DWC LPDDR5/4/4X PHY Quick Start

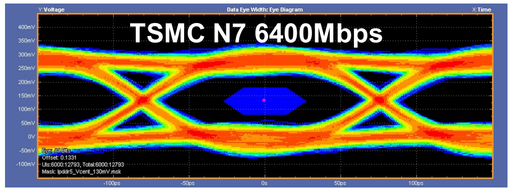
Confidential

Agenda

LPDDR5/4/4X PHY: Feature Overview

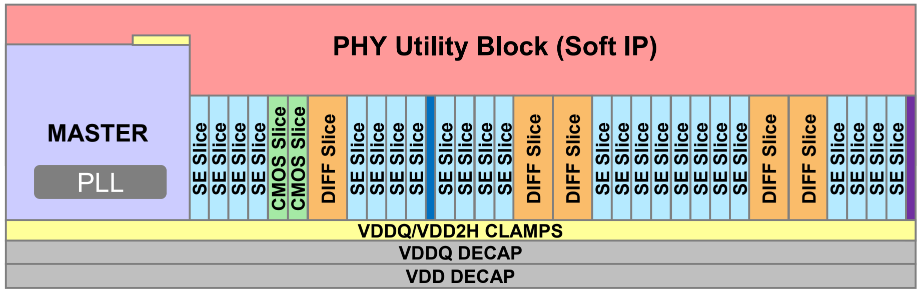
LPDDR5/4/4X PHY Overview

Next Generation LPDDR PHY



- Wide range of LPDRAM protocol & system support
  - LPDDR5 up to 6400Mbps and LPDDR4/4X up to 4267Mbps
  - Dual channel support in one PHY
  - DRAM on PCB (embedded) & PoP
- Advanced PHY training & programmable features
  - µ-processor based training using firmware
  - PHY Independent mode with DFI 5.0 Controller interface
  - Up to 15 trained states/frequencies
  - PHY Utility Block (soft IP component) for training, BIST, configuration registers
- Highly configurable physical implementation
  - IO slice based architecture eliminates area consumed by unused pins
  - Wide variety of supported configurations
  - PHY can be implemented on any side of die or "L-shaped"

- Best in class Signal Integrity
  - VT compensated DLLs for DQS centering, WL, per bit deskew
  - IO receiver DFE to increase data eye opening
  - Driver boost for improved write performance
  - On-the-Fly IO calibration & PVT compensation



- Major area reduction vs. LPDDR4X multiPHY
- Large power reduction vs. LPDDR4X multiPHY

LPDDR5/4/4X PHY Differentiation

- Support for data rates up to 6400Mbps in 5/6/7nm and 16/12nm
- Small area, low power
  - Hard IP is 40% less area, lower power versus LPDDR4X multiPHY
- Unique Dual 16-bit PHY option for improved PPA
- Firmware based training using microcontroller-based training engine
  - Firmware training uses complex patterns for optimal training
  - Field upgradable, *reduces risk of adopting emerging LPDDR5 protocol*
  - Per-bit 2D eye mapping diagnostic tool
- Input receiver equalization to add margin to the data eyes and timing budget
- Only ONE PLL per PHY
- Support for up to 15 active operating states, each can have an independent operating frequency, feature set optimization [e.g. ODT on/off, PLL on/off, DFE on/off etc.]
  - Trained at boot time, settings updated as VT drifts
  - Power becomes proportional to performance required

LPDDR5/4/4X PHY: Performance Features

Design Improvements to get to 6400 Mbps …

- Major design improvements:
  - PLL updated to reach 6400
    - DfiClk->PClk path through PLL replaced with synchronizer to enable this
  - LCDLs tuned for finer granularity delays and reduced insertion delay
    - Controls also updated, from programming in 1/32 UI in LPDDR4x multiPHY to 1/64 UI in this PHY
  - Thin-oxide driver greatly reduces insertion delay, and hence Power Supply-Induced Jitter (PSIJ)
  - Rx Data insertion reduced to near-zero
    - No BDLs; sampler-on-the-pad
  - 1-tap fully loop-unrolled DFE in PHY
  - Duty-cycle-adjuster (DCA) added to WCK
    - DRAM includes a DCA and Duty Cycle Monitor (DCM)
- And major firmware improvements to manage these improvements! (coming up in later slides)

LPDDR5/4/4X PHY: Performance Features

to support for 6400Mbps

- Training
  - CA training for per-bit delay + Vref
  - DRAM WCK DCD training
  - Rx DQ trained independently to RDQS_t vs RDQS_c
    - Removes DQS DCD dependence
  - Rx DQ DFE and Tx DQ DFE
    - No assumption of an eye without DFE
  - Support for DQ 2D training (Rx & Tx) at any data rates

- Easier implementation and timing closure
  - Bit slice architecture to optimize area
  - Easier timing closure with ARC running DFICLK/2
  - Support for DFI 1:4 mode
    - for LP4/LP4X 1:2 mode will give lower latency, 1:4 will make timing closure easier
  - All clock gating is moved inside of the HardIP
  - PUB DFI clock tree skew matching is straight forward
  - PllRefClk is separated out from DFICLK to support ASST and reduced jitter

Synopsys Confidential Information

LPDDR5/4/4X PHY Power Supplies

VDD[1]	VDD2H[1]	VAA_VDD2H[1,2]	VDDQ[1]
• "Core" power rail • Value is process-nominal – TSMC N5 = 0.75V – TSMC N7 = 0.75V – TSMC 16/12FFC = 0.8V	• Power supply for CMOS driver, POR, and miscellaneous circuits • LPDDR4/4X = 1.1V • LPDDR5 = 1.05V (same as DRAM VDD2H)	• Power supply for PLL • Options – Independent Supply 1.05V – 1.8V (nominal) – Connect to VDD2H Supply	• LPDDR5 = 0.5V / 0.3V • LPDDR4X = 0.6V • LPDDR4 = core VDD

Notes:
1. Rails may power-on in any order
2. The PLL supply (VAA_VDD2H) MUST be separated from all other supplies at the die level. VAA_VDD2H and VDD2H can be driven from the same regulator however, there must be package or PCB filtering of the VAA_VDD2H supply

