Breaking barriers in Advanced Multi-Chiplet AI SoCs using scalable UCIe and Boot Verification and Emulation techniques

Harshal Kothari, Samsung Semiconductor India Research, Bengaluru, India
Eldin Ben Jacob, Samsung Semiconductor India Research, Bengaluru, India
Ayush Agrawal, Samsung Semiconductor India Research, Bengaluru, India
Vaishali Sahu, Samsung Semiconductor India Research, Bengaluru, India
Jerin M Jose, Samsung Semiconductor India Research, Bengaluru, India
Jasobanta Sahoo, Samsung Semiconductor India Research, Bengaluru, India
Madhukar Ramegowda, Samsung Semiconductor India Research, Bengaluru, India

Abstract—**The need for complex computing networks are on the rise and this calls for more advanced logic incorporated on a smaller form factor. Size of monolithic SoCs for Generative AI, hyperscalers & enterprise grade data-centre applications is becoming too big for manufacturability. With increasing complexity and number of gates, simulation time is also increasing and the adoption of next-gen verification techniques are on the rise. With the emergence of complex multi-die SoC designs to tackle yield issues on advanced manufacturing nodes, the challenges in verification have increased manifold. This involves integration of pre-verified IPs, sub-systems, and partially verified chiplets using multiple vendor simulator platforms and Verification IPs. Artificial Intelligence (AI) is at the forefront of ASIC breakthroughs. With time to market being a critical factor, adherence to aggressive schedules has become the new normal. Rebuilding these complex verification environments onto a multi-die SoC testbench in the given timelines is extremely challenging. Universal Chiplet Interconnect Express (UCIe) is an open industry standard, multi-protocol, high-bandwidth (up to 64GT/s per lane), die-to-die (chiplet) interconnect that standardizes inter-die communication on-package. With introduction of multichiplet architecture, the requirement to develop a single and multichiplet testbench is imperative. Rebuilding such an environment is challenging and requires a lot of time, which calls for a scalable technique where the single chiplet testbench environment can be reused for the development of a multichiplet environment, which is done as part of the distributed simulation and ndie simulation technology. When compared to traditional DUT back-to-back setup in a single die testbench, distributed simulation and ndie simulation setup has drastically reduced the overall testbench development time and also helped in achieving 3x improvement in simulation speed and flexibility to make this solution ubiquitous for chiplet based architectures. The time of simulation for any SoC block with initialization process that extends to multiple millisecond range additionally calls for the adoption of emulation techniques for design verification acceleration. The multichiplet boot and the initialization takes >100ms and the simulation environment takes weeks to run tests even with hybrid environment. The adoption of emulation and the development of emulation environment for verification helps validate the multichiplet scenarios in hours reducing the total turnaround time by over 1000 times. This also potentially makes use of licenses in most effective way and results in reduction of both development cycle time and saves the precious license cost during design verification of of 2.5DIC for High Performance Compute (HPC) generative AI inference multi-chiplet SOC to make this solution ubiquitous for chiplet based architectures.**

I. INTRODUCTION

Size of monolithic SoCs is becoming too big for manufacturability on advanced technology nodes. Therefore the current SoC consists of 4 homogeneous 4nm chiplets with size 350mm2 per die (total 11.2B gate count) on a single interposer communicating with each other via UCIe 1.1 protocol, QSPI and GPIOs. It is primarily targeted at high-bandwidth, low-power and low-latency generative AI applications for High Performance Computing (HPC) product. First chiplet acts as the primary director die and the others are secondary chiplets. The boot process is different in each chiplet based on the reset release and source of booting of each processor. Simulation of a large design always has a high demand for the IT infrastructure and cost. The above limitation are removed when the simulations of these are performed in new age tools such as distributed simulation or ndie solutions where the single chiplet environment has been reused to create the primary chiplet and the secondary chiplet environments. In this case the IT infrastructure required for the single die simulations and the multi die simulations are the same.

The complexity and the runtime of real use case based simulations cannot be performed on the simulators as the real world use cases involved Secure Boot Processors powered by RISC-V core and QSPI which is a low speed protocol for the transfer of boot code data from primary die to secondary dies spanning over 100ms amounting to weeks of run times with RTL dynamic simulations. So the multidie environment is also developed in emulation environment with a significant reduction in the run times. The major challenges in setting up a multidie UCIe environment with such complex digital logic is number of gates supported on the emulators. The current capacity of the system supports only 1.25 chiplets and the environment should contain minimum of 2 chiplets to develop and validate the UCIe real world use cases and the boot scenarios. This led to the adoption of hybrid testbench environment. The emulation environment also faced many challenges to bring up the processors and verifying real AI workload transfer across the chiplets over UCIE which was overcome by the hardware acceleration techniques.

