2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Performance Analysis and Acceleration of High Bandwidth Memory System

Amitayu Banerjee, Senior Engineer, Samsung Semiconductor India Research, Bangalore, India (amitayu13.b@samsung.com)

Rohit Devidas Chavan, Associate Staff Engineer, Samsung Semiconductor India Research, Bangalore, India (chavan.rohit@samsung.com)

Jyoti Verma, Associate Director, Samsung Semiconductor India Research, Bangalore, India (jyoti.verma@samsung.com)

Sekhar Dangudubiyyam, Associate Director, Samsung Semiconductor India Research, Bangalore, India (sekhar.d@samsung.com)

***Abstract*- With truly expanding intricacy in SoC (System on Chip) plans, it is turning out to be increasingly more hard for design verification engineers to debug designs, measure execution and identify system bottlenecks in design. Performance measurement is a critical aspect of present day SoCs, which has high performing DDRs such as High Bandwidth Memory (HBM). HBM3 is a high-speed computer memory interface for 3D-stacked synchronous dynamic random-access memory (SDRAM). It is used in conjunction with high-performance graphics accelerators, network devices, high-performance data-center, AI ASICs and FPGAs and in some supercomputers.**

The Performance Analyzer is an intuitive graphical application that gives graphical and statistical reports of IPs performance alongside transactional level analysis of protocol related information. It provides a convenient method for estimating the performance of design utilizing protocol-related transaction information that has been recorded in a Fast Signal Database (FSDB). We have used Protocol Analyzer on HBM3 subsystem along with Performance Analyzer to get a perfect picture of protocol failures or violations affecting performance. After generating the reports, we can pinpoint bugs in design, understand protocol behavior and improve performance.

I. INTRODUCTION

As the complexity in chip design increases, the challenges in quality assurance also increases drastically. The quality of design verification is an essential part in any robust chip with a better customer acceptance. Nowadays for any SoC delivery, verification tasks are much more time consuming than design tasks as the latter involves reusing of IPs. Moreover, verifying IPs at its maximum performance potential at the SoC level becomes extremely critical due to inclusion of complex protocols and higher runtime. Any sort of protocol violation that directly impacts the system performance, if identified quickly can drastically reduce the efforts required.

In this paper, a new methodology of improvement has been introduced which enables quickly achieve the performance target numbers of the design in the most effective manner compared to existing methods where waveforms are manually analyzed for system performance. We have used Verdi's protocol analyzer, performance analyzer, and debugger window in a synchronized fashion which helps us to compare and find system bottlenecks easily. This helps in quicker evaluation and sign-off.

II. HBM3 SUBSYSTEM

The HBM3 DRAM is tightly coupled to the host compute die with a distributed interface. The interface is divided into independent channels. Each channel is completely independent of one another. Channels are not necessarily synchronous to each other. The HBM3 DRAM uses a wide-interface architecture to achieve high-speed, low power operation. Each channel interface maintains a 64-bit data bus operating at double data rate (DDR). Each HBM3 device has stacks of DRAM dies where each stack can support at max 16 channels.

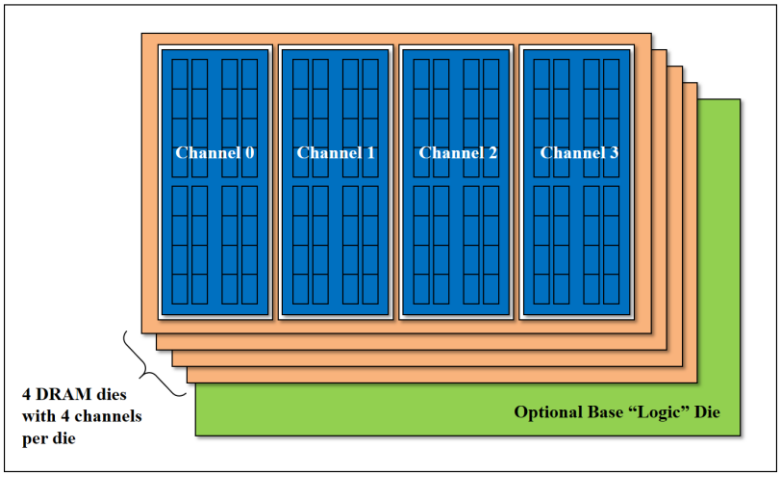

Fig 1: Overview of DRAM stack [1]

Fig 2. Represents our HBM3 subsystem which consists of 16 AXI4 interfaces which acts as a AXI Master for the 16 channel HBM3 model. We have integrated DFI monitor between PHY and memory to catch any protocol violation. Our subsystem is verified at 4800MHz, 5600MHz and 6400MHz with different density/stack configurations such as 16GB_8High, 24GB_12High, and 64GB_16High.

