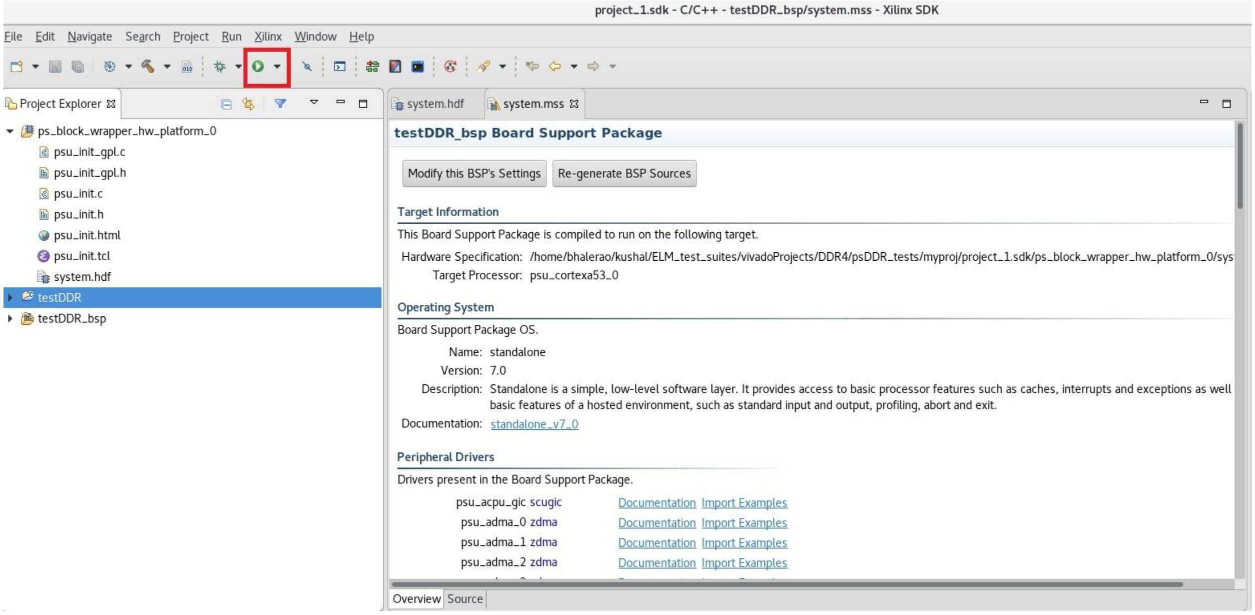
Create one of the available templates to generate a fully-functioning application project.

Available Templates:

- Empty Application
- Hello World
- lwIP Echo Server
- lwIP TCP Perf Client
- lwIP TCP Perf Server
- lwIP UDP Perf Client
- lwIP UDP Perf Server
- Memory Tests
- Peripheral Tests
- Zynq MP DRAM tests
- Zynq MP FSBL

This application runs out of OCM and performs memory tests on Zynq MP DRAM. For more information about the test, refer to ZYNQMP_DRAM_DIAGNOSTICS_TEST.docx, in the src directory of the application

3. SDK adds this project to the Project Explorer window with its required bsp automatically generated. Once added, SDK automatically builds the project. Wait till the ‘Build Finished’ message prints on the console.Once Project is built successfully, select the application in Project Explorer window, locate the green button ‘Run’ tab and select ‘Run Configuration’.



- ps_block_wrapper_hw_platform_0
  - psu_init_gpl.c
  - psu_init_gpl.h
  - psu_init.c
  - psu_init.h
  - psu_init.html
  - psu_init.tcl
  - system.hdf
- testDDR
- testDDR_bsp

testDDR_bsp Board Support Package

Modify this BSP's Settings | Re-generate BSP Sources

Target Information

This Board Support Package is compiled to run on the following target.

Hardware Specification: /home/bhalerao/kushal/ELM_test_suites/vivadoProjects/DDR4/psDDR_tests/myproj/project_1.sdk/ps_block_wrapper_hw_platform_0/sys
Target Processor: psu_cortexa53_0

Operating System

Board Support Package OS.