The boot scenarios consist of majorly 3 boot techniques 1) RAM based boot 2) Flash based boot and 3) OTP based boot. These were performed on distributed simulation and emulation platforms which was a key requirement to reduce the turnaround time required to validate the multi-die use cases in the real world environment. UCIe consists of 3 layers: logical and electrical physical layer (PHY), die-to-die adapter and protocol layer. The UCIe PHY enables multi-die system in package integration for high performance compute, AI/ML, 5G, automotive and networking applications. Physical layer includes multi-module PHY support with 2 instances which perform link initialization, training, power management states, lane mapping, lane reversal, and scrambling. The UCIe controller includes the die-to-die adapter layer and the protocol layer. The adapter layer ensures reliable transfer through link state management and parameter negotiation of the protocol and flit formats. The UCIe architecture supports multiple standard protocols such as PCIe, CXL and streaming mode. Streaming mode with AMBA AXI4 protocol is the primary use case of this paper with 256B latency optimized UCIe flit formats. There are 6 blocks of UCIe-A subsystem and 2 blocks of 8-core NPU cluster per die which need to be verified concurrently across 4 dies. Each master shares 144GB of HBM3E (9.6Gbps per lane) non-coherently.

II. Application

Real interchiplet AI traffic workloads were generated from the neural cores and 2 testbenches were set up: DUT back-to-back and DUT-VIP. The need to verify the functionality of multiple multi-billion gate count chiplets was achieved through 2 competing EDA solutions from industry leading vendors: Distributed Simulation and Ndie solution. It enabled simulating 4 dies in parallel with dedicated executables on different machines where functional datapath with Link Training State Machine (LTSM) bringup between UCIe link partners on 2 dies ensues, followed by inter-chiplet AXI HBM and MMIO transfers from real AI and CPU workloads.

Standardization was achieved by keeping a common testbench skeleton. As the SoC design is huge (11.2B gate count), multi-step incremental elaboration flow is adopted to ensure any change in tests or sequences does not necessitate re-elaboration of Design Under Test (DUT). The testbench generation is automated based on excel input spec. It has same look and feel on various verification scopes (IP/Sub-system/SoC) (Figure 1). This also bolsters the reusability aspect of an IP or sub-system verification environment at SoC level and vice-versa.

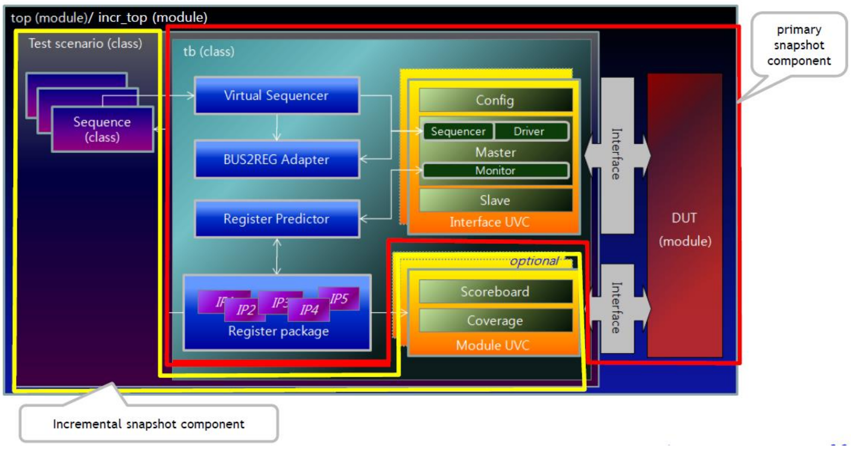


Figure 1 Multi snapshot single chiplet testbench

Verifying multi-chiplet boot, UCIe link bringup and interchiplet data transfer functionality and NPU AI workload transfer across the dies were the major applications targeted using distributed simulations. The end-to-end product use cases combining these scenarios led to 300+ hours run times with extremely high memory utilization. The bootloader makes use of CHIPID to load the input parameters for each chiplet. The UCIe 1.1 channels are the primary interconnect for chip-to-chip transfer to share the workload. Low speed IO is connected on the interposer between the chiplets to transfer the UCIe boot code and help secondary chiplets to boot-up. To handle the SoC level data flow for ML workloads and secure booting, 4 CPUs are used in each chiplet.