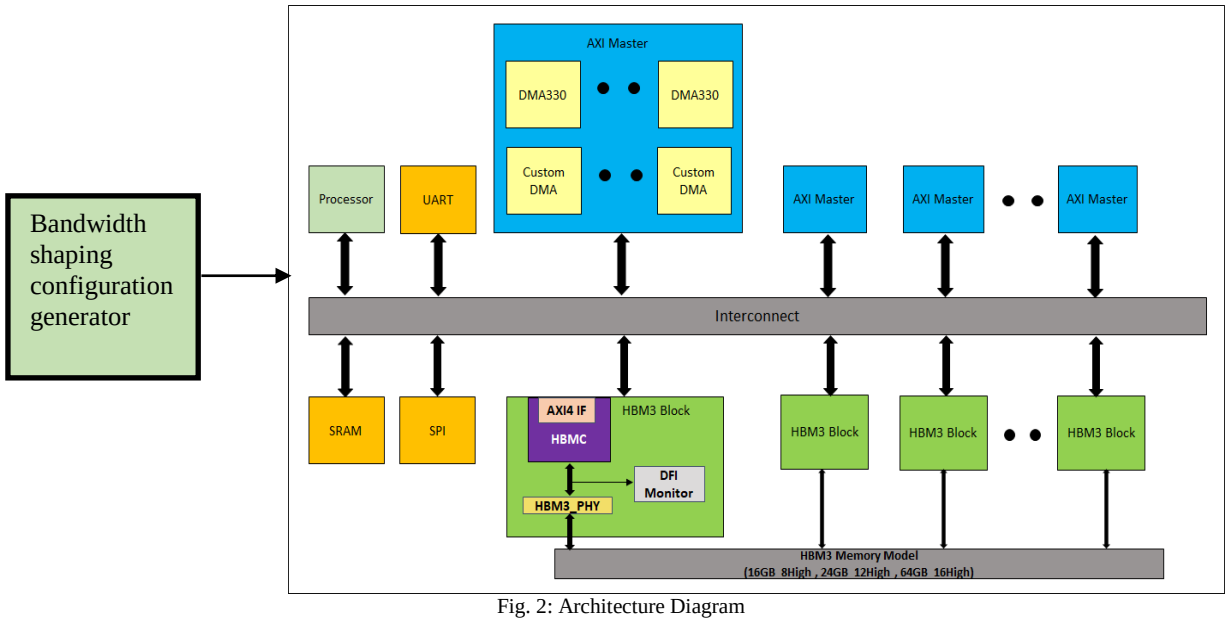

Fig. 2: Architecture Diagram

III. RELATED WORK

This paper discusses how the performance can be improved for HBM3 device, which is a highly critical aspect for modern day memory SoCs which find applications in AI, ML, data centres etc. The most important standardized parameters which we have used for the measurement of device performance are Bandwidth (Mbps), page hit count, page miss count and page empty count. A few other timing parameters that we have also considered are active_to_active_delay, active_to_write_delay, and active_to_read_delay.

A. Performance Analyzer

Performance analyzer [4] gives us a detailed report on the performance of our system by providing us data on different metrics and parameters. The graphical reports generated are easy to understand and analyze, and can also be ported to different exportable formats such as HTML and CSV. Thresholds can be set for each memory parameter where violations get highlighted.