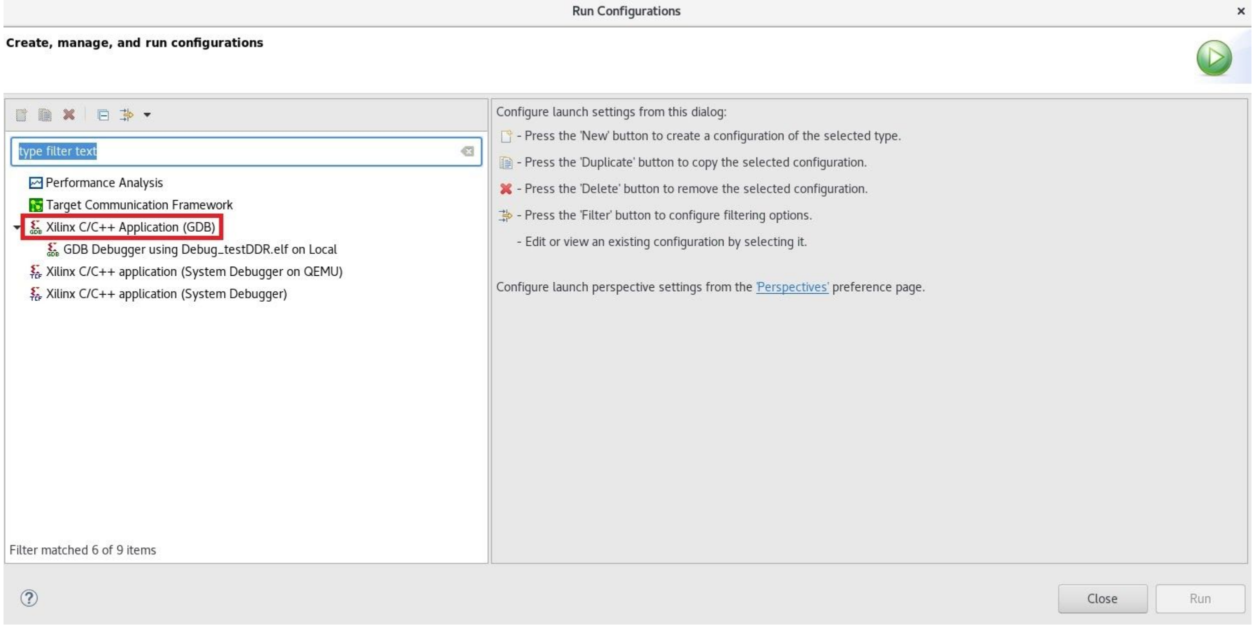
Name: standalone
Version: 7.0
Description: Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well basic features of a hosted environment, such as standard input and output, profiling, abort and exit.
Documentation: standalone_v7_0

Peripheral Drivers

Drivers present in the Board Support Package.

psu_acpu_gic	scugic	Documentation	Import Examples
psu_adma_0	zdma	Documentation	Import Examples
psu_adma_1	zdma	Documentation	Import Examples
psu_adma_2	zdma	Documentation	Import Examples