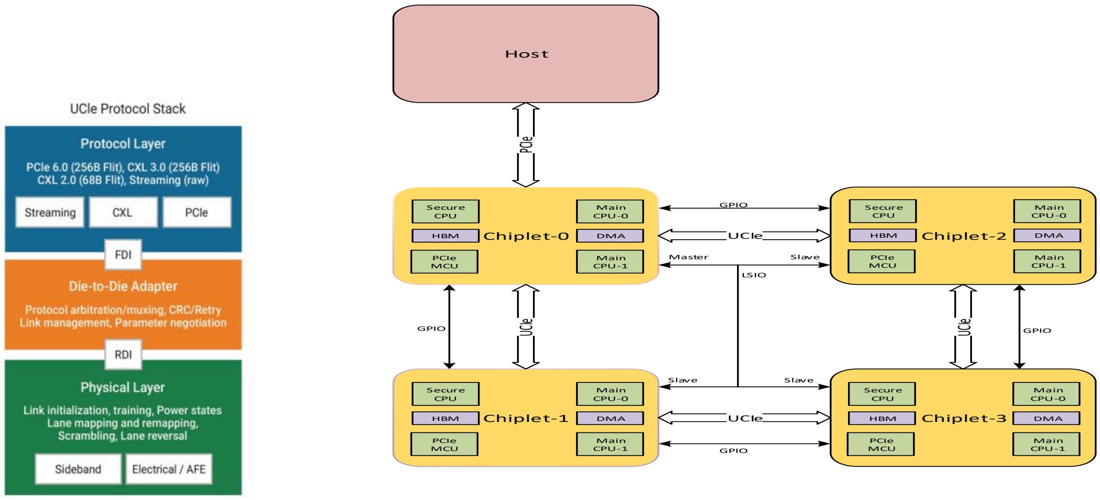

UCIe Protocol Stack
Protocol Layer
PCIe 6.0 (256B Flit), CXL 3.0 (256B Flit) CXL 2.0 (68B Flit), Streaming (raw)
Streaming
CXL
PCIe
FDI
Die-to-Die Adapter
Protocol arbitration/muxing, CRC/Retry Link management, Parameter negotiation
RDI
Physical Layer
Link initialization, training, Power states Lane mapping and remapping, Scrambling, Lane reversal
Sideband
Electrical / AFE
Host
PCIe
Secure CPU
Main CPU-0
HBM
Chiplet-0
DMA
PCIe MCU
Main CPU-1
GPIO
UCIe
Master
Slave
LSIO
Chiplet-2
Chiplet-1
Chiplet-3
Slave
Slave
GPIO

Figure 2 Multi-Chiplet 2.5DIC SoC Architecture

In secure boot mode, the system operation starts from the secure processor subsystem after the initial power is applied and reset is released. The secure boot has three stages FBOOT, SBOOT and TBOOT, each stage bootloader of secure processor subsystem checks the integrity of the next stage bootloader. After release of SoC reset, secure processor performs memory integrity check during FBOOT stage and then it verifies, loads and transitions to SBOOT stage. During SBOOT and TBOOT stages DRAM is initialized, LSIO is setup to transfer boot code to secondary chiplet, PCIe/UCIe are initialized to establish link between host <-> primary chiplet <-> secondary chiplets and CPU subsystem is initialized to prepare for boot up of secondary chiplets.

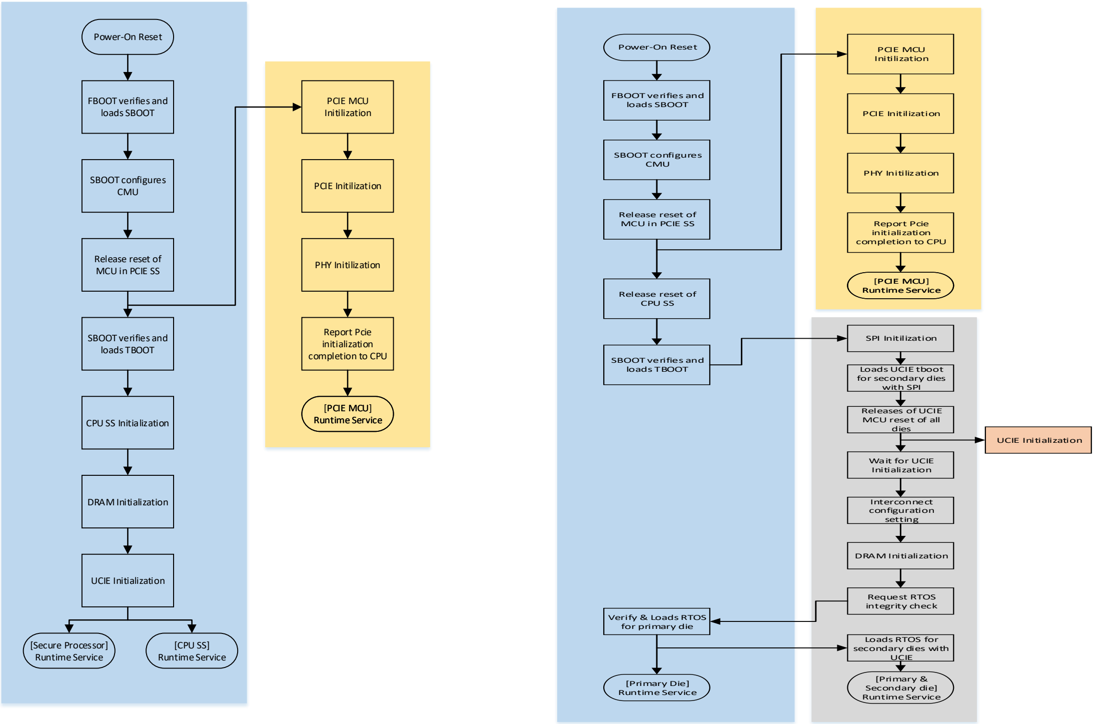
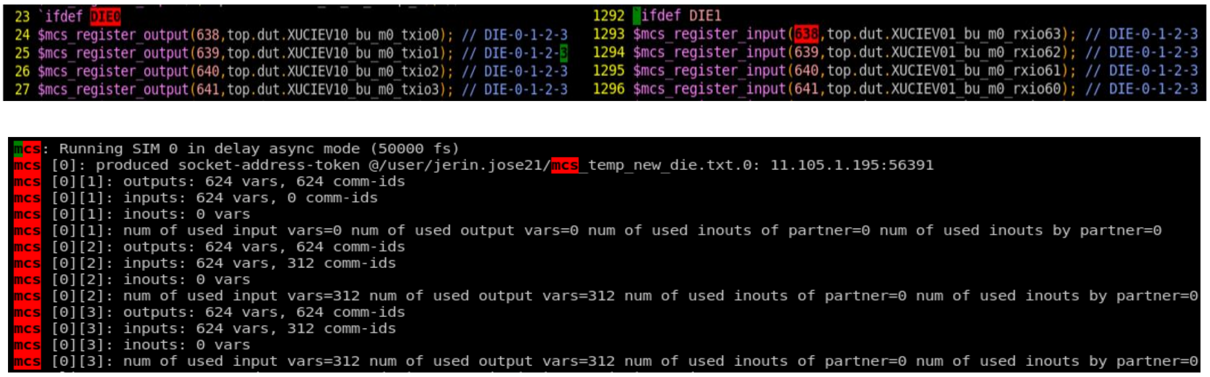