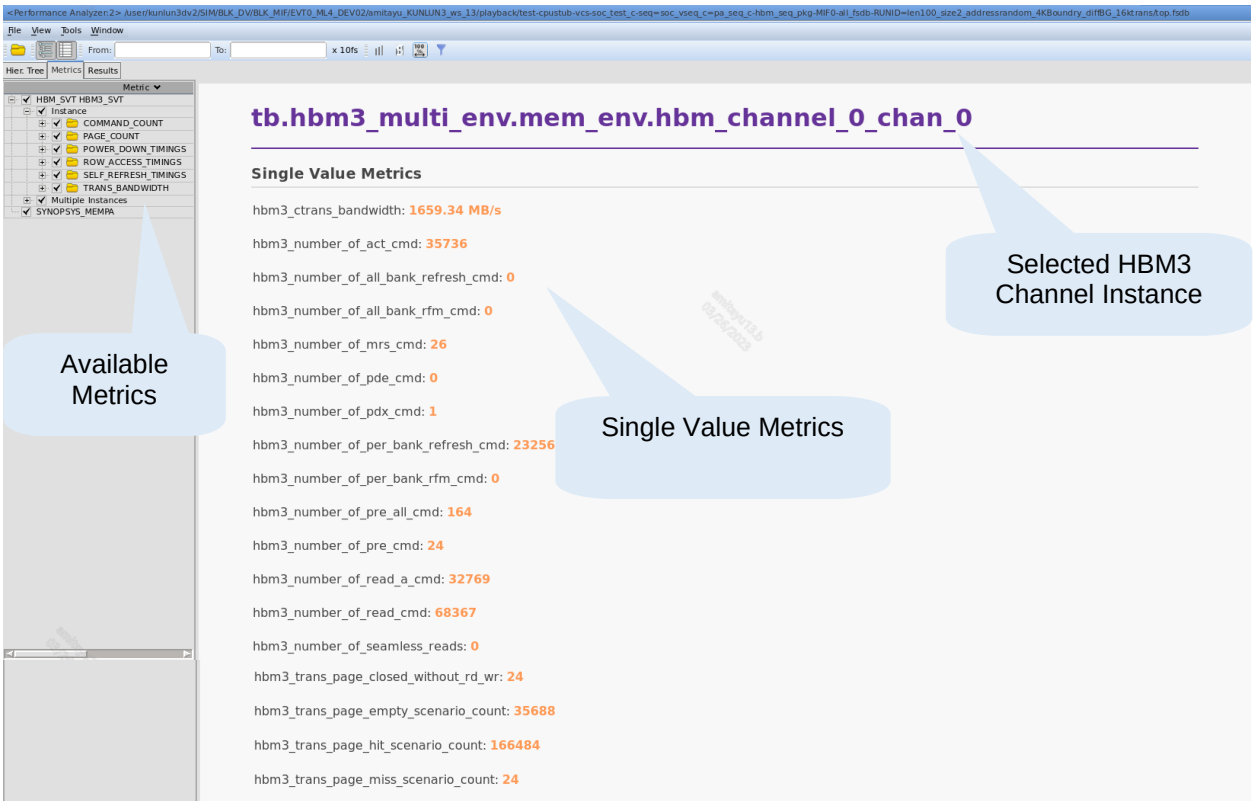

Fig 3: Metric view of Performance Analyzer

Fig. 3 provides metrics related to HBM3 memory such as bandwidth, page related info, number of read/writes issued etc. These metrics help in finding gaps in our traffic pattern which can be rectified to improve performance further.

