Aiding Left-shift strategy for early functional, interoperability and performance bottlenecks with Synopsys solution portfolio

Prodip Kundu, Principal Engineer Verification, Arm Bangalore
Puneet Singh, Senior Verification Engineer, Arm Bangalore
Charanpreet Singh, Principal Engineer Verification ,Arm Bangalore
Piyush Kumar Gupta, Director Emulation, Arm Bangalore
Rakesh Singh, Sr. Architect (ZeBu & HAPS), Synopsys Bangalore

Arm and Synopsys Collaboration

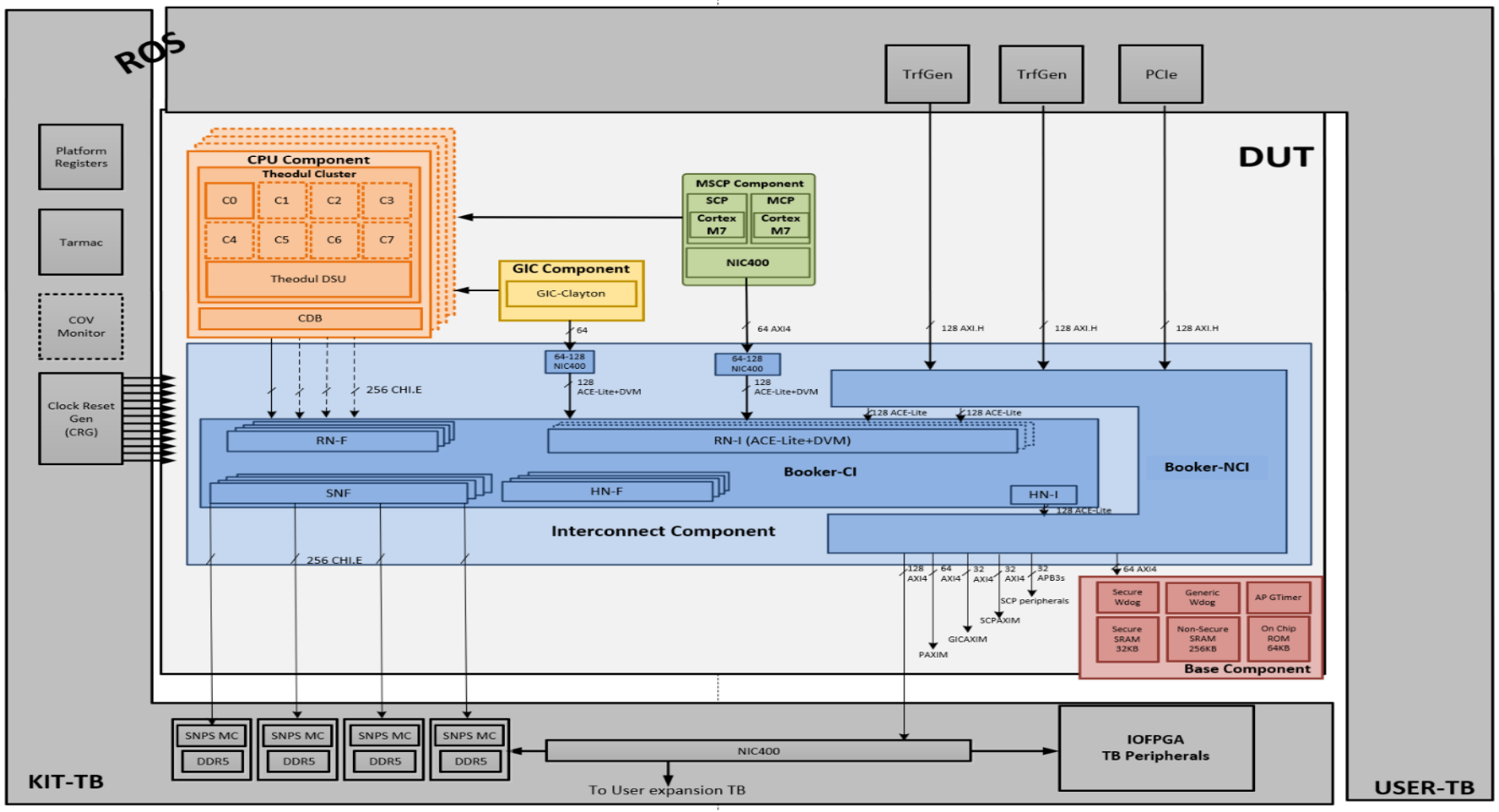
- Arm is using Synopsys Memory controller(MC) portfolio in both Infra & Client solutions
- For Infra Solution – Arm is extensively using Synopsys Chi-based MC (Chi-E) along with with Arm's internally developed Phy & DDR5
- Further exploration ongoing for the Synopsys Chi-F MC integration & further validation
- For Client Solution – The Synopsys Zebu compatible Phy (zDFI) & LPDDR5 (zLPDDR5) is getting used with Arm's AXI based MC