4. In the Run configuration window, double click, Xilinx C/C++ Application (GDB).



Create, manage, and run configurations

type filter text

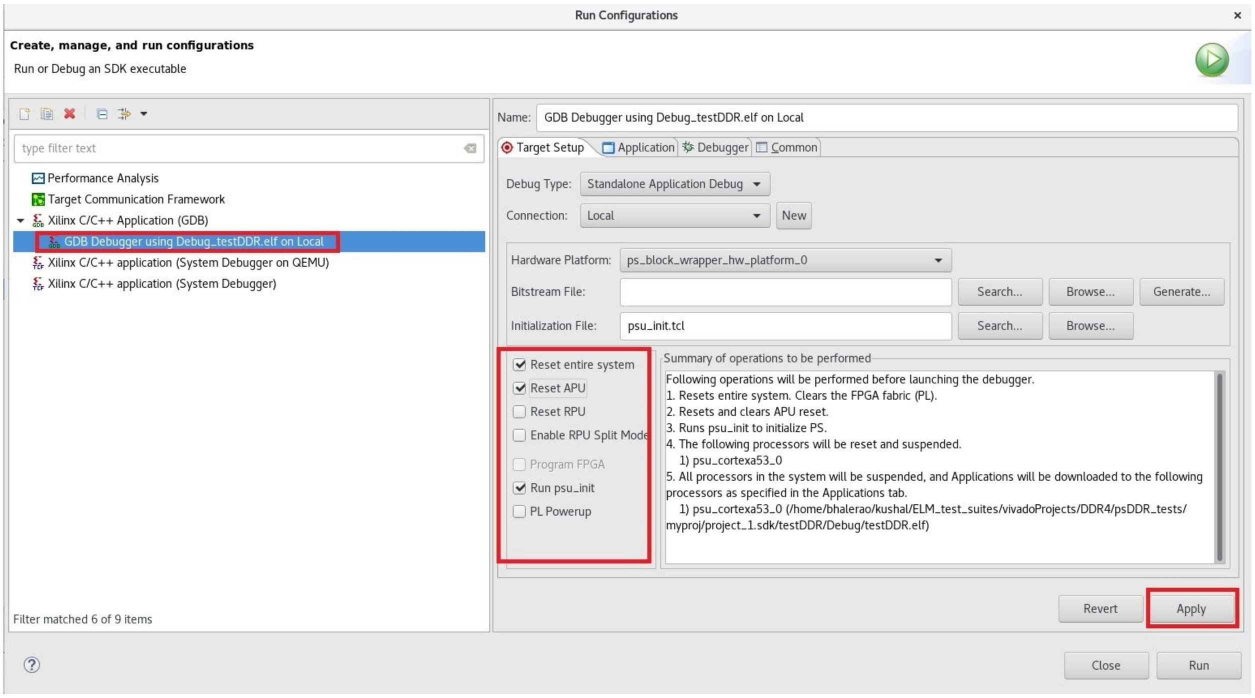
- Performance Analysis
- Target Communication Framework
- Xilinx C/C++ Application (GDB)
  - GDB Debugger using Debug_testDDR.elf on Local
- Xilinx C/C++ application (System Debugger on QEMU)
- Xilinx C/C++ application (System Debugger)

Filter matched 6 of 9 items

Configure launch settings from this dialog:
- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

5. Then select the GDB Debugger using Debug_testDDR.elf on local and follow the settings from the figure below and Click ‘Apply’



Run Configurations

Create, manage, and run configurations

Run or Debug an SDK executable

type filter text
Performance Analysis
Target Communication Framework
Xilinx C/C++ Application (GDB)
GDB Debugger using Debug_testDDR.elf on Local
Xilinx C/C++ application (System Debugger on QEMU)
Xilinx C/C++ application (System Debugger)
Filter matched 6 of 9 items
Target Setup
Application
Debugger
Common
New
Search...
Browse...
Generate...
Revert
Apply
Close
Run

Name: GDB Debugger using Debug_testDDR.elf on Local

Debug Type: Standalone Application Debug

Connection: Local

Hardware Platform: ps_block_wrapper_hw_platform_0

Bitstream File:

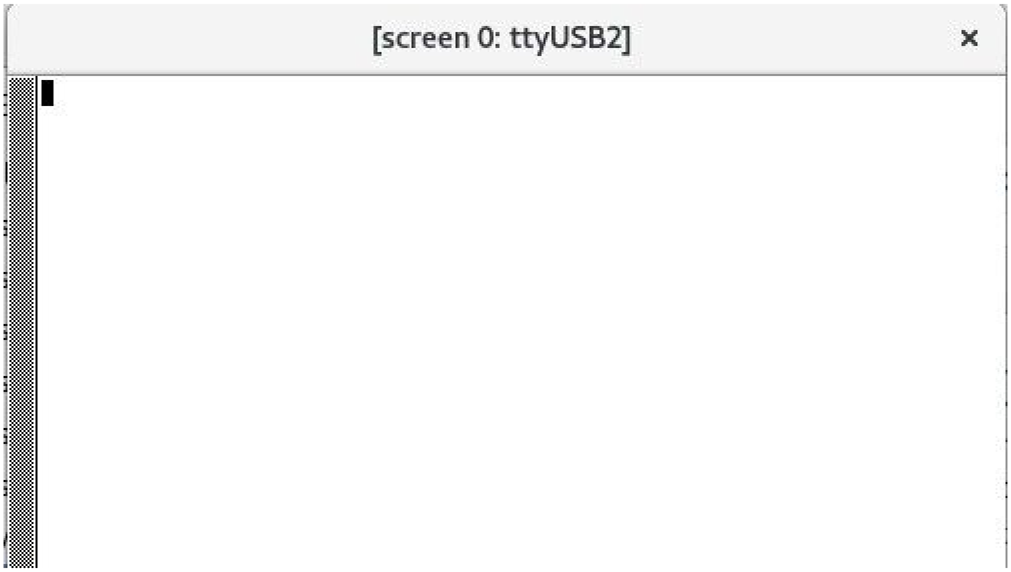
Initialization File: psu_init.tcl

☑ Reset entire system
☑ Reset APU
☐ Reset RPU
☐ Enable RPU Split Mode
☐ Program FPGA
☑ Run psu_init
☐ PL Powerup

