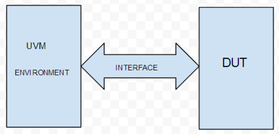
PERFORMANCE VERIFICATION OF NAND FLASH CONTROLLER USING UVM

Advisor: Prof. Morris Jones **Kriti Ashok, Sayali Basole**

Department of Electrical Engineering, San Jose State university, San Jose, California 95192.

Introduction

The objective of this project is to verify the performance of a NAND flash controller by designing the reusable test bench using Universal Verification Methodology (UVM). Verification is very important aspect as it avoids the cost of rework of chip which is already built. The verification environment will verify the performance of NAND flash controller by monitoring the throughput and latency for read and writes operation for different number of pages in the block[6]. The DUT transactions are monitored for specific performance parameters and checked against its actual expected values for needed performance and performance is evaluated. These features provide capability to build reusable verification environment which is very lightly coupled with the type of controller. Developed environment is made so generic that most of the NAND flash controller can be verified.

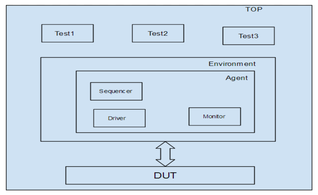


Main task of UVM verification environment is to generate stimulus which are given to the DUT and monitor the response coming from the DUT. Design under test consist of two main parts: NAND flash controller and NAND flash memory and there is one bus master which is used to connect memory and controller and it is responsible for handling the different read , write , read Id transactions.[2]

Methodology

UVM Environment :

Reusable verification environment is built using UVM. Results is monitored and recorded using performance monitor for each scenario of NAND flash controller[1]. Environment communicates to DUT through virtual interface. Driver class communicates send data to DUT by Pin wiggling and Function of monitor is to monitor these pin wiggles and observe the output coming from the DUT.



UVM environment is implemented using the components shown above in the figure. Once the base of verification environment is set and all the components are placed in proper hierarchy and well connected same environment can be used for different tests [3].

Methodology

The NAND flash design used in this project can support the following operations or commands

- Reset
- Read ID
- Erase (per block basis)
- Program Page (copy content of data buffer into Flash memory)
- Read Page (content of a Flash page is copied into the data buffer)

Function	Command Code	Description
Reset	FFh	The reset functions is used to abort read and write
Read ID	90h	This function is used the read the id of the memory used. Complete description
Read	00h and 30h	Read completes one page or 4314 bytes
Page Program	80h and 10h	Write completes one page or 4314 bytes
Block Erase	60h and D0h	Erases one complete block of the flash

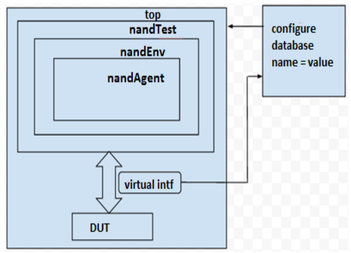
Reset: Reset command is used to reset all memory locations and start the transaction from known state[2].
Erase: Erase command is used to erase memory location. Before every particular block should be erase. Erase happens at block level.[2].
Program Page: It happens at the page level. Programming is done sequentially in NAND controller. It writes randomly to some locations in the same page after writing page.2].
Page Read: Page read also happens at page level. Data can be read randomly from the same page after reading from it sequentially [2].
Verification environment sends randomized input data and Address for performing operations. Constraints are applied to address so it can not exceed limitations of memory and controller. Sequence class responsible for generating different test cases [5]. Here are different test cases implemented in the verification environment, to check the performance of NAND flash controller in each scenario.

Sr. No	Sequence Name	Address Constraint	Input Data Constraint	Write Enable Constraint	Description
1	Write	Random	Random	Logic 1	random data is written to page in a random block
2	Read after write	Constrained to access valid bank, row and columns	Random	Write = Logic 1 Read = Logic 0	random data is written to page in a random block stores address of each location reads data present at the above locations.
3	Block write	Random	Random Writes 128 pages	Write = Logic 1 Read = Logic 0	writes one complete block at one time
4	Write after write	Write same page after erasing block	Random	Write = Logic 1 Read = Logic 0	random data is written to page in a random block

Performance is monitored for above different cases in terms of throughput and latency i.e. number of cycles taken for completing the instruction in each scenario .

Results

To verify the performance of the NAND Flash controller, the verification environment is designed. Using the UVM methodology the verification environment is designed and various test cases are tested on the NAND Flash controller. The performance of the controller is verified by counting the total number of cycles required by the controller for individual read, write operation and when burst was sent to the controller.
The below verification environment was implemented to verify the performance and below results were obtained

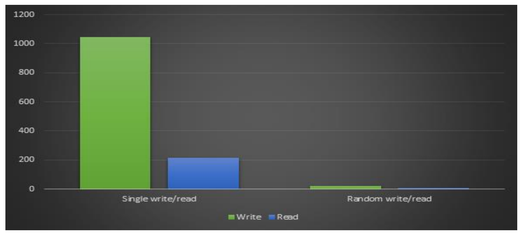


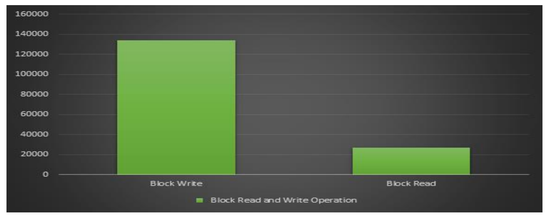
The performance monitor is designed to monitor the performance of the NAND Flash controller and it monitor the number of cycles taken by the NAND flash controller during burst write and burst read operation. It also monitors the total number of cycles taken by the controller for individual read and the page program operation.

Below table represents the cycle count of different operations

Sr No	Transaction Type	Data Size	Length	Time Taken
1	Single write	4314 bytes	1 page	1045.0 ns
2	Single read	4314 bytes	1 page	216.5 ns
3	Block Erase	512k+27k	1 block	2.9 ms
4	Random Write	12 bytes	1 page	20.9 ns
5	Block Write	512k+27k	128 pages	133760ns

Table No. Performance Analysis





The above graphs and tables represents the result of the various tests initiated on the controller by the UVM environment on the NAND Flash controller

Summary

Various test cases are generated to verify the performance of the NAND Flash controller according to the datasheet specifications. The timing parameter specified for all the operations have been successfully met by the UVM verification environment designed. The verification environment designed is reusable and any NAND Flash controller's performance can be verified using this environment. Since various tests cases can be designed and can be tested on the controller which signifies that the verification environment created is reusable

Key References

[1] Universal Verification Methodology (UVM) 1.1 Class Reference , www.accellera.org
[2] Intellectual Property Lattice Semiconductor-Reference Design (Reference Design RD1055
[3]Hillel Miller, Freescale, Thomas Alsop, Intel, Accellera, UVM-1.0 Reference Manual, 2011
[4] IEEE Std 1800-2009 "IEEE Standard for System Verilog-Unified Hardware Design, Specification, and Verification Language
[5] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigraphy.:- "Design Tradeoffs for SSD
[6] Performance." USENIX Annual Technical Conference. By Yangyang Pan, Guiqiang Dong, and Tong Exploiting Memory Device Wear-Out Dynamics to Improve NAND Flash Memory System Performance
[7] Samar Abdi, "Functional and Performance Verification of System Level Model Refinements", University of California Irvine

Acknowledgements

We thank professor Morris Jones for providing valuable information, laboratory space and equipment.
For further information: Please contact kriti.ashok@sjsu.edu, sayali.basole@sjsu.edu