Arm Sub-System Overview

Integration, Interoperability, Functional Verification and Validation

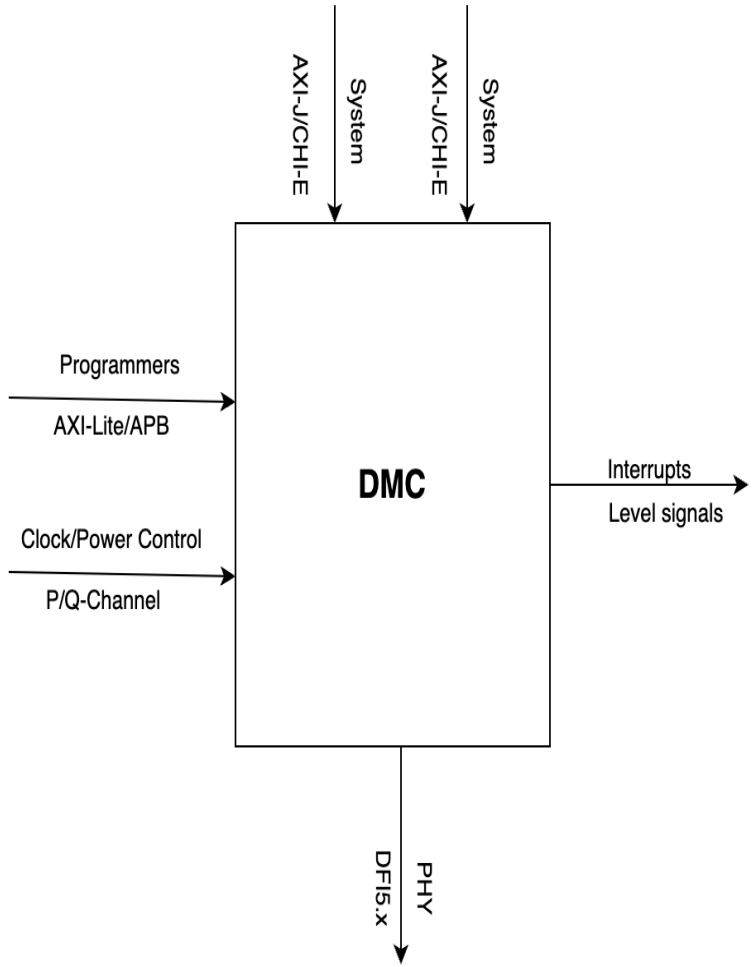
Arm Sub-system : Infra Solution

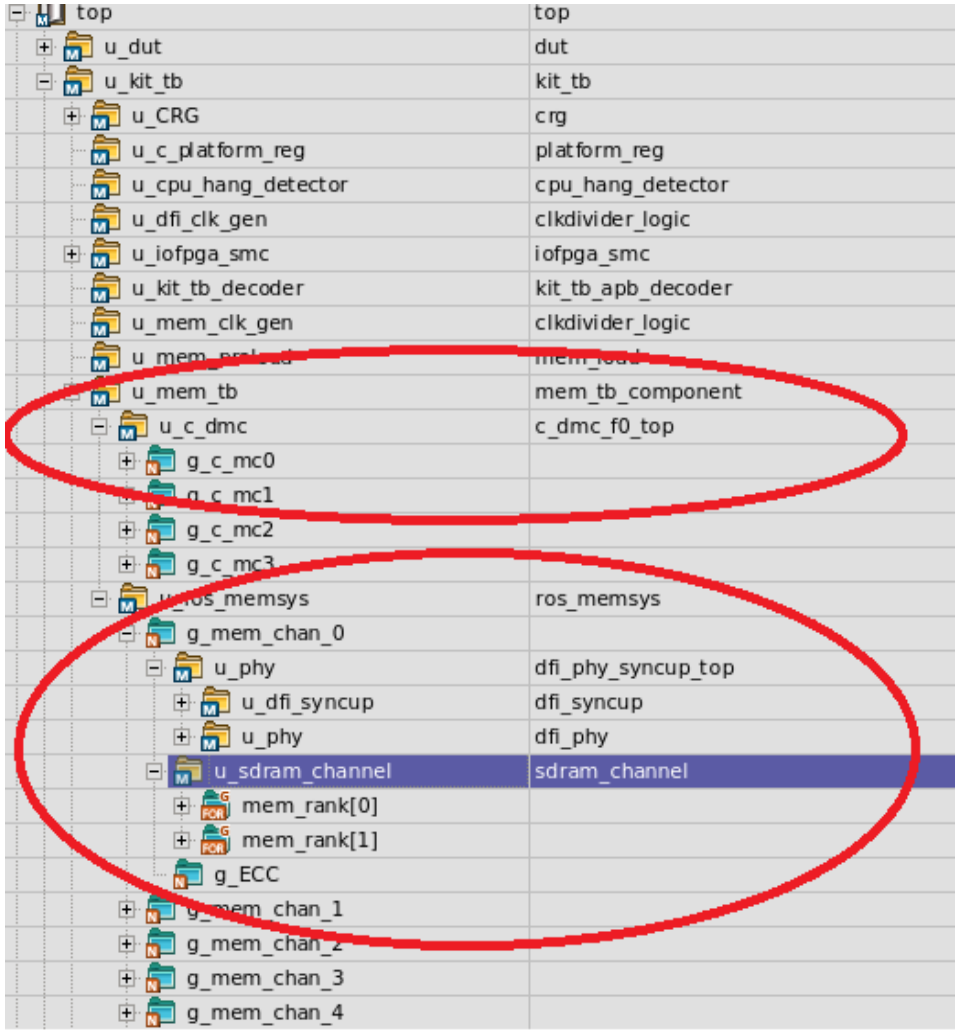
- Big Core CPUs with Chi Based interface
- Interconnect - Mesh Topology (CI-Coherent Interconnect)
- Chi based Dynamic Memory Controller(DMC)
- Memory subsystem with Phy & DDR5