Summary of operations to be performed

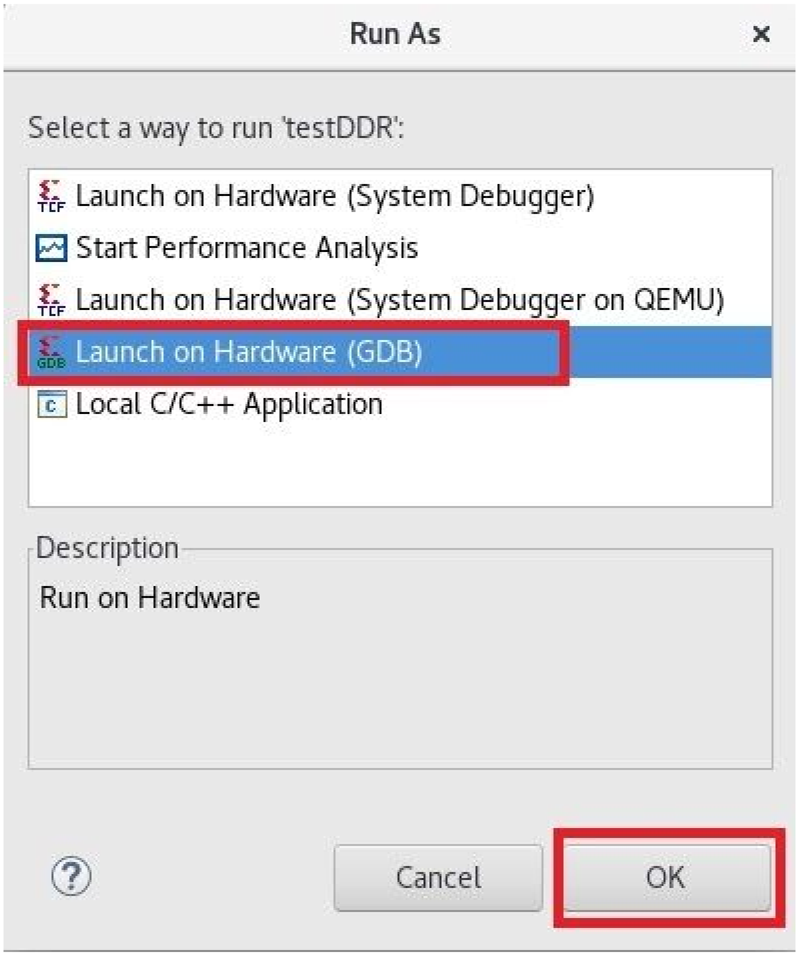
```
Following operations will be performed before launching the debugger.
1. Resets entire system. Clears the FPGA fabric (PL).
2. Resets and clears APU reset.
3. Runs psu_init to initialize PS.
4. The following processors will be reset and suspended.
   1) psu_cortexa53_0
5. All processors in the system will be suspended, and Applications will be downloaded to the following
processors as specified in the Applications tab.
   1) psu_cortexa53_0 (/home/bhalerao/kushal/ELM_test_suites/vivadoProjects/DDR4/psDDR_tests/
myproj/project_1.sdk/testDDR/Debug/testDDR.elf)
```

6. Before hitting ‘Run’, make sure the JTAG and USB-UART connection is done on the board and board is powered ON.
7. Verify the USB port by command- dmesg | grep ttyUSB. In case of madorsky-d2 machine, USB-UART bridge is connected to /dev/ttyUSB2
8. Open up a terminal/xterm and do- screen /dev/ttyUSB2 115200.