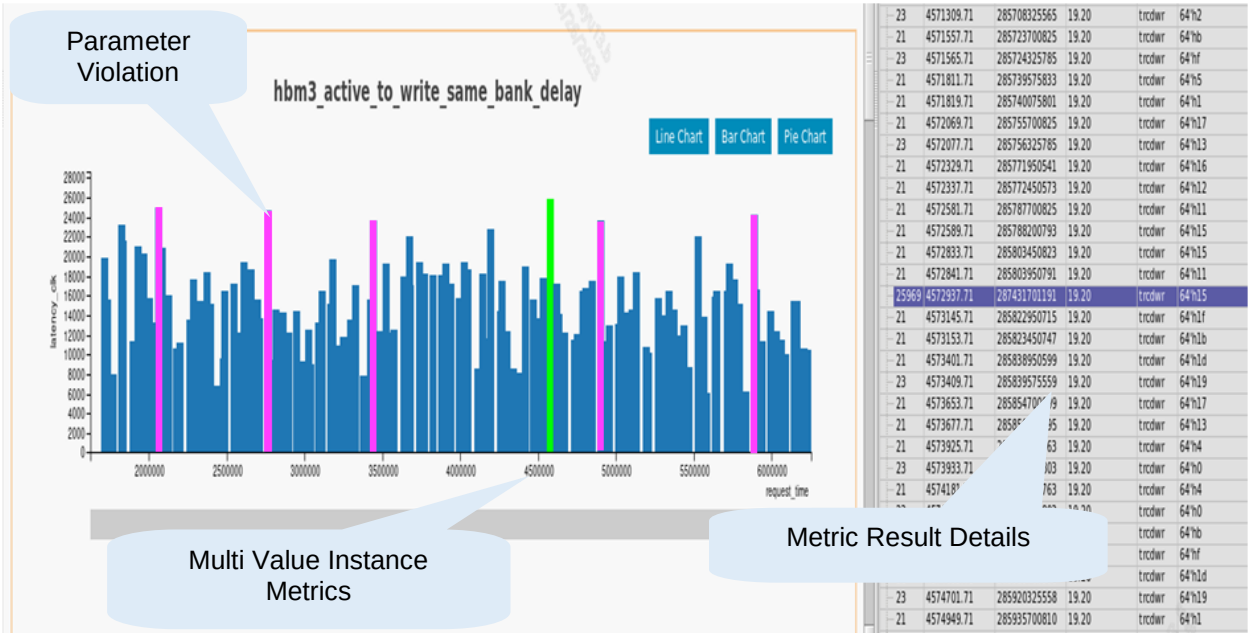

Fig. 4: Parameter graphical view

Fig. 4 shows the parameter graphical view of activate_to_write command delay and how that behaved over time. The "details" column shows the detailed transaction data for any selected bar of the graph. The violations with respect to threshold value if any would be highlighted in a different colour.

B. *Protocol Analyzer*

Protocol Analyzer [3] provides complete details of issued transactions such as command type, physical address etc. Filtering feature added to the transactional data by which engineers could only look at protocols and violations that are of interest to them. Synchronized protocol analyzer and smart log feature can be used in quick error detection wherein all the errors are back annotated to the specification.

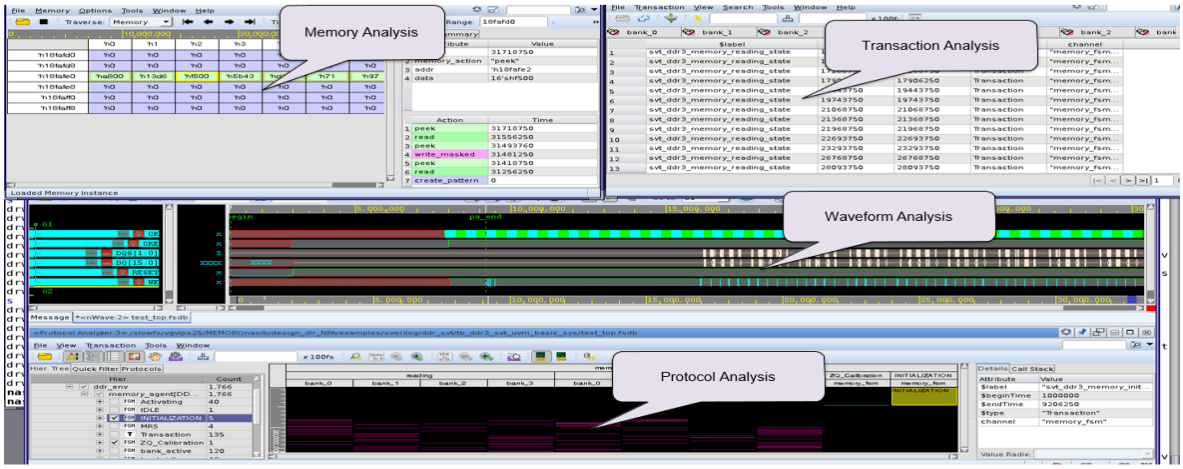

Fig. 5: Detailed view of Protocol Analyzer

Fig. 5 shows an example of memory array [2] which gives a pictorial view of data written/read to/from the memory along with their Error Correction Code (ECC) values. The smart log and protocol analyzer window shows us the details of each command issued to the memory. All the tabs shown in Fig. 5 are time synchronized which makes debug easy.