Power-On Reset
FBOOT verifies and loads SBOOT
SBOOT configures CMU
Release reset of MCU in PCIE SS
SBOOT verifies and loads TBOOT
CPU SS Initialization
DRAM Initialization
UCIE Initialization
[Secure Processor] Runtime Service
[CPU SS] Runtime Service
PCIE MCU Initilization
PCIE Initilization
PHY Initilization
Report Pcie initialization completion to CPU
[PCIE MCU] Runtime Service
Power-On Reset
FBOOT verifies and loads SBOOT
SBOOT configures CMU
Release reset of MCU in PCIE SS
Release reset of CPU SS
SBOOT verifies and loads TBOOT
Verify & Loads RTOS for primary die
[Primary Die] Runtime Service
PCIE MCU Initilization
PCIE Initilization
PHY Initilization
Report Pcie initialization completion to CPU
[PCIE MCU] Runtime Service
SPI Initilization
Loads UCIE tboot for secondary dies with SPI
Releases of UCIE MCU reset of all dies
UCIE Initialization
Wait for UCIE Initialization
Interconnect configuration setting
DRAM Initialization
Request RTOS integrity check
Loads RTOS for secondary dies with UCIE
[Primary & Secondary die] Runtime Service

Figure 3 Single and Multi-Chiplet Boot Flow

In case of 4 chiplet boot, the primary and secondary dies perform boot steps(i.e., fboot, sboot) in parallel. The neural engines in both blocks are then loaded with predefined vectors from the customer. The framework to start the processed generative AI dataloads is incorporated into simulation and emulation based design verification acceleration flow. Once all the 8 UCIe Link Training and State Machine (LTSM) linkups are performed across 16 UCIe instances with their respective chiplet partners, the AXI interchiplet transfers begins from Neural Processing Unit (NPU) data cluster cores towards High Bandwidth Memory (HBM3E). It performs generative AI training across 175 billion parameters and accesses the empty memory from interchiplet HBM3E as well as shared memory and scratch memory across the 4 dies as per the real usage scenarios.



```
`ifdef DIE0
$mcs_register_output(638,top.dut.XUCIEV10_bu_m0_txio0); // DIE-0-1-2-3
$mcs_register_output(639,top.dut.XUCIEV10_bu_m0_txio1); // DIE-0-1-2-3
$mcs_register_output(640,top.dut.XUCIEV10_bu_m0_txio2); // DIE-0-1-2-3
$mcs_register_output(641,top.dut.XUCIEV10_bu_m0_txio3); // DIE-0-1-2-3
```

```
`ifdef DIE1
$mcs_register_input(638,top.dut.XUCIEV01_bu_m0_rxio63); // DIE-0-1-2-3
$mcs_register_input(639,top.dut.XUCIEV01_bu_m0_rxio62); // DIE-0-1-2-3
$mcs_register_input(640,top.dut.XUCIEV01_bu_m0_rxio61); // DIE-0-1-2-3
$mcs_register_input(641,top.dut.XUCIEV01_bu_m0_rxio60); // DIE-0-1-2-3
```

```
mcs: Running SIM 0 in delay async mode (50000 fs)
mcs [0]: produced socket-address-token @/user/jerin.jose21/mcs_temp_new_die.txt.0: 11.105.1.195:56391
mcs [0][1]: outputs: 624 vars, 624 comm-ids
mcs [0][1]: inputs: 624 vars, 0 comm-ids
mcs [0][1]: inouts: 0 vars
mcs [0][1]: num of used input vars=0 num of used output vars=0 num of used inouts of partner=0 num of used inouts by partner=0
mcs [0][2]: outputs: 624 vars, 624 comm-ids
mcs [0][2]: inputs: 624 vars, 312 comm-ids
mcs [0][2]: inouts: 0 vars
mcs [0][2]: num of used input vars=312 num of used output vars=312 num of used inouts of partner=0 num of used inouts by partner=0
mcs [0][3]: outputs: 624 vars, 624 comm-ids
mcs [0][3]: inputs: 624 vars, 312 comm-ids
mcs [0][3]: inouts: 0 vars
mcs [0][3]: num of used input vars=312 num of used output vars=312 num of used inouts of partner=0 num of used inouts by partner=0
```

Figure 4 Chiplet IO bump connections

All 4 dies runs a separate simulation thread on a different LSF machine thereby optimizing the memory requirements to simulate the whole quad-chiplet package together. Across-the-die communication is achieved through LSF Ethernet socket based on the configuration file which is passed at elaboration stage as per Figure 4. The run time socket address information distribution across the simulations is automated. This was started with 2 dies and later extended to 4 dies with no theoretical upper-cap. With simulation across 2 or more dies getting executed in parallel on separate machines, it helps utilize the resources better effectively eliminating the need for bigger memory or compute machines which are 11x costlier than normal capacity machines.