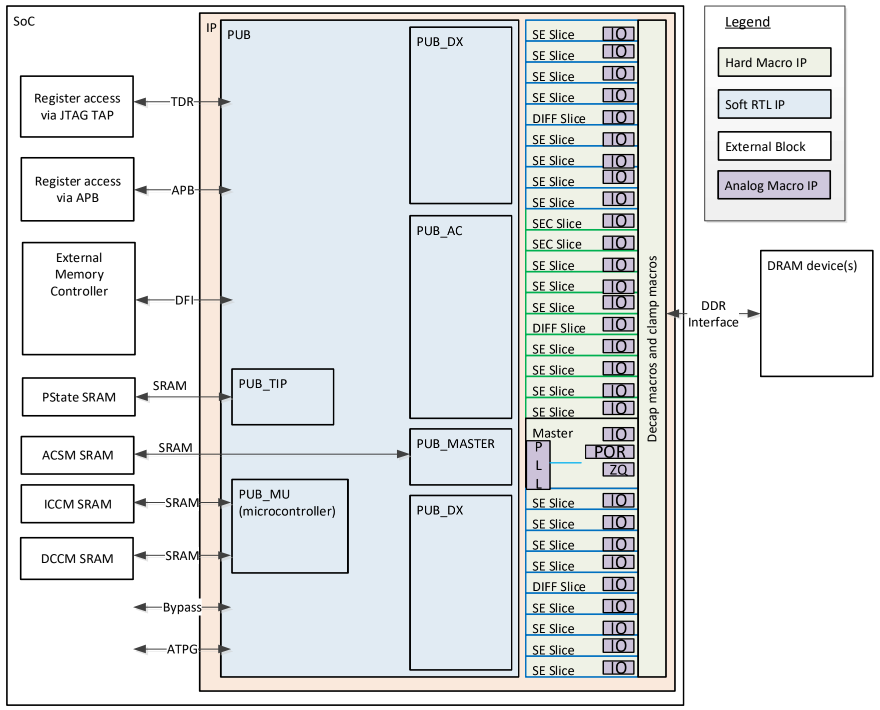
Architecture Overview

Organization, function, and DFT features

LPDDR5/4/4X PHY TOP Overview

External Interface

- The LPDDR5/4/4X PHY interfaces to the following:
  - JEDEC-compliant DRAM
    - See applicable JEDEC specification
  - DFI 5.0-compatible memory controller
    - See DFI 5.0 specification
  - AMBA APB Master
    - See PUB Databook section 7.2 APB Interface
    - Primarily used for CSR access
  - JTAG TAP controller
    - See PUB Databook section 7.2.2 TDR Serial Interface
    - Used for test access to everything that is available via APB
  - Fly-over pins to allow direct access to IOs (e.g., for boundary scan and scan shift)
    - See PUB Databook section 9.7 Bypass Mode
  - Interrupt interface
    - See PUB Databook section 7.11 Interrupt Implementation
  - BP_PWROK input from SoC
    - See PUB Databook section 6.3.1 BP_PWROK.
    - indicator that clocks and power supplies are clean i.e. stable and within their specifications.

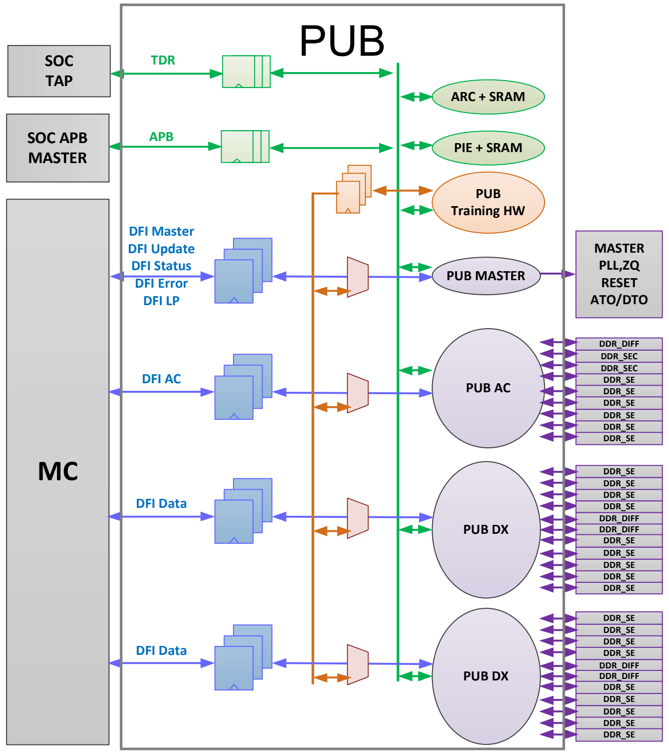


LPDDR5/4/4X PHY PUB Overview

- SOC interfaces
  - JTAG TDR (test)
  - APB (configuration register access)
  - DFI 5.0 (Expected revision supporting LPDDR5)
- SRAM Interfaces
  - ECC protected SRAM interfaces ported to top level for ease of integration
- Functions
  - Training logic
  - Dynamic power savings logic
  - Timing and impedance calibration
  - Hard-IP interfaces
    - All timed signals on DFI clock or pseudo-static
- Topology
  - Reduced number of cross-PHY wires
  - Organized as multiple independent logical blocks
  - Enables a “split-PUB” design with pipelining on non-latency critical paths
  - Physical design teams can place / partition non-performance critical blocks out of critical areas.
  - Reduced Timing exceptions from previous design
  - Reduced Register count from previous design
  - Removed Ring-Oscillator logic from PUB for easier timing closure

LPDDR5/4/4X PHY PUB

16-bit PHY Topology Diagram



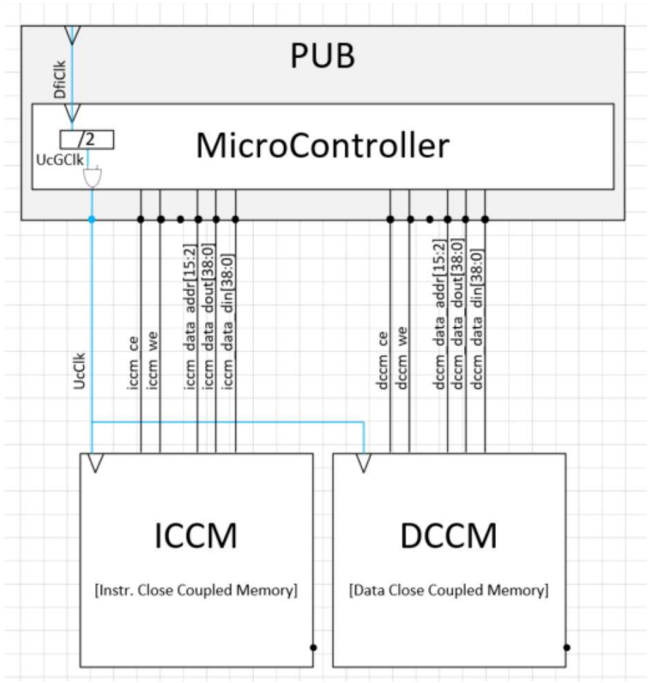
- Fully modularized PUB architecture
  - Cross-PUB signals are fully pipelined
  - Enables placement aware synthesis to achieve easy timing closure
- Allows for architectural clock gating
  - Training logic is gated when not in use
  - Compensation logic is clocked on demand
  - Configuration registers only clocked when accessed
  - PUB data paths dynamically clock gated based on DFI LP inputs
- PUB: Hard-IP Interface
  - All timing paths to/from Hard-IP are on DFICLK