DMC : Memory Controller Overview

- The DMC provides a system interface for the system to gain read/write access to DRAM memory
- It also provides a programming interface to configure and initialise the DMC and DRAM along with the DRAM interface to the external DRAM





Configurable Arm Sub-System

- Configurable SoC like system-
  - Built using SIF (System Integration Framework) – Contains both HW & SW layers
  - Interface level connections via standard AMBA bus definitions to stitch IPs
- SIF Integrates IPs as early as Alpha Milestone
- SIF provides ever platform for -
  - Interoperability testing
  - Functional validation to expose early bugs
  - Running industry standard benchmarks for performance analysis
  - Comparison with previous generation IPs to emphasize improvement

```
## Platform Parameters
m4_define(def_NUM_DMC,`4')
m4_define(def_NUM_PP,`1')
m4_define(def_NUM_DSU_CHI,`4'
m4_define(def_NUM_DMC,`4')
m4_define(def_NUM_ITS,`0')
m4_define(def_NUM_TRFGEN,`4')
m4_define(`def_NUM_THREADS',m4
m4_define(`def_NUM_INT_Q',m4_
m4_define(def_NUM_CHIPS,`0')
m4_define(def_NUM_CLUSTERS_PER_CHI
m4_define(def_NUM_MCN,`8')
m4_define(def_LCC_ENABLED,`1')
m4_define(def_L1_CACHE,`64KB')
m4_define(`def_DMC725_ENABLED',`1'
```

```
#include "mem_defs.h"
#include <platform.h>
#include "platform_register_tb_externs.h"

void dmc725_init (unsigned int DMC_BASE, unsigned int DMC_NUM
//[------ 9600 ------] : dmc-fenix_r0p0_00dev2_nahpc2_1227202
//[------ Program DMC Registers (FSP-0)------] :
MEM_RW(DMC_BASE, 0x038)   = 0x01010501;  //'address_control'
MEM_RW(DMC_BASE, 0x15c)   = 0x00043210;  //'decode_column_bit_
MEM_RW(DMC_BASE, 0x160)   = 0x00008765;  //'decode_bank_rank_b
MEM_RW(DMC_BASE, 0x164)   = 0x1f001f00;  //'address_base_mask'
MEM_RW(DMC_BASE, 0x168)   = 0x00000000;  //'decode_bg0_hash_ma
MEM_RW(DMC_BASE, 0x16c)   = 0x00000000;  //'decode_bg1_hash_ma
MEM_RW(DMC_BASE, 0x170)   = 0x00000000;  //'decode_ba0_hash_ma
MEM_RW(DMC_BASE, 0x174)   = 0x00000000;  //'decode_ba1_hash_ma
MEM_RW(DMC_BASE, 0x178)   = 0x00000000;  //'decode_cs_hash_map
MEM_RW(DMC_BASE, 0x768)   = 0x00000205;  //'memory_type'
MEM_RW(DMC_BASE, 0x76c)   = 0x00000300;  //'format_control'
MEM_RW(DMC_BASE, 0x770)   = 0x00000003;  //'feature_control[0]
```