[screen 0: ttyUSB2]

9. Now, hit the ‘Run’ button in the SDK. A pop- up window named ‘Run-As’ would appear, select Launch on Hardware (GDB).



Run As

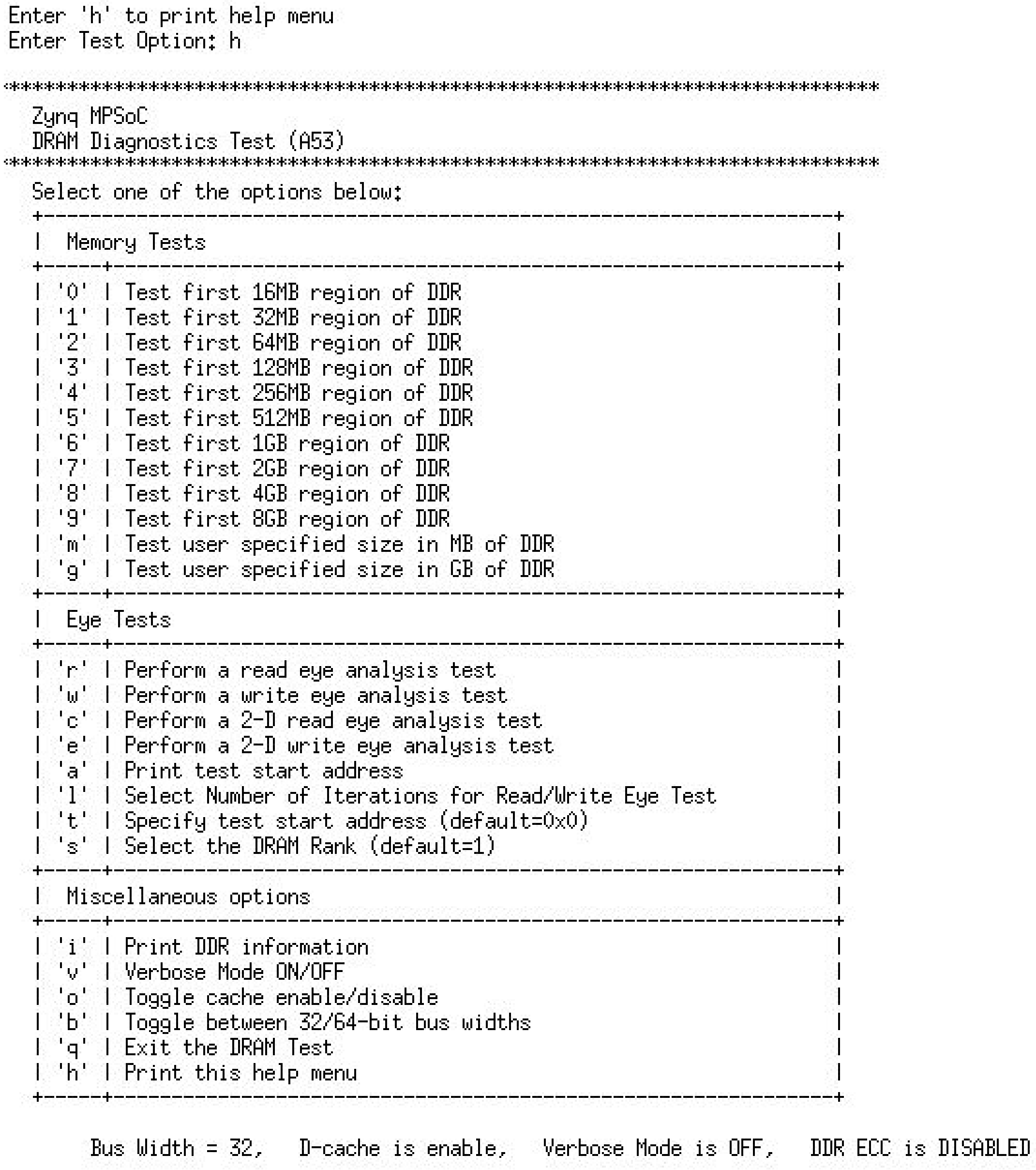
Select a way to run 'testDDR':

Description

Run on Hardware

×
Launch on Hardware (System Debugger)
Start Performance Analysis
Launch on Hardware (System Debugger on QEMU)
Launch on Hardware (GDB)
Local C/C++ Application
?
Cancel
OK