Storage in the PUB Module

- PUB (PHY utility block) logic in Synopsys PHYs is responsible for carrying out FW based PHY training
- Storage in the PUB module
  - FW executable code
  - Data-structures (message blocks) for the FW code
  - Address Command State Machine, which can program command sequences to the DRAM for training
  - PHY Power State (PState) info for storing the configuration parameters for each supported frequency
- Two ways for implementing storage in general
  - Registers (flip-flop) or SRAMs
- SRAMs are preferred over Register arrays for storing a large amount of data
  - Register array is a simpler implementation/integration option, although area and power intensive
  - SRAMs require a memory compiler for integration, but more efficient in area and power savings
- LPDDR5 standards require longer training sequences and hence larger densities for storage

SRAMs in LPDDR5/4/4X PHY PUB

- SRAMs needed in PUB
  1. ICCM (Instruction Closely-Coupled Memory): FW instruction SRAM
  2. DCCM (Data Closely-Coupled Memory) : Data SRAM
  3. ACSM (Address Command State Machine) SRAM
  4. PState SRAM (optional)
     - If less than or equal to 2 frequency states, info is stored in registers, PState memory array is not required
     - If greater than 2 frequency states, 1 PState memory array is required. The size of memory array depends upon number of power states supported; up to 16 such power states can be supported by the PHY
- ICCM and DCCM SRAMs are optionally ECC protected
- ACSM and Pstate SRAMs are optionally parity protected
- Frequency of the SRAMs
  - ICCM/DCCM: DFICLK or DFICLK/2
  - ACSM SRAM, Pstate SRAM: DFICLK
- Mission mode usage
  - ICCM/DCCM are not used and can be powered down
  - ACSM SRAM is needed for periodic retraining on LPDDR5/4
  - PState SRAM is needed for frequency changes
- Refer to PUB data book for exact SRAM sizes

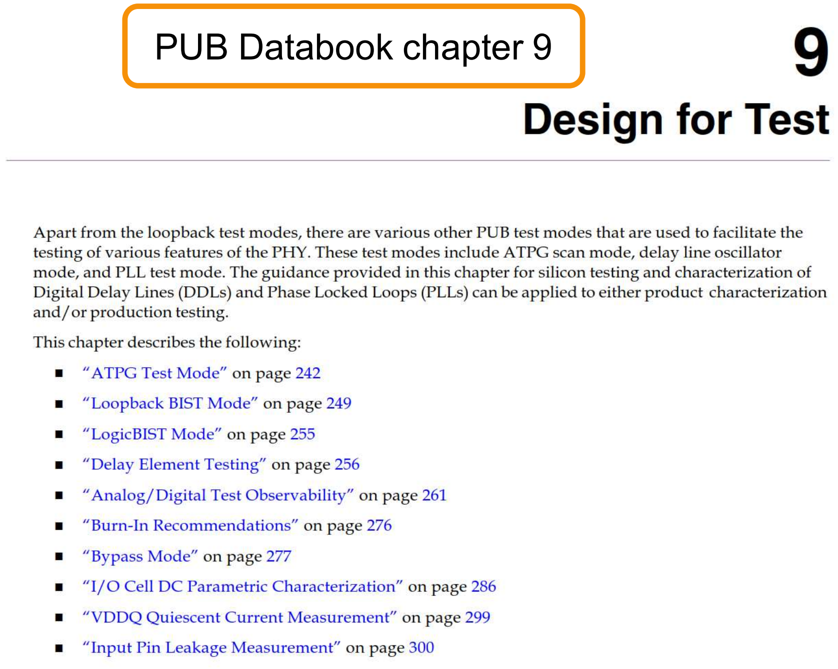


LPDDR5/4/4X PHY: DFX Features Part 1

DFX new Features

- Enabling the use of SNPS PHY PLL as frequency source for customer's at-speed test solution
- Dedicated BP_DTO and BP_ATO pins for debug and observability.
- Dedicate pin (BurnIn) to enable burn-in mode.
- Dedicated VIX output pins for differential receivers.
- Dedicated clock pin (RxTestClk) for mission mode JTAG SAMPLE.

PUB Databook chapter 9



9

Design for Test

Apart from the loopback test modes, there are various other PUB test modes that are used to facilitate the testing of various features of the PHY. These test modes include ATPG scan mode, delay line oscillator mode, and PLL test mode. The guidance provided in this chapter for silicon testing and characterization of Digital Delay Lines (DDLs) and Phase Locked Loops (PLLs) can be applied to either product characterization and/or production testing.

This chapter describes the following:

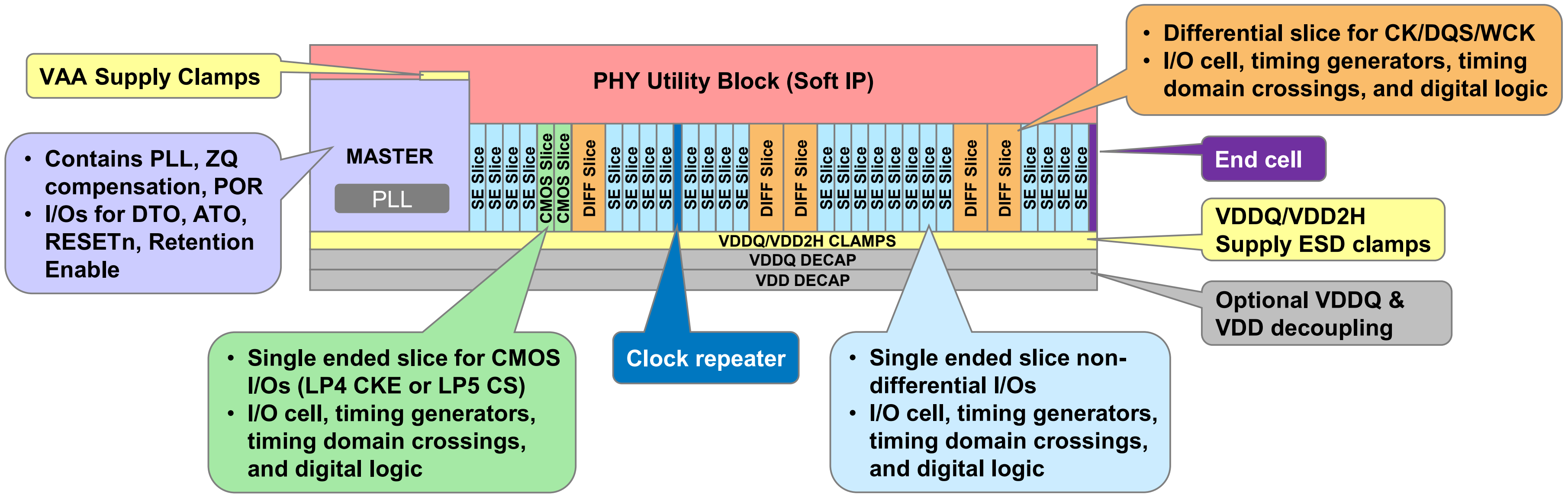
LPDDR5/4/4X PHY: DFX Features Part 2

- Hard IPs:
  - Scan: Multiple chains per block, with 100 flops max, and separate chains per clock
  - Delay lines: Can overwrite any value and put lines into an oscillator mode
    - 100% of oscillator hardware is contained inside the Hard-IP (no customer timing issues)
  - Loopback: both pad-side and core-side loopback supported
    - Core-side loopback can be configured to bypass all I/O and prevent I/O toggle
  - Digital and analog test points: Allow observation of internal signals
  - Impedance codes: Can overwrite any value for test / characterization
  - VREF: Direct programming and observation options
  - IO Flyover: IOs have a test mux to provide asynchronous flyover paths for SoC uses
  - IO Characterization: DC parametric, quiescent IDDQ, input pin leakage
    - Improved visibility on differential receivers
- PUB:
  - Firmware control of BIST features located in the PUB
  - Customer can add any DFT they desire (scan, scan isolation, compression, etc.)
  - Test Data Register interface to support SOC JTAG infrastructure to access any CSR (i.e., just like the APB port)

Hard Macros

LPDDR5/4/4X PHY Hard-IP Overview

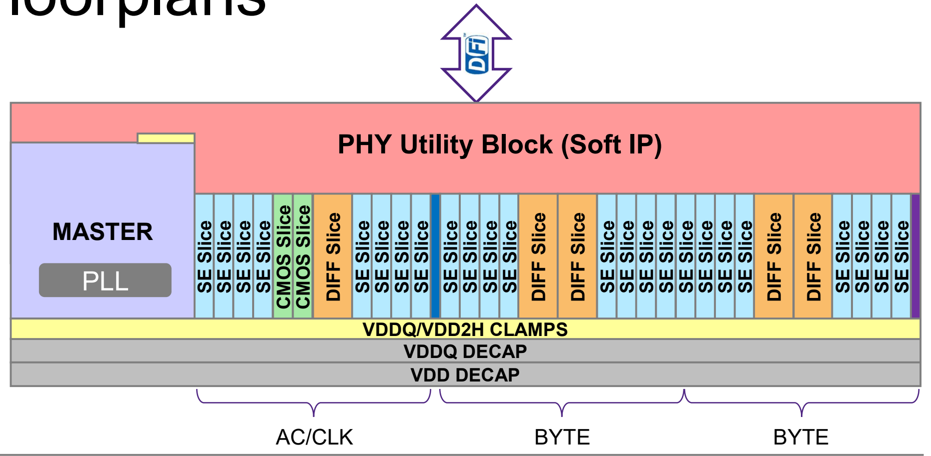
- All hard-IP inputs/outputs from PUB are synchronous with DFICLK or pseudo-static
- All hard-IP inputs/outputs, other than IO pad nodes, are on VDD rail
- All hard-IP blocks may be reordered in floorplan, no RTL dependence


VAA Supply Clamps
PHY Utility Block (Soft IP)
Differential slice for CK/DQS/WCK
I/O cell, timing generators, timing domain crossings, and digital logic
Contains PLL, ZQ compensation, POR
I/Os for DTO, ATO, RESETn, Retention Enable
MASTER
PLL
SE Slice
CMOS Slice
DIFF Slice
End cell
VDDQ/VDD2H CLAMPS
VDDQ DECAP
VDD DECAP
VDDQ/VDD2H Supply ESD clamps
Optional VDDQ & VDD decoupling
Single ended slice for CMOS I/Os (LP4 CKE or LP5 CS)
I/O cell, timing generators, timing domain crossings, and digital logic
Clock repeater
Single ended slice non-differential I/Os
I/O cell, timing generators, timing domain crossings, and digital logic