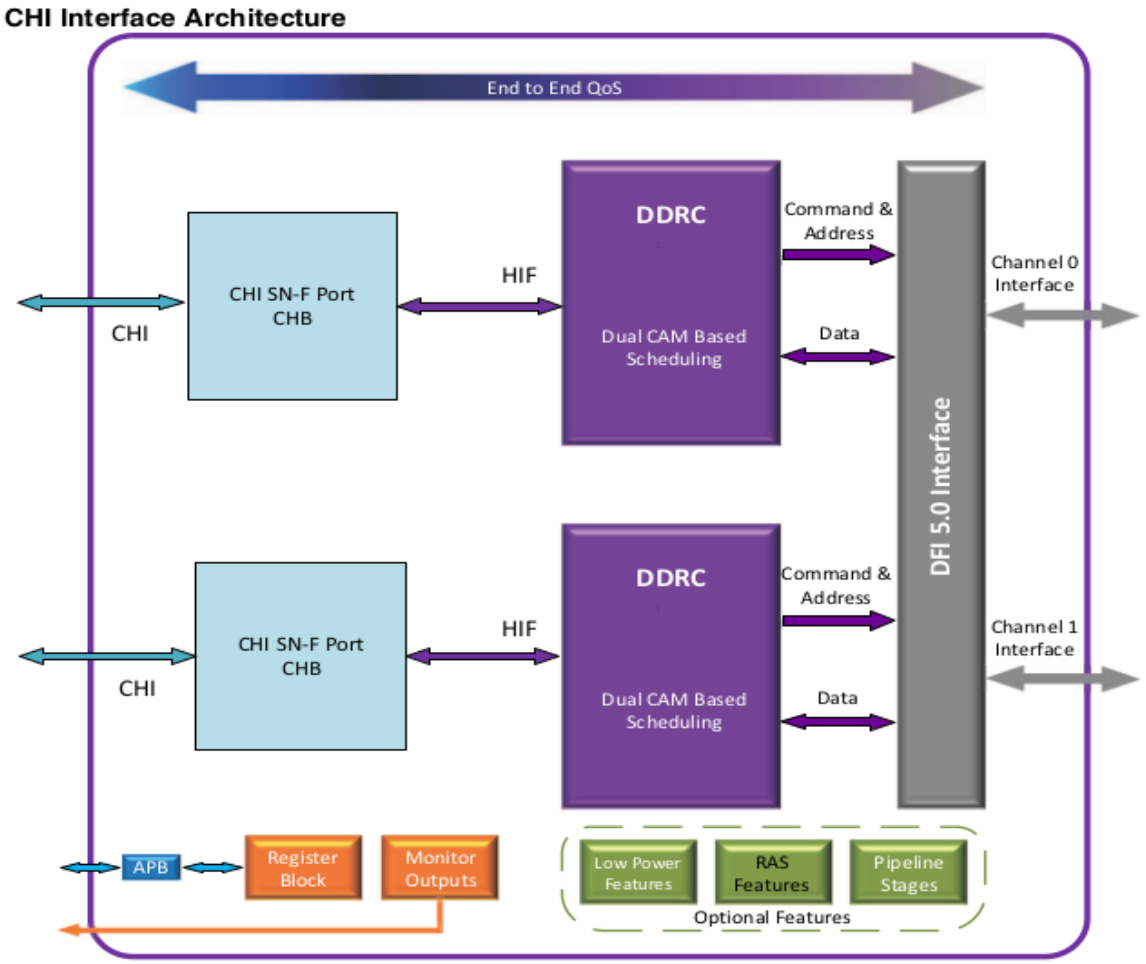
DDR5 - Chi-E based Memory Subsyem

System level Verification/Validation of Synopsys Chi-E based Memory controller and Arm’s internal PHY and SDRAM memory model

Synopsys Chi-E Memory Controller

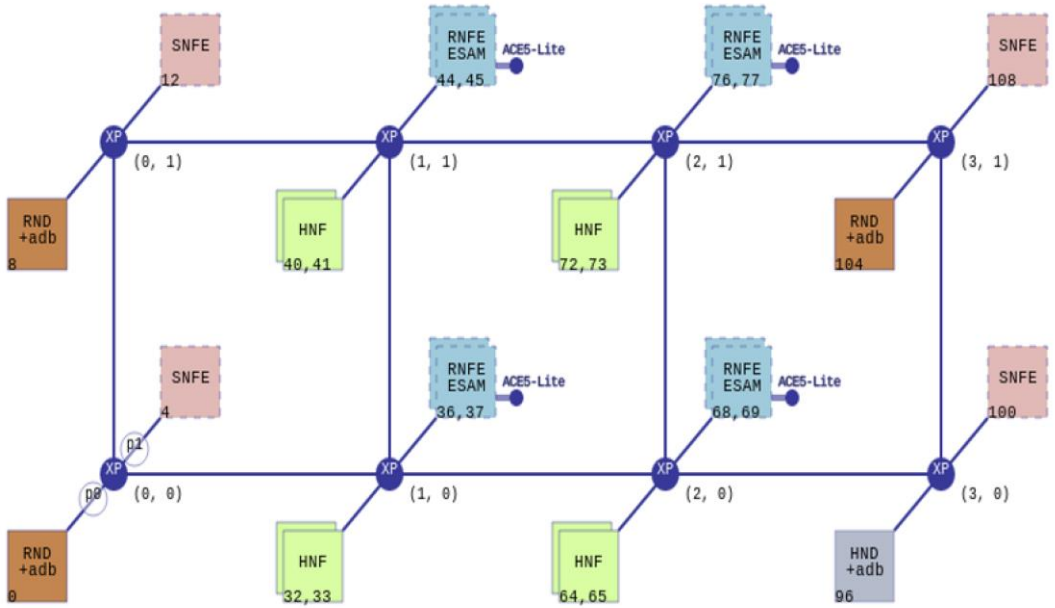
Dual Channel MC along with Dual Rank SDRAM