C. *Synchronization of Performance Analyzer and Protocol Analyzer*

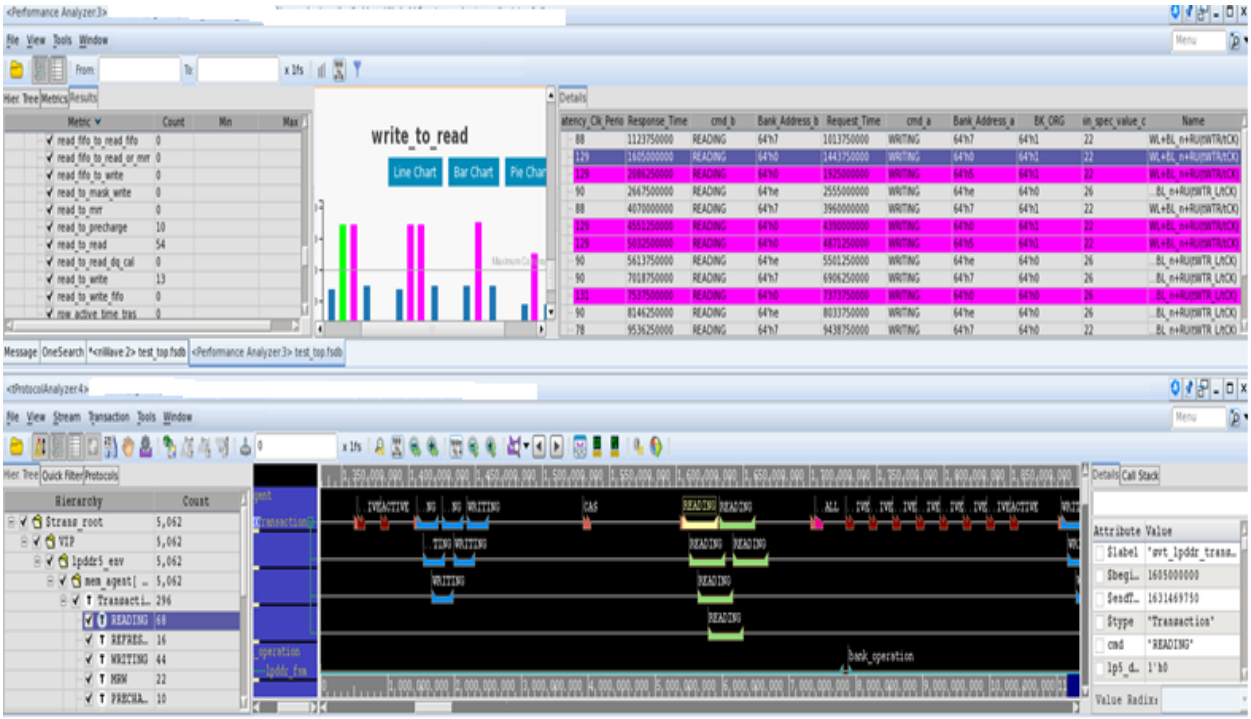

Fig. 6 Synchronized view of Protocol and Performance Analyzer

Fig. 6 shows the synchronized view of protocol and performance analyzer. Pink colored rows/bars show that the matric number for that transactions are more than threshold set and those can easily be analyzed using protocol analyzer window.

D. *Bandwidth Shaping*

In addition to performance measurement, we have integrated a Bandwidth shaping configuration generator as shown in Fig. 2 to our testbench (TB), which has helped us to significantly improve our device performance. We have tested this with different operating frequencies such as 4800MHz, 5600MHz and 6400MHz and with different density/stack configurations such as 16GB_8High, 24GB_12High, and 64GB_16High.

Bandwidth shaping configuration generator is a TB component wherein we use different mapping schemes between AXI interface and of memory interface (Bank Group [BA] /Row [RA] /Column [CA] /Stack ID [SID]).