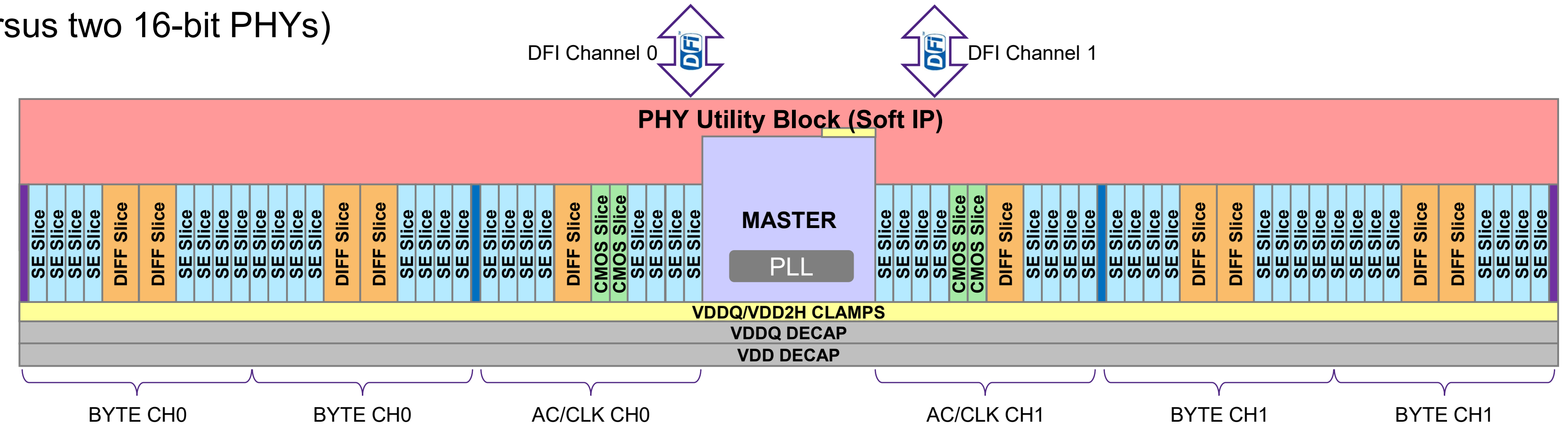
LPDDR5/4/4X PHY Example Floorplans

Discrete Optimized Floorplans Shown

Single channel 16-bit PHY with one DFI interface



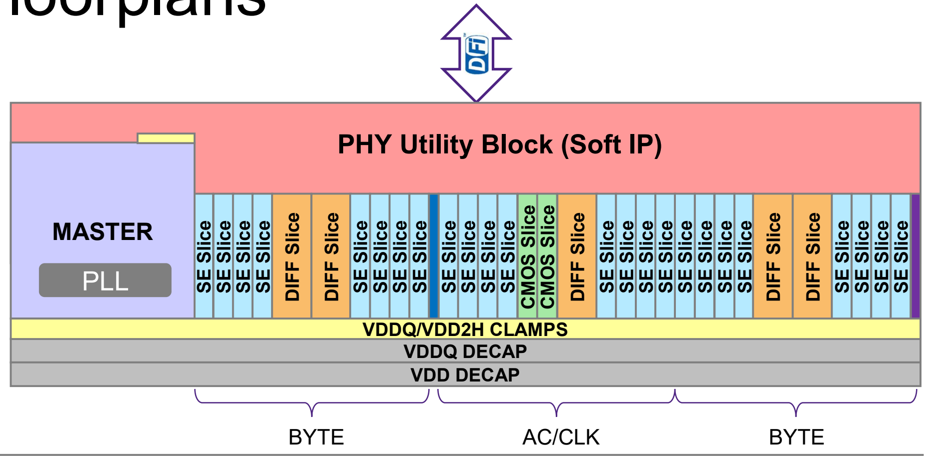
Dual channel 32-bit PHY with two 16-bit channels each with a DFI interface (Single MASTER, PUB & SRAM saves area versus two 16-bit PHYs)



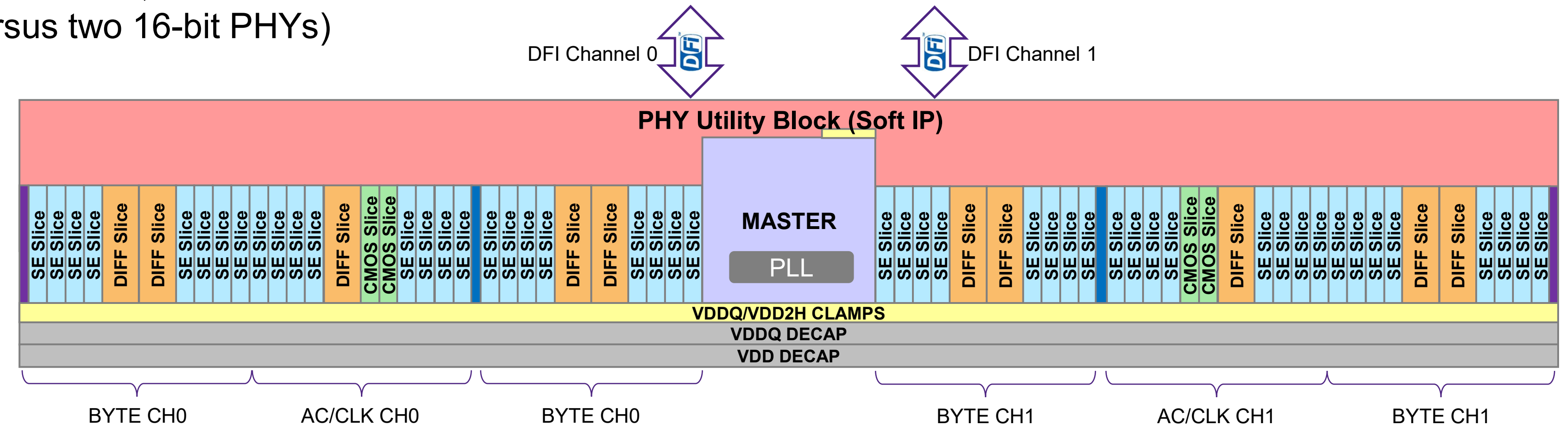
LPDDR5/4/4X PHY Example Floorplans

PoP Optimized Floorplans Shown

Single channel 16-bit PHY with one DFI interface



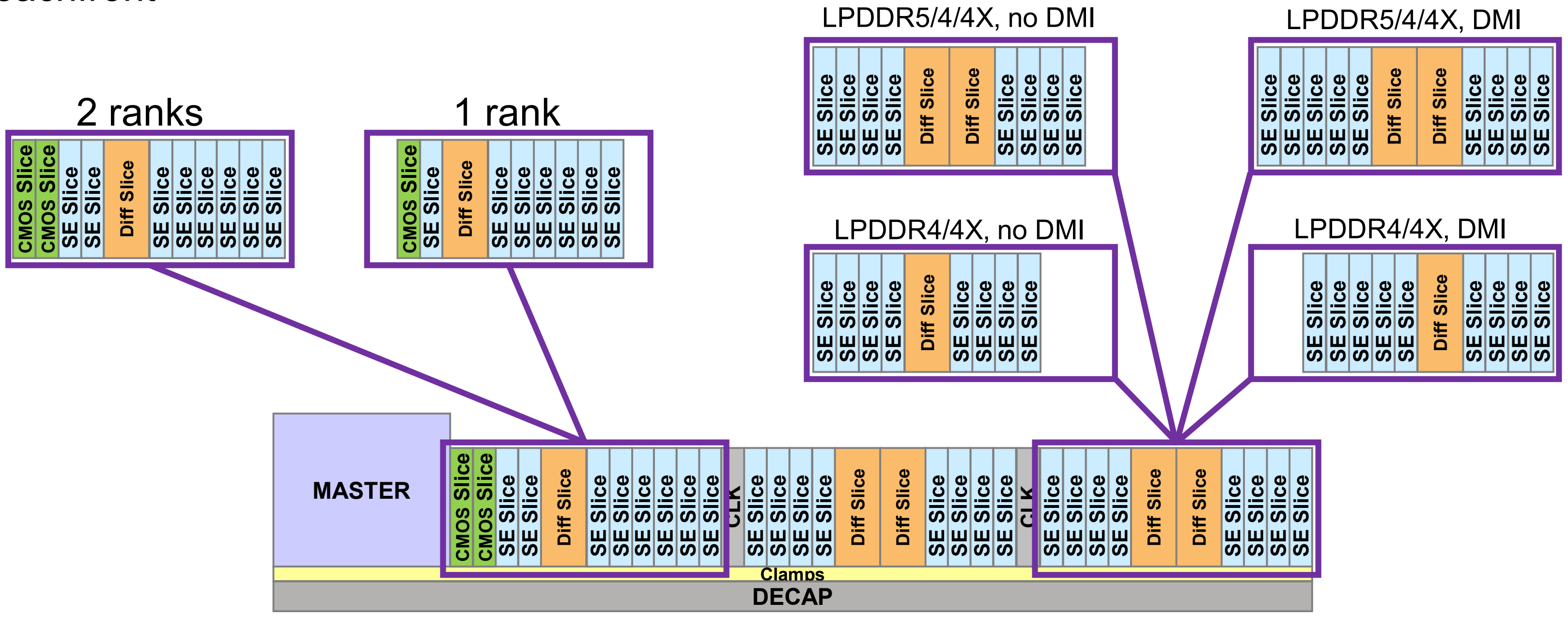
Dual channel 32-bit PHY with two 16-bit channels each with a DFI interface (Single MASTER, PUB & SRAM saves area versus two 16-bit PHYs)