- Speed :
  - Data Rate 4.8 to 6.4 Gbps(giga bit per secon
  - 2.4 to 3.2 GHz clock rate
- Channel Architecture :
  - 40-bit data channel (32 Data + 8 ECC)
  - 2 channels per DIMM(Dual Inline memory module)
  - Higher Memory efficiency, Lower latency
- DIMM topology :
  - Dual Channel, with each channel being 32bit wide for data
  - ECC DIMM are generally 80-bits wide, with 4 bits per channel



System level configurations : Interconnect & DMC

❖ Scaling Number of Memory controller/Memory Channel in System



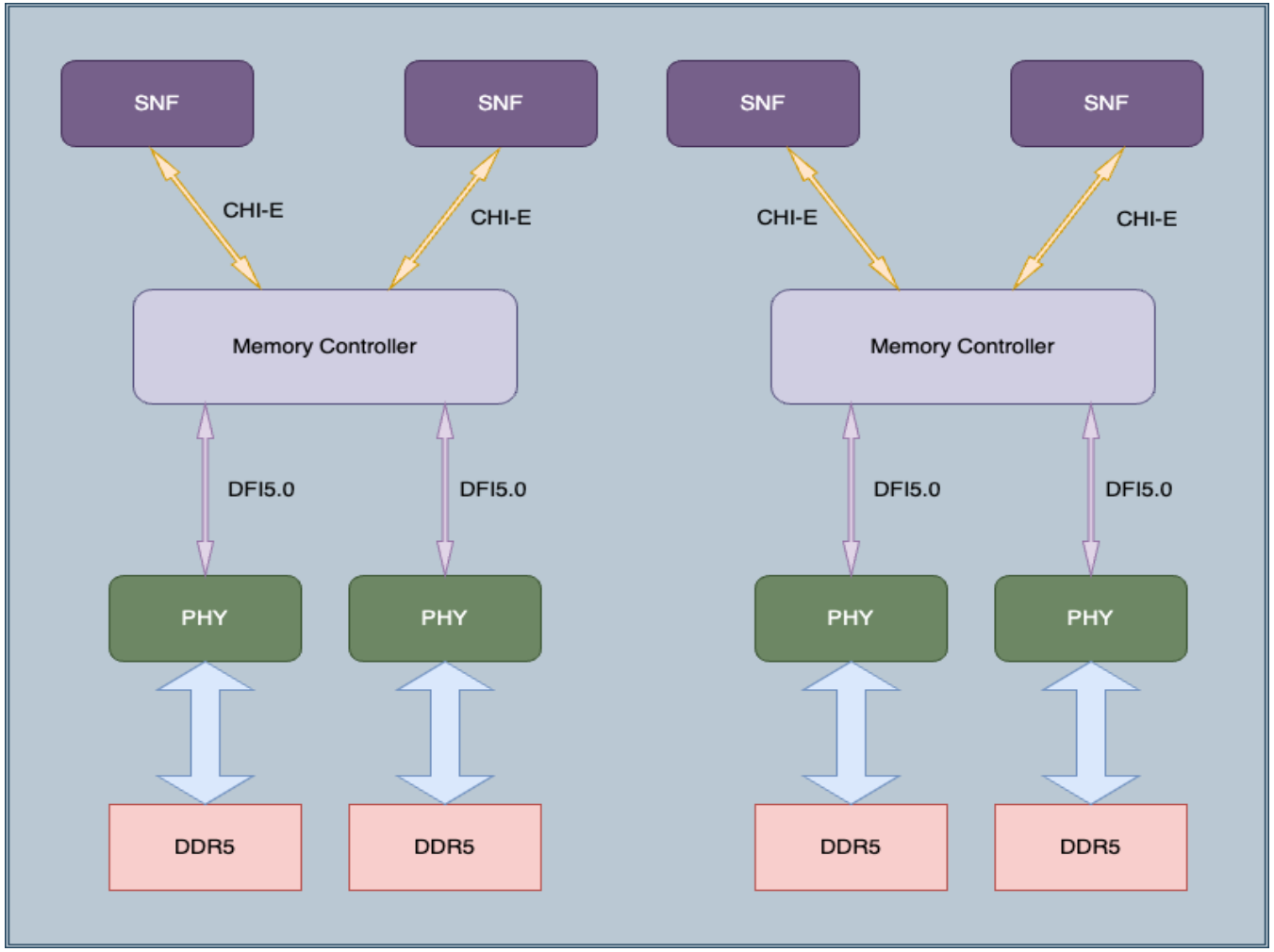
CMN(Coherent mesh node) based small mesh