1	AXI [29:16]	AXI [15:12]	AXI [11]	AXI [10:6]		
	RA [13:0]	{BA3,BA2, BA1, BA0 }	SID	CA [4:0]		
2	AXI [29:16]	AXI [15]	AXI [14:11]	AXI [10:6]		
	RA[13:0]	SID	{BA3, BA2, BA1, BA0}	CA [4:0]		
3	AXI [29:16]	AXI [15]	AXI [14:12]	AXI [11:9]	AXI [8]	AXI [7:6]
	RA[13:0]	SID	{ BA3, BA1, BA0 }	CA [4:2]	BA2	CA [1:0]

Table 1: Address Mapping

Table 1 shows different mapping schemes that have been verified. Case 1 is the worst case scenario where column bits will frequently change and accesses will be going to the same bank group resulting in low bandwidth. Case 3, is the best case scenario where every 256-bit write/read transaction goes to a different bank group (BG) resulting in the best performance numbers.

IV. EFFORT TABULATION

		Total Turnaround Time / Effort*	
Category	**Highlights**	**Without Protocol Analyzer**	**With Protocol Analyzer**
HBM3 Initialization	Protocol analyzer helped us in resolving tINIT related parameter failures.	~ 47-51	< 7
IEEE 1500 inteface bring-up	Proper release of RESETn and WRSTn. Violations while issuing IEEE 1500 instructions.	~ 8-10	2-3
Bandwidth (BW) Improvement	Protocol analyzer helped us find scenarios to improve performance of our subsystem. We tweaked the row, column, bank and bank group mapping to achieve max BW.	~ 20	~ 5

Table 2: Turn Around Time and Effort Table

* All values are in term of number of iteration

V. RESULTS

With the help of performance and protocol analyzer we could identify few gaps in our test scenario and after selecting the optimal bank, row, column mappings w.r.t previous sent data, we could see a significant improvement in performance. Table 3 tabulates the performance v/s scenario details.

Metrics / Scenario	Bandwidth (MB/s)	Page hit scenario count	Page empty scenario count	Page miss scenario count
Case 1: 400 KB transactions without BG interleaving.	453.14	15,615	5,243	6
Case 2: 400 KB transactions with BG interleaving	1,103.95	1,66,484	35,688	24
Case 3: 6.4 MB transactions with BG interleaving	1,659.34	1,90,550	85,277	97

Table 3: Bandwidth Shaping

For a low traffic random transaction without BG interleaving (Case 1) the performance of the system is very low. But, when we changed the address mapping configuration to support BG interleave (Case 2) we could see a significant improvement of 143.62% in our bandwidth. In our best performing scenario (Case 3), where we have sent data to different bank groups, with increased data size, we could achieve a much higher bandwidth of 1.659 GB/s for pseudo channel 0 of HBM channel 0.

VI. CONCLUSION

In this paper, a new methodology of improvement has been introduced which enabled us to quickly achieve the performance target numbers of a bandwidth critical system like High Bandwidth Memory (HBM3) in the most effective manner compared to existing methods where waveforms are manually analyzed for system performance. In addition to this, we have used protocol analyzer to rectify the violations which increase latency in the system. Hence, helping us boost up the performance numbers quickly.

REFERENCES

[1] JEDEC HBM3 specification: https://www.jedec.org/standards-documents/docs/jesd238a
[2] Verdi Memory Array User Guide: https://spdocs.synopsys.com/dow_retrieve/qsc-u/vg/verdi/U-2023.03/verdi_olh/index.htm#page/Verdi_Transaction_and_Protocol_Debug_User_Guide/MemoryPA_MemoryMode.05.5.htm
[3] Verdi Transaction and Protocol User Guide: https://spdocs.synopsys.com/dow_retrieve/qsc-t/vg/verdi/T-2022.06/PDFs/Verdi_Transaction_and_Protocol_Debug.pdf
[4] Verdi Performance Analyzer User Guide: https://spdocs.synopsys.com/dow_retrieve/qsc-t/vg/verdi/T-2022.06/Verdi_olh/index.htm#page/Verdi_Performance_Analyzer_User_Guide/Use_Model.3.01.htm1