LPDDR5/4 PHY Bit-slice Based Design = Maximum Flexibility

PHY PPA Tailored to Requirements

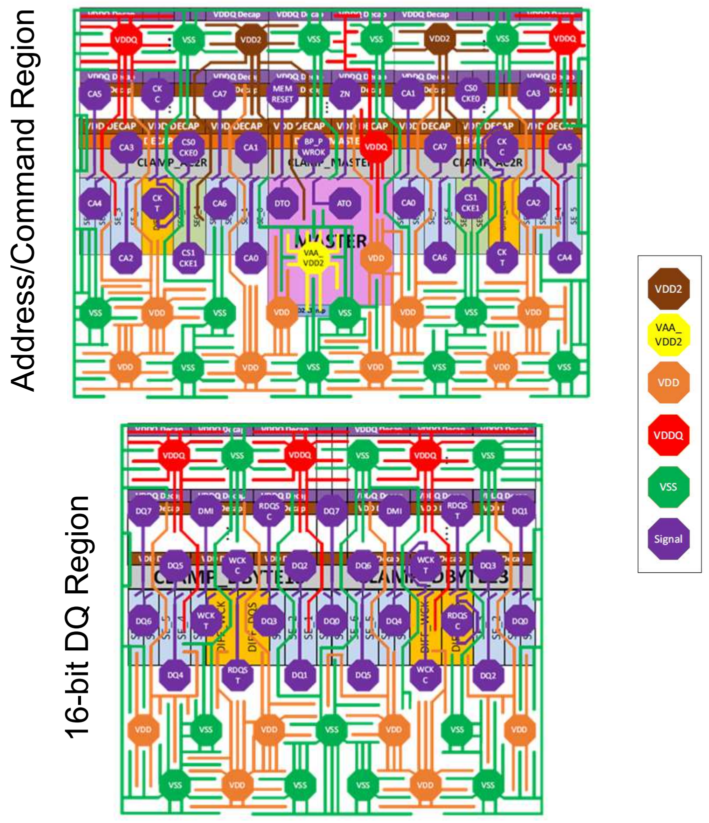
- Number of address/command IOs can be adjusted for number of ranks to reduce beachfront
- DBYTE can be customized to only include the IOs needed, minimizing beachfront



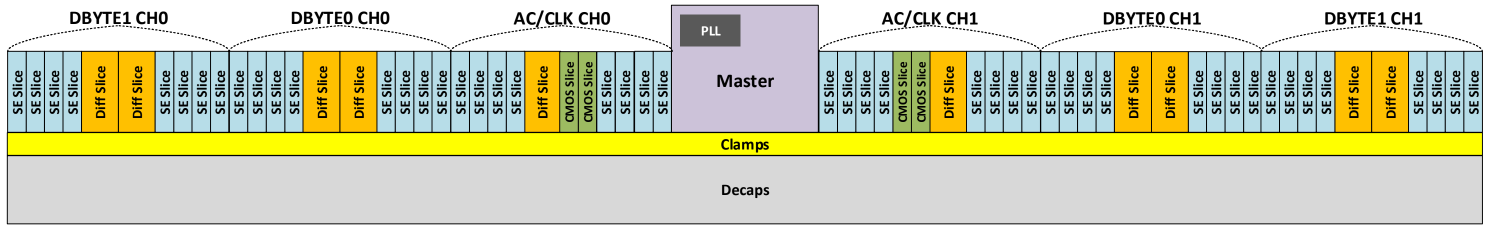
LPDDR5/4/4X PHY Example Bump Pattern

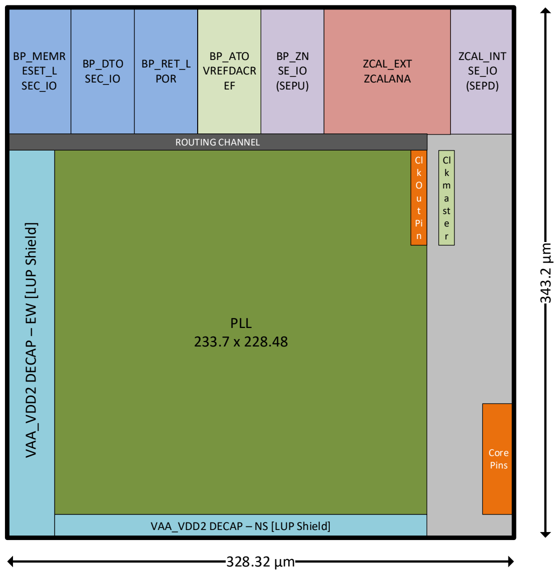
Flexible C4 bump options

- Support for wide variety of bump pitches
- Recommended pattern minimizes RDL route on signals
- Bump routes should be constructed to minimize resistance and capacitance
- The power and ground connections made in AP/LB are intended to boost the existing grid
- Bump arrangements shown are examples only
  - Actual bump assignments will depend on specific package and die bump rules, IR drop analysis, die bump EM current limits