Configuration	Mesh	Cores per chip	HNF Nodes	SNF Nodes	No. DMCs Channel/MC instance
1	Small	8	8	4	4/2
2	Full	32	32	8	8/4

- RNF – Request Node – CPUs/Cluster Interface
- HNF – Home Node – Internal to Interconnect with SLC(System level Cache)
- SNF – Subordinate Node – DMCs Interface

Memory Subsystem – DDR5

❖ CHI-E based DMC with DFI 5.0 and JEDEC DDR5 speed grade compatible solution



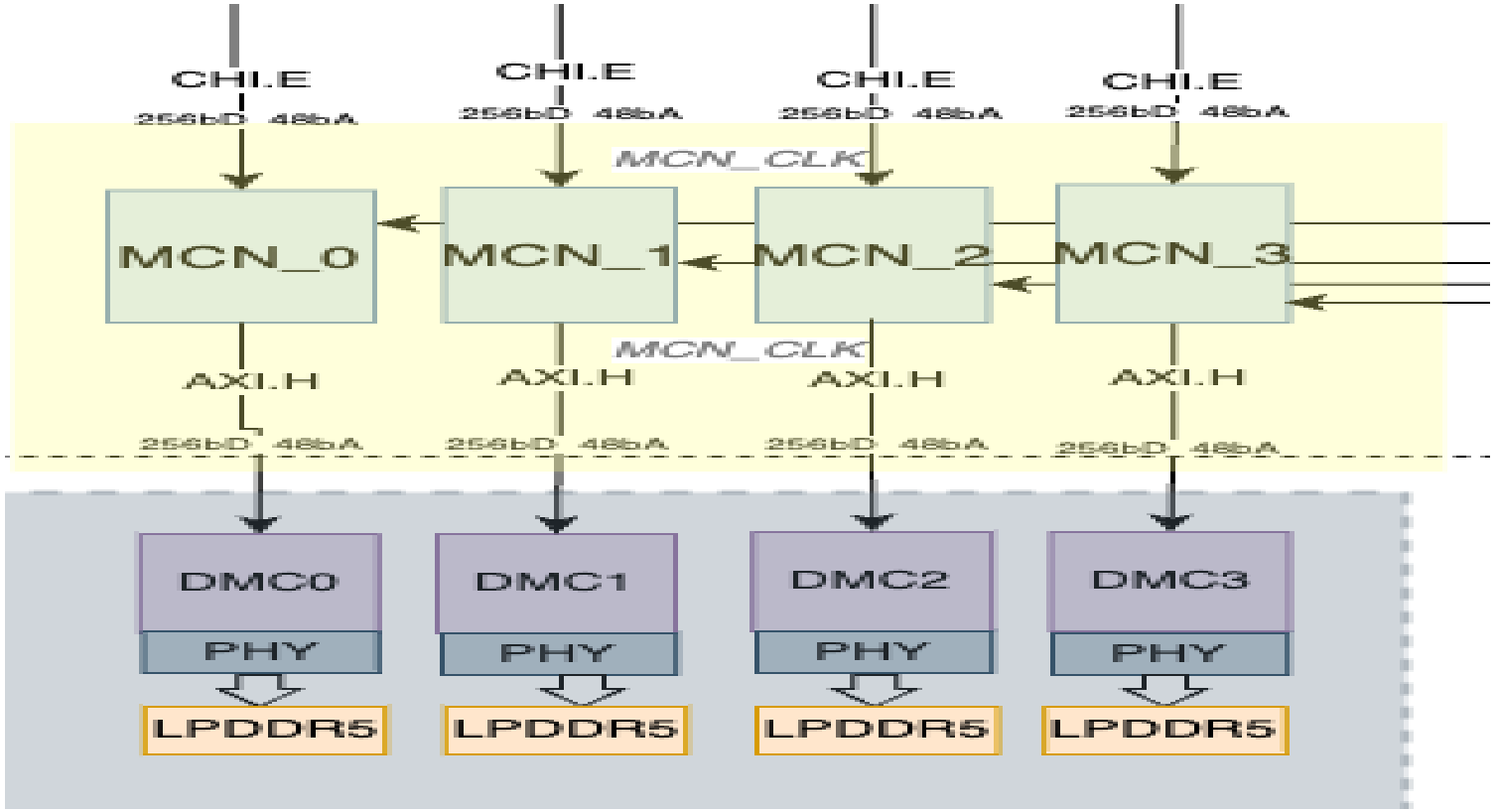
- Memory Organization : Dual-Channel memory Controller, Dual Rank SDRAM.
- DFI5.0 compliant interface DesignWare DDR5/4 PHY or other DDR5/4 Phy.
- System Interface : Chi-E
- Programming Interface : APB
- PHY & DDR5 : Arm's Internal RTL based solution.
- Clock Mechanism :
  - 1:4 CKR(Clock Ratios) – DDR5 mode only supports DFI 1:4 frequency ratio mode
  - DFICLK : 800 MHz
  - MEMCLK : 3200 MHz