III. Results

Systems depend on a tight coupling of the dies, which requires verification specifically targeted at complex datapaths that span the multiple dies and performance that is a function of the multi-die interconnect. With a complex, high-speed (2Tbps bidirectional bandwidth) multi-layered IP like UCIe, it was able to leverage the benefits offered by distributed simulation and ndie setup to a great extent. As shown in Figure 5, DV engineers can combine independent designs into one simulation environment and enable synchronization and testbench communication with cooperating executables. Since most of the multidie env have same design/TB for most of the dies, a simple config file can help the user connect the 2 env together without having to generate a top wrapper testbench or DUT component. Instead, the top multi-die interposer wrapper which performs the across-the-die UCIe serial IO bump connections mimicking the final interposer design aggressively helped in left shifting the multi-die testbench development 12 weeks before interposer implementation was finalized.

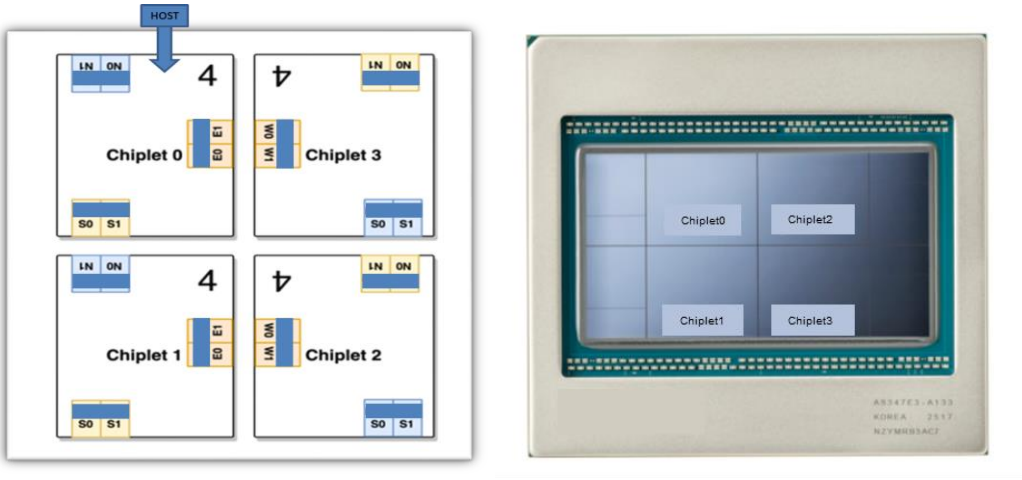


Figure 5 4-Chiplet Floorplan and common TB alongside final package

Once all the basic features are verified in single chiplet, the verification scope has been extended to multi chiplet verification. Since four chiplets are homogenous and logically the same with each other, common address remap of each chiplet having the unique address map depending on the chiplet ID is architected. 40 bit memory map is defined which splits 128GB region for each of the 4 chiplets and 512GB for the global host. 2 bit mode_direction (north/south/east/west) and 1 bit mode_host is used to determine the direction where the address type has to be routed based on the CHIPID value driven by TB. (Figure 6)

This framework can be easily extended with industry leading EDA simulation tools across any UCIe based chiplet design for AI/HPC/application processor/automative/industrial applications.

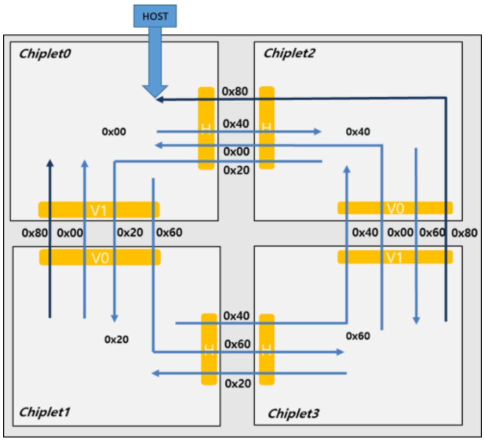


Figure 6 4-Chiplet global address map routing

The Multi-chiplet full chip verification included fruitful experiments across distributed simulation with Tool 1 and ndie simulation with Tool 2 with different testbenches (UCIe DUT-VIP v/s DUT-DUT back-to-back) save and restart, hierarchical reference permissions, read/write/connectivity debug access permissions, simulator performance switches, wave dump options, LSF and compute memory optimization et al which will be discussed in detail in final paper. We also faced roadblocks with respect to Verilog and System Verilog LRM and UPF compliance design and testbench code which was modified on-the-go as well as tool crashes which was rectified.