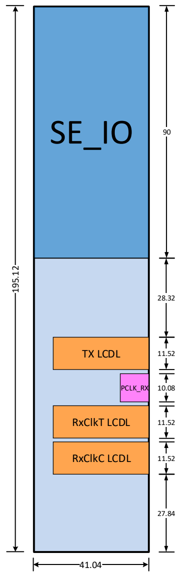


Hard Macros & Custom Cells DDR CKT Delivers

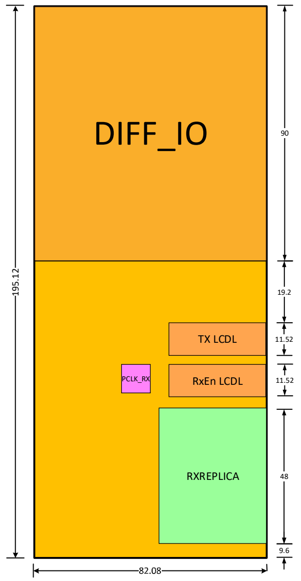




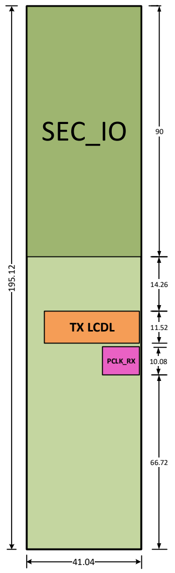
MASTER



SE SLICE



DIFF SLICE

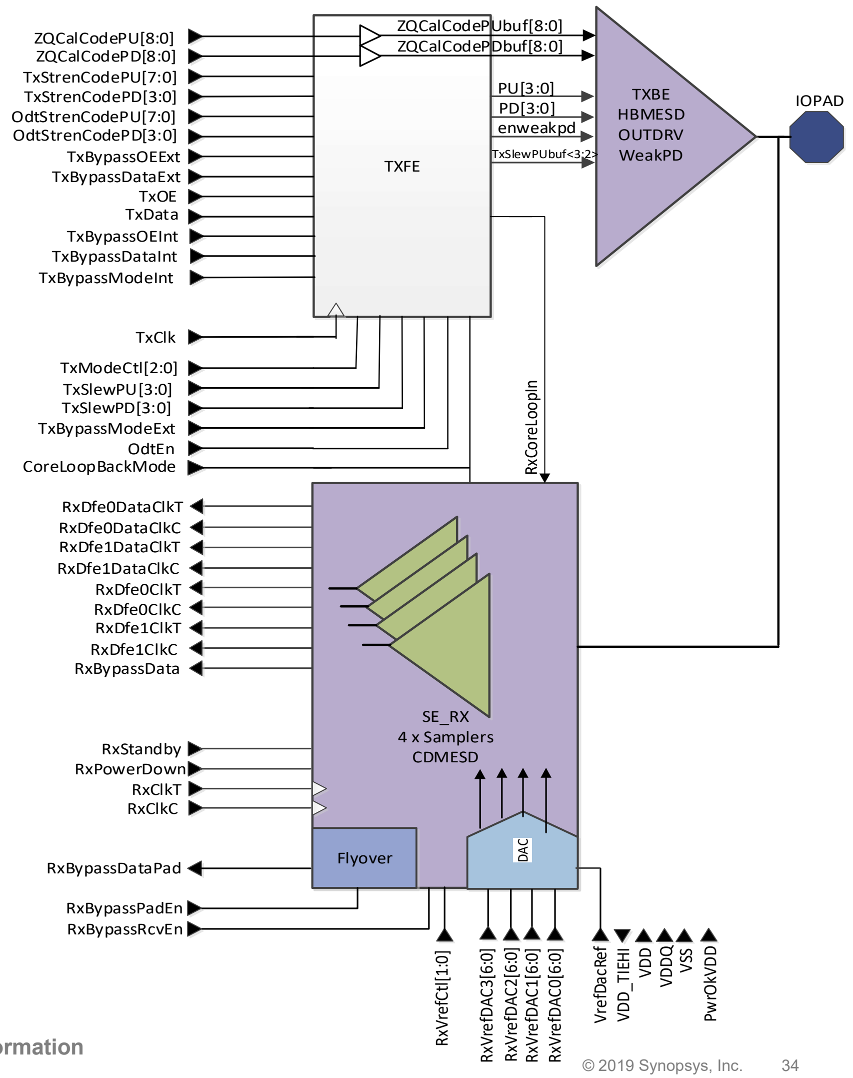


SEC SLICE

LPDDR54 SE IO

Single Ended IO (DQ, DMI, CA, LP4 CS)

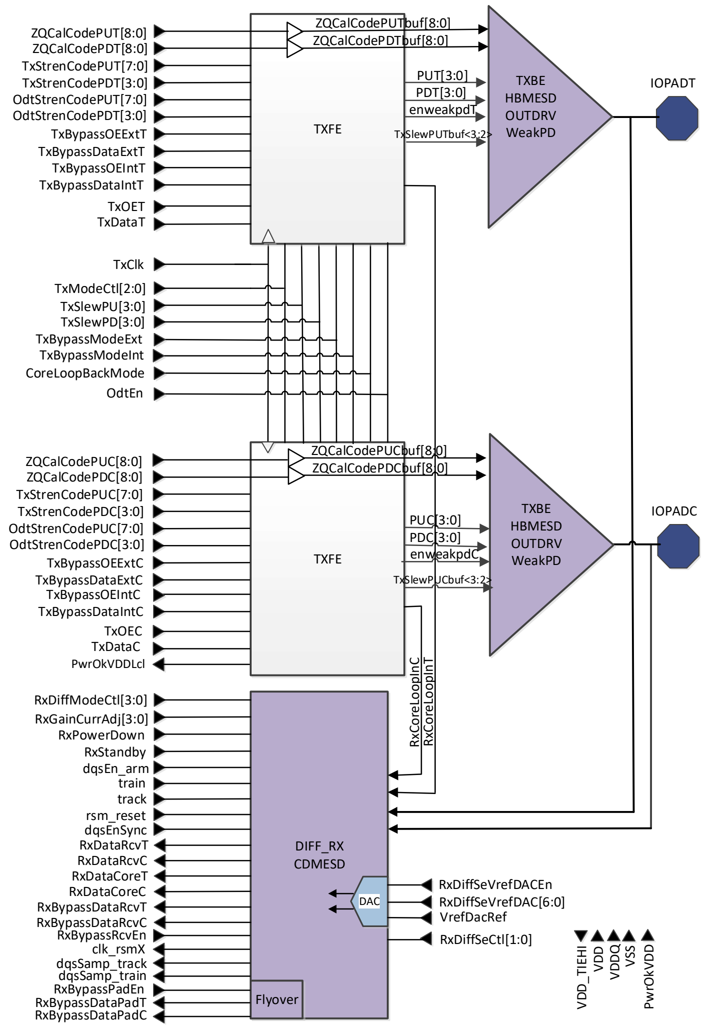
- Complete thin-ox design: VDD/VDDQ ≤ 0.96V
- TXFE – Transmitter Front End
  - VDD domain control logic and interface to the DI
  - Slew rate control (4th Segment Edge Boost)
- TXBE – Transmitter Back End
  - Transmitter BE segments 4x120Ω
  - Pre-drivers with slew rate control option
    - Feed-forward Cap Coupling
  - Output drive/ODT strength-control
  - ESD (HBM D/D2 diodes)
  - Weak pull-down on IOPAD
    - Over voltage protection @ HiZ
- SE_RX – Single Ended Receiver
  - Four sense amp samplers (positive edge) on PAD
  - Four channel 7-bit DAC circuit for VREF generation
    - Dedicated reference voltage “VrefDacRef” to track DRAM VDDQ
    - Step Size < 5mV
  - 1-tap DFE (Decision Feedback Equalization)
    - DFE is done in post-FIFO inside PUB
  - ESD (CDM: R_{ESD}=200Ω, D3/D4 Diodes)
  - Flyover & CoreLoopBack path