Synopsys zDFI & zLPDDR5 with AXI Based MC

System level Verification/Validation with Synopsys zDFI & zLPDDR5

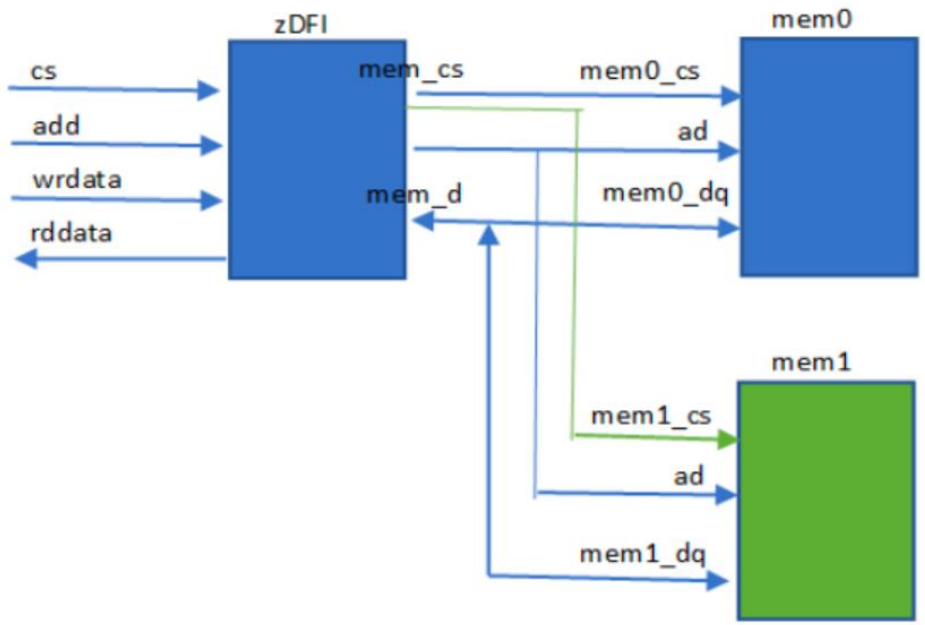
- Combination of Small + Big Cores CPUs with Chi based interface
- AMBA AXI-based Dynamic Memory Controller(DMC) connected with MCN(Memory Controller Node) based interconnect
- Memory subsystem with Phy (DFI 5.1) & JDEC LPDDR5
- Used Synopsys Solution – Zebu emulation friendly Phy (zDFI) & Memory (zLPDDR5)