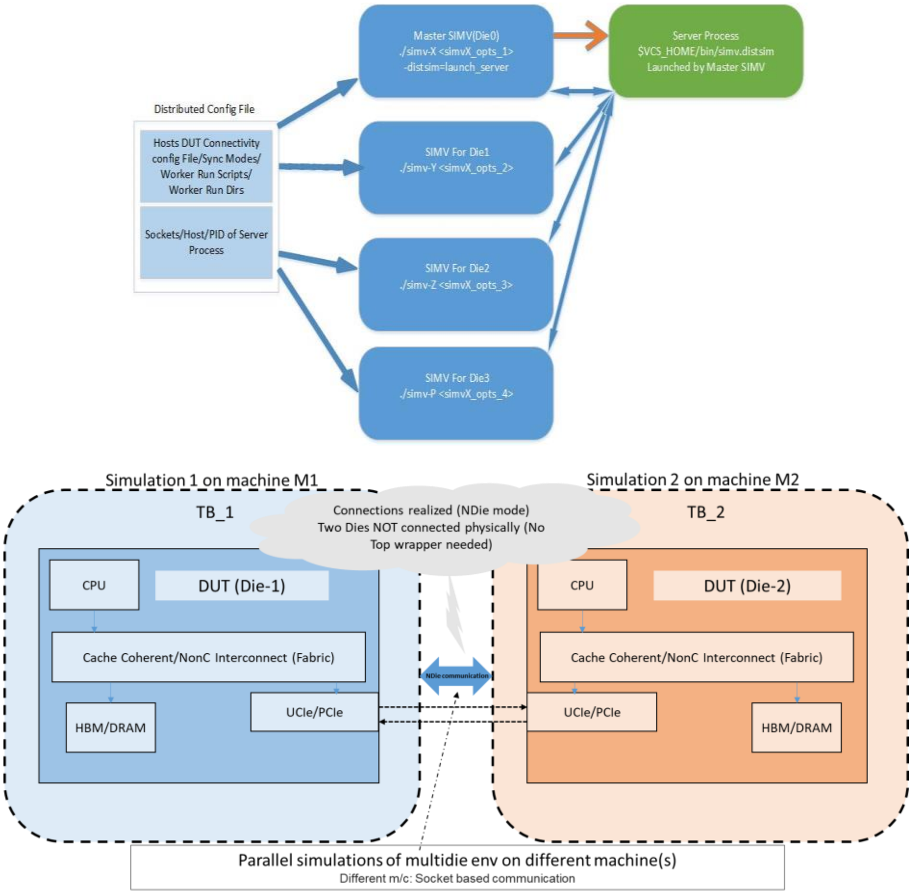
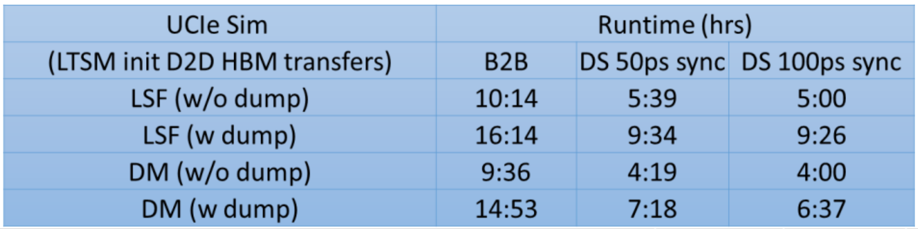
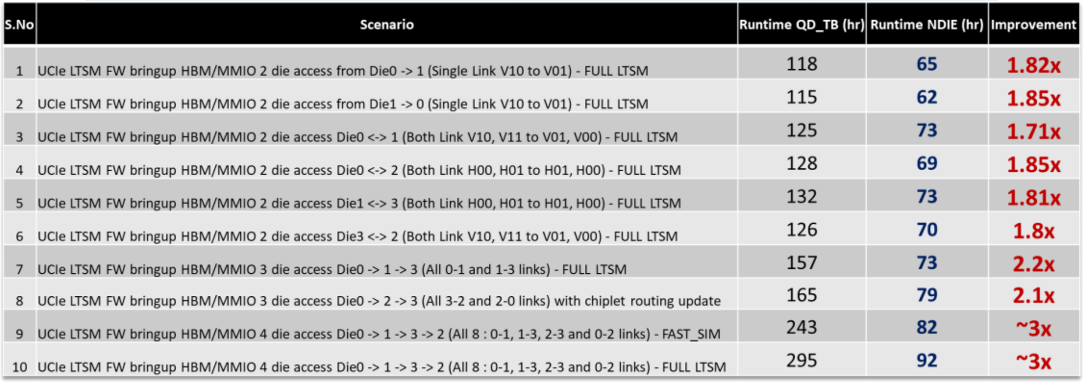