10. If run is successful, below window would be appeared on the serial console.

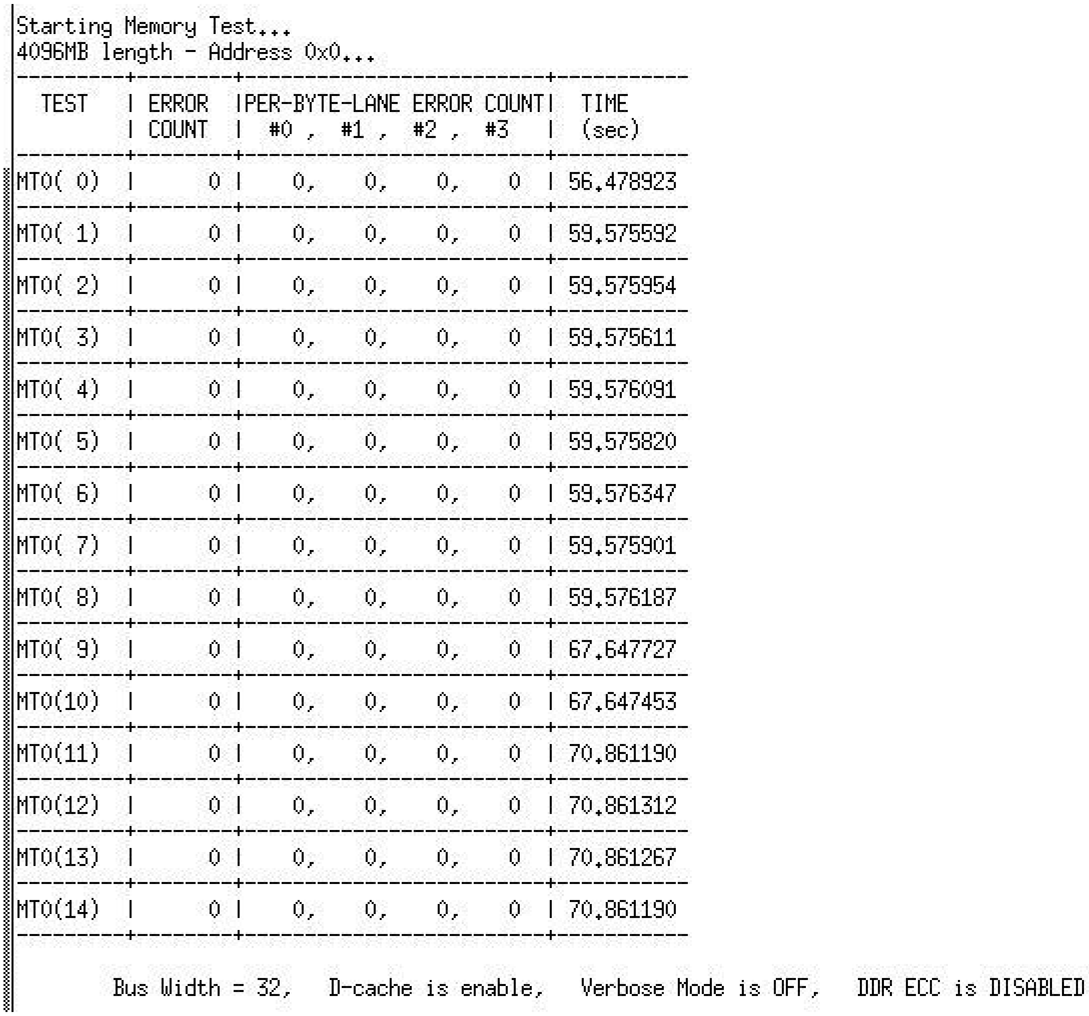


```
Enter 'h' to print help menu
Enter Test Option: h

******************************************************************************
  Zynq MPSoC
  DRAM Diagnostics Test (A53)
******************************************************************************
  Select one of the options below:
  +--------------------------------------------------------------------------+
  |  Memory Tests                                                            |
  +-----+--------------------------------------------------------------------+
  | '0' | Test first 16MB region of DDR                                      |
  | '1' | Test first 32MB region of DDR                                      |
  | '2' | Test first 64MB region of DDR                                      |
  | '3' | Test first 128MB region of DDR                                     |
  | '4' | Test first 256MB region of DDR                                     |
  | '5' | Test first 512MB region of DDR                                     |
  | '6' | Test first 1GB region of DDR                                       |
  | '7' | Test first 2GB region of DDR                                       |
  | '8' | Test first 4GB region of DDR                                       |
  | '9' | Test first 8GB region of DDR                                       |
  | 'm' | Test user specified size in MB of DDR                              |
  | 'g' | Test user specified size in GB of DDR                              |
  +-----+--------------------------------------------------------------------+
  |  Eye Tests                                                               |
  +-----+--------------------------------------------------------------------+
  | 'r' | Perform a read eye analysis test                                   |
  | 'w' | Perform a write eye analysis test                                  |
  | 'c' | Perform a 2-D read eye analysis test                               |
  | 'e' | Perform a 2-D write eye analysis test                              |
  | 'a' | Print test start address                                           |
  | 'l' | Select Number of Iterations for Read/Write Eye Test                |
  | 't' | Specify test start address (default=0x0)                           |
  | 's' | Select the DRAM Rank (default=1)                                   |
  +-----+--------------------------------------------------------------------+
  |  Miscellaneous options                                                   |
  +-----+--------------------------------------------------------------------+
  | 'i' | Print DDR information                                              |
  | 'v' | Verbose Mode ON/OFF                                                |
  | 'o' | Toggle cache enable/disable                                        |
  | 'b' | Toggle between 32/64-bit bus widths                                |
  | 'q' | Exit the DRAM Test                                                 |
  | 'h' | Print this help menu                                               |
  +-----+--------------------------------------------------------------------+

       Bus Width = 32,   D-cache is enable,   Verbose Mode is OFF,   DDR ECC is DISABLED
```