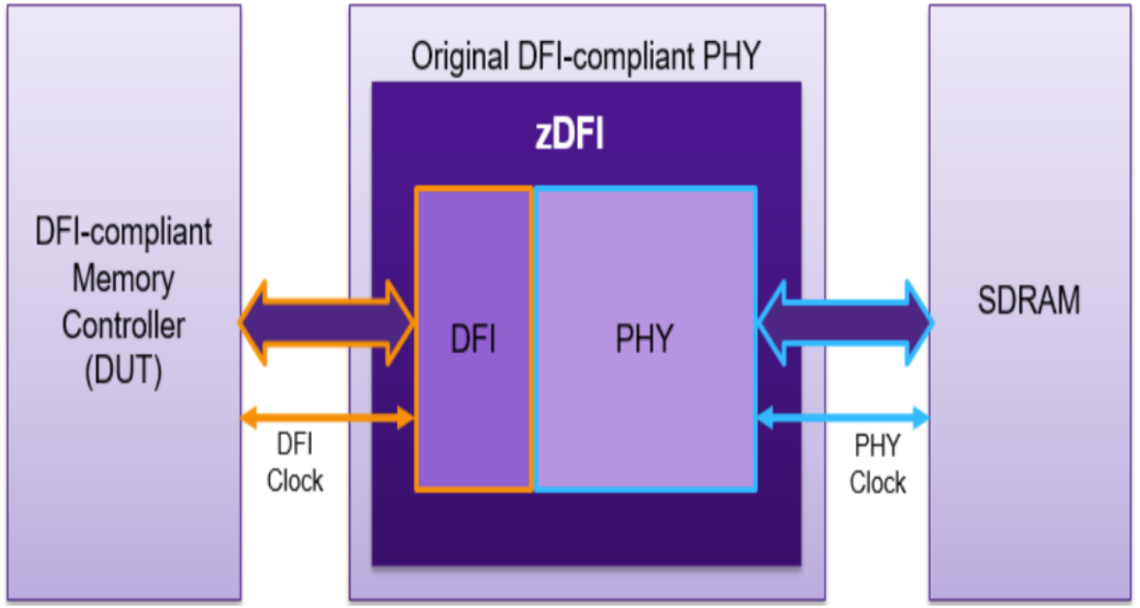
System Level Design - Client Solution



AXI-based Memory Subsystem

- AXI-based DMC with DFI 5.1 and JEDEC LPDDR5.x speed grade compatible solution
- Non-existing Arm Phy/Memory Solution : Requirement of using 3rd Party EDA Vendor's Emulation(Zebu) friendly industry standard (JEDEC) Memory Model & Phy
- Configurable LPDDR5 dual Rank model & support of different speed grades (8533,9600 Mbps)
- The DMC init programming – proving IP level init sequence working on System level (with required changes) & back-door access validation.
-



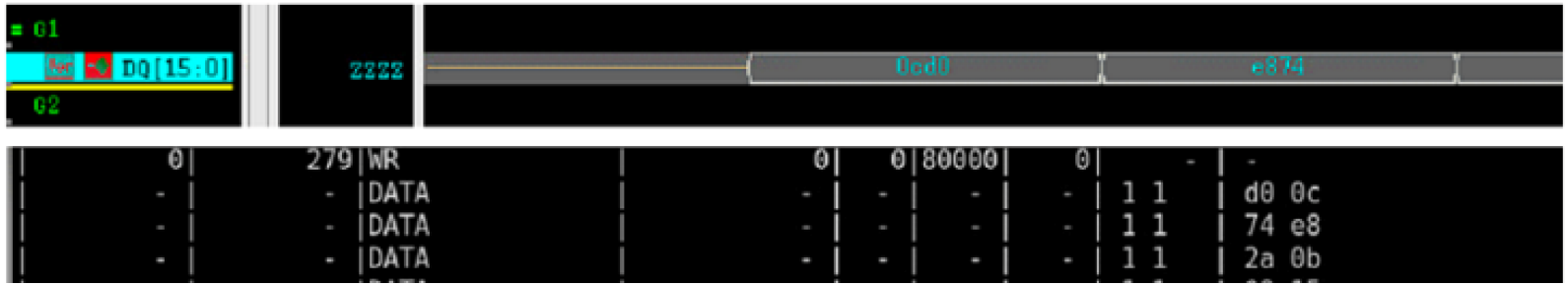


How Synopsys Helped Arm

- The Synopsys solutions helped Arm to verify all next generation IPs

(e.g. CPUs, Interconnect etc.) at the Sub-system level context

- The Synopsys solution helped Arm –
  - Validation of Memory Front-door access
  - DDR Preload Or Back door access validation
  - Speed grade changes , WR-RD delay control ,DDR Memory Size Control, Encodings-RBC/BRC etc.
- The Chi Based DMC helped on validating the Chi based next generation Arm CPUs & Interconnect
- The Synopsys zDFI & zLPDDR5 portfolio helped arm to validate Arm memory controller with industry standard Phy & Memory Models
- The zLPDDR5/zDFI analyzer feature helped for faster debug (by reporting DRAM txns with data payloads)



Feedbacks & Future Work

- The ZS5 Zebu emulator's high design frequency helped early enablement of complete SoC on emulation Platform- the achieved design frequency is around half of driver frequency

```
Timing     | driverClk: 2631 kHz (380 ns)                                              |
           | Pre-Post delta before rounding: {driverClk: 3415 -> 2653 (-22%)}          |
           | bottleneck: routing data paths : 377ns, Longest Delays: {data: 377 ns, memory: 164 ns}, #MGT in top path: {data: 1, filter: 0
-----------+----------------------------------------------------------------------------------------------
FPGA       | Total Num. of completed FPGAs: 84                                         |
```

- More collaborations with cross functional teams. The delivered Sub-system RTL (integrated with Synopsys IPs) enabled other teams (e.g. performance Analysis, FPGA etc.) for their further validation
- Faster verification closure on Performance & Benchmarking - CPU To memory path Read latency exploration (with industry standard workloads e.g. SPEC, Geekbench, Speedometer etc.)
- **Future Work :**
  - ➢ Enablement of Synopsys Real world Phy & Synopsys DDR5 memory model at System Level
  - ➢ Further validation of Arm's Chi-F IPs with Synopsys Chi-F MC & future collaboration on Chi-G MC
- **Some improvement areas:**
  - ➢ Observed long Zebu compile time
  - ➢ More fine tune control on Memory Content , could be useful for Back-door /preload access debug


THANK YOU
Our
Technology,
Your
Innovation™