Master SIMV(Die0)
./simv-X <simvX_opts_1>
-distsim=launch_server
Server Process
$VCS_HOME/bin/simv.distsim
Launched by Master SIMV
Distributed Config File
Hosts DUT Connectivity config File/Sync Modes/ Worker Run Scripts/ Worker Run Dirs
Sockets/Host/PID of Server Process
SIMV For Die1
./simv-Y <simvX_opts_2>
SIMV For Die2
./simv-Z <simvX_opts_3>
SIMV For Die3
./simv-P <simvX_opts_4>
Simulation 1 on machine M1
Simulation 2 on machine M2
TB_1
TB_2
Connections realized (NDie mode)
Two Dies NOT connected physically (No
Top wrapper needed)
CPU
DUT (Die-1)
Cache Coherent/NonC Interconnect (Fabric)
HBM/DRAM
UCIe/PCIe
NDie communication
CPU
DUT (Die-2)
Cache Coherent/NonC Interconnect (Fabric)
UCIe/PCIe
HBM/DRAM
Parallel simulations of multidie env on different machine(s)
Different m/c: Socket based communication

Figure 7 Ndie/Distributed Simulation overall use model

When different simulations are run on different LSF machines, the simulation threads synchronize through ethernet socket based communication across LSF farm. There is a fixed asynchronous delay specified by user at run time using which the simulations run on different machines synchronize only at specified intervals rather than communicating at every simulation timestamp causing unintentional lockstep mode overhead. Modifying the asynchronous delay at LSF socket communication across the simulations on 4 die to an optimum 50ps value also helped alleviate the run times and further bolster the n-die distributed simulation performance. This reduced the need for lockstep mode where synchronization between the 4 runs happens at every simulation event.



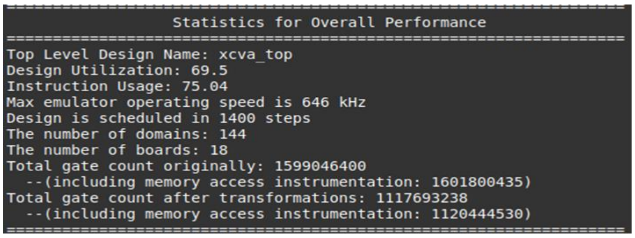
UCIe Sim	Runtime (hrs)		
(LTSM init D2D HBM transfers)	B2B	DS 50ps sync	DS 100ps sync
LSF (w/o dump)	10:14	5:39	5:00
LSF (w dump)	16:14	9:34	9:26
DM (w/o dump)	9:36	4:19	4:00
DM (w dump)	14:53	7:18	6:37



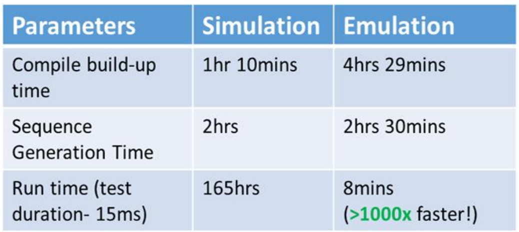
S.No	Scenario	Runtime QD_TB (hr)	Runtime NDIE (hr)	Improvement
1	UCIe LTSM FW bringup HBM/MMIO 2 die access from Die0 -> 1 (Single Link V10 to V01) - FULL LTSM	118	65	1.82x
2	UCIe LTSM FW bringup HBM/MMIO 2 die access from Die1 -> 0 (Single Link V10 to V01) - FULL LTSM	115	62	1.85x
3	UCIe LTSM FW bringup HBM/MMIO 2 die access Die0 <-> 1 (Both Link V10, V11 to V01, V00) - FULL LTSM	125	73	1.71x
4	UCIe LTSM FW bringup HBM/MMIO 2 die access Die0 <-> 2 (Both Link H00, H01 to H01, H00) - FULL LTSM	128	69	1.85x
5	UCIe LTSM FW bringup HBM/MMIO 2 die access Die1 <-> 3 (Both Link H00, H01 to H01, H00) - FULL LTSM	132	73	1.81x
6	UCIe LTSM FW bringup HBM/MMIO 2 die access Die3 <-> 2 (Both Link V10, V11 to V01, V00) - FULL LTSM	126	70	1.8x
7	UCIe LTSM FW bringup HBM/MMIO 3 die access Die0 -> 1 -> 3 (All 0-1 and 1-3 links) - FULL LTSM	157	73	2.2x
8	UCIe LTSM FW bringup HBM/MMIO 3 die access Die0 -> 2 -> 3 (All 3-2 and 2-0 links) with chiplet routing update	165	79	2.1x
9	UCIe LTSM FW bringup HBM/MMIO 4 die access Die0 -> 1 -> 3 -> 2 (All 8 : 0-1, 1-3, 2-3 and 0-2 links) - FAST_SIM	243	82	~3x
10	UCIe LTSM FW bringup HBM/MMIO 4 die access Die0 -> 1 -> 3 -> 2 (All 8 : 0-1, 1-3, 2-3 and 0-2 links) - FULL LTSM	295	92	~3x