11. The DDR test application is an interactive test, where the user can test the DDR for various memory ranges by selecting the appropriate command shown above.
12. The total range till which we can test is 4GB. So press‘8’ and hit <Enter>, it will start testing the selected range (4GB) with 15 different data patterns, starting from the logical address ‘0x0000000000000000’.



```
Starting Memory Test...
4096MB length - Address 0x0...
----------+---------+-------------------------+-----------
  TEST    | ERROR   |PER-BYTE-LANE ERROR COUNT|  TIME
          | COUNT   |  #0 ,  #1 ,  #2 ,  #3   |  (sec)
----------+---------+-------------------------+-----------
MT0( 0)   |       0 |     0,    0,    0,    0 | 56.478923
----------+---------+-------------------------+-----------
MT0( 1)   |       0 |     0,    0,    0,    0 | 59.575592
----------+---------+-------------------------+-----------
MT0( 2)   |       0 |     0,    0,    0,    0 | 59.575954
----------+---------+-------------------------+-----------
MT0( 3)   |       0 |     0,    0,    0,    0 | 59.575611
----------+---------+-------------------------+-----------
MT0( 4)   |       0 |     0,    0,    0,    0 | 59.576091
----------+---------+-------------------------+-----------
MT0( 5)   |       0 |     0,    0,    0,    0 | 59.575820
----------+---------+-------------------------+-----------
MT0( 6)   |       0 |     0,    0,    0,    0 | 59.576347
----------+---------+-------------------------+-----------
MT0( 7)   |       0 |     0,    0,    0,    0 | 59.575901
----------+---------+-------------------------+-----------
MT0( 8)   |       0 |     0,    0,    0,    0 | 59.576187
----------+---------+-------------------------+-----------
MT0( 9)   |       0 |     0,    0,    0,    0 | 67.647727
----------+---------+-------------------------+-----------
MT0(10)   |       0 |     0,    0,    0,    0 | 67.647453
----------+---------+-------------------------+-----------
MT0(11)   |       0 |     0,    0,    0,    0 | 70.861190
----------+---------+-------------------------+-----------
MT0(12)   |       0 |     0,    0,    0,    0 | 70.861312
----------+---------+-------------------------+-----------
MT0(13)   |       0 |     0,    0,    0,    0 | 70.861267
----------+---------+-------------------------+-----------
MT0(14)   |       0 |     0,    0,    0,    0 | 70.861190
----------+---------+-------------------------+-----------

        Bus Width = 32,   D-cache is enable,   Verbose Mode is OFF,   DDR ECC is DISABLED
```

13. In the ‘src’ tab, a document named ‘ZYNQMP_DRAM_DIAGNOSTICS_TEST.doc’ would be available, which briefs about the test patterns being used.