Table 1: Distributed sim run times

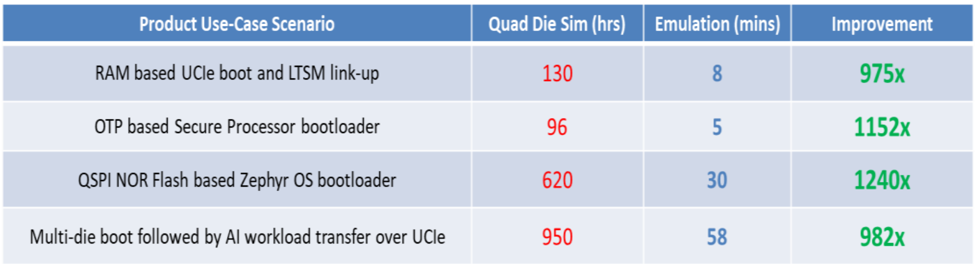
Emulation setup further helped ameliorate the humongous boot run times by 1000 times. The design environment was developed with dual die DUT, top testbench and clock generation logic. Compilation followed for front-end synthesis and back-end compilation and place & route for emulator. At run time, image is loaded into emulator and after loading the secure processor's boot binary into memory via backdoor, reset to DUT is released. Sanity tests for secure boot, UCIe linkup and AI interchiplet workload transfer between NPU cores and HBM/shared scratch pad SRAMs were performed. 2 parallel testbench versions were developed for faster bringup. Secure Processor was replaced with transaction based acceleration for UCIe & interchiplet AI workload transfer verification which ensured no dependency on secure processor firmware. For early FW bringup, real secure Root-of-trust processor based multi-die boot up was verified. Where distributed simulation struggles with scale and speed for full boot processes and end-to-end bigger chunk data transfer to realize the PFLOPS scale of GenAI operations, emulation bridges the gap by enabling high performance, system level and software aware boot verification. Palladium was employed for DV acceleration and ZeBu for early FW and Zephyr OS bringup for Silicon Validation readiness.



```
=====================================================================
                 Statistics for Overall Performance
=====================================================================
Top Level Design Name: xcva_top
Design Utilization: 69.5
Instruction Usage: 75.04
Max emulator operating speed is 646 kHz
Design is scheduled in 1400 steps
The number of domains: 144
The number of boards: 18
Total gate count originally: 1599046400
  --(including memory access instrumentation: 1601800435)
Total gate count after transformations: 1117693238
  --(including memory access instrumentation: 1120444530)
=====================================================================
```



Parameters	Simulation	Emulation
Compile build-up time	1hr 10mins	4hrs 29mins
Sequence Generation Time	2hrs	2hrs 30mins
Run time (test duration- 15ms)	165hrs	8mins (>1000x faster!)



Product Use-Case Scenario	Quad Die Sim (hrs)	Emulation (mins)	Improvement
RAM based UCIe boot and LTSM link-up	130	8	975x
OTP based Secure Processor bootloader	96	5	1152x
QSPI NOR Flash based Zephyr OS bootloader	620	30	1240x
Multi-die boot followed by AI workload transfer over UCIe	950	58	982x

Table 2: Emulation run times

IV. Conclusion

UCIe fosters collaboration in the industry toward the future of chiplet innovation in markets and provides a reliable interoperable solution, which meets industry demand of making transistors smaller and cramming more into chips. Fitting the environment into existing limited resources for compute and memory becomes a bottleneck for DV engineers and the overall performance while simulating the usecases of the communication across dies degrades. Running the multi-die simulation in parallel on separate resources/machines while keeping the functionality/intent intact minimizes the efforts of DV engineer in creating a single environment instantiating the two or more dies to realize across-the-die scenarios. Distributed simulation provides around 3x savings in terms of simulation time when compared to UCIe DUT back-to-back in the single die setup. When a big design is split into multiple parallel sims, each individual sims takes less memory. For simulations having huge memory, distributed sim setup can allow users to fit the job in lower memory machines/existing cloud setups, instead of running it in costlier big mem machines which further helped saved $125k. The initial setup had 2 chiplets and then the setup was extended for 4 chiplet simulation. Based on implementation point of view there is no limitation in simulating SoCs with more chiplets as long as there is no limitation from the LSF machine used for distributed simulation. Across the 4 dies there were close to 6000 interposer connections and we were facing issues while generating the interchiplet configuration file as it was prone to human errors. To overcome this, an in house shell script was devised which generated the chiplet configuration file by taking the design XLS as input. Despite distributing the simulations in different LSF machines, due to the huge size of single chiplet we were able to get only 3x improvement in simulation speed though theoretically it should have been around 4x. We are still working on various simulator and testbench optimizations and trying out different simulators to improve the simulation speeds. Emulation worked wonders to address the run-time and early FW bringup concerns providing gains upwards of 1000x. Multiple challenges were overcome during the execution of this dual-pronged verification approach with emulation. Analog components like PLL, sensors, hard PHYs had to be driven from testbench forces. Electrical portion of UCIe PHY and serial IOs had to be bypassed. Due to emulator resource capacity, we ran into size limitation of the huge die. Hybrid DUT was employed and few blocks were black-boxed. QSPI transfer DV function for NOR flash boot had to be modified from hierarchical to memory mapped register access. To make this methodology platform agnostic, host integration with Accelerated Verification IPs (AVIPs) was also incorporated. Power aware emulation with UPF and power estimation using waveform is also giving considerably reliable results. Future scope trials runs are in final stages for optimization of all vendor simulation and emulation run times.