LPDDR54 DIFF IO

Differential IO (CK, RDQS, WCK)

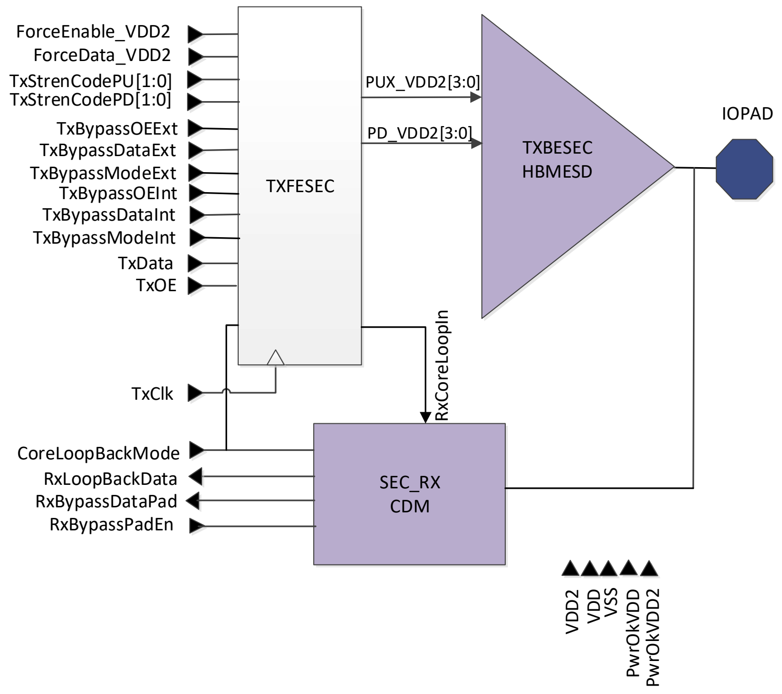
- Complete thin-ox design: VDD/VDDQ ≤ 0.96V
- 2 x TXFE – Transmitter Front End
  - VDD domain control logic and interface to the DI
  - Slew rate control (4th Segment Edge Boost)
- 2 x TXBE – Transmitter Back End
  - Transmitter BE segments 4x120Ω
  - Pre-drivers with slew rate control option
    - Feed-forward Cap Coupling
  - Output drive/ODT strength-control
  - ESD (HBM D/D2 diodes)
  - Weak pull-down on IOPAD
    - Over voltage protection @ HiZ
- DIFF_RX – Differential Receiver
  - Amplifier chain with bias circuitry, followed by CML2CMOS which converts the output to CMOS level
  - A programmable VREF DAC to support single ended strobe for LPDDR5
  - RSM/RDGate logic that controls the read gate
  - ESD (CDM: R_{ESD}=200Ω, D3/D4 Diodes)
  - Flyover & CoreLoopBack path



LPDDR54 SEC IO

Single Ended CMOS IO (LP4/4X CKE, LP5 CS)

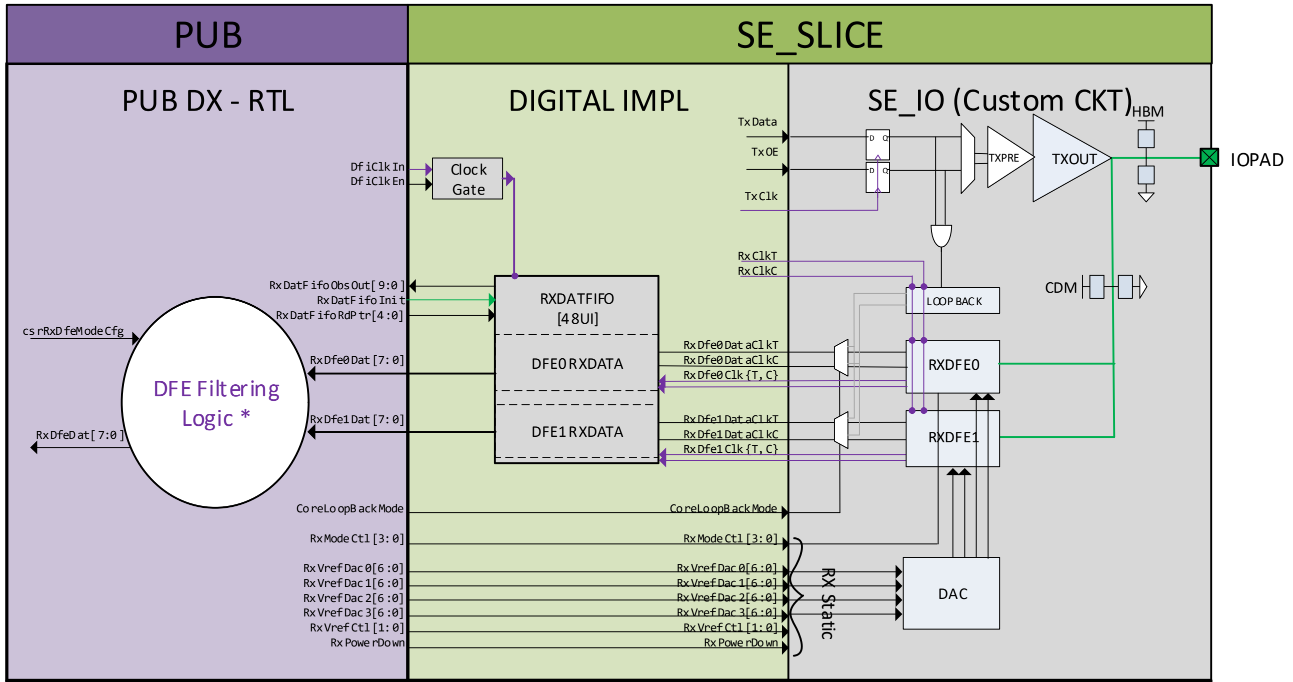
- Design uses:
  - 1.2V devices on VDD2 supply
  - Core devices on VDD supply
- TXFE – Transmitter Front End
  - VDD domain control logic and interface to the DI
  - Driver signal level shifting to VDDQ domain
- TXBE – Transmitter Back End
  - Uncalibrated output driver
  - Supports 4 settings: 50Ω, 66.6Ω,100Ω, 400Ω
  - ESD (HBM D/D2 diodes)
- SEC_RX – Single Ended Receiver
  - CMOS Schmitt Trigger
  - CoreLoopBack path
  - ESD (CDM: 200Ω + D3/D4 Diodes)



LPDDR5/4/4X PHY 1-Tap RX DFE Overview

Making Loop-unrolled DFE run @ 6400 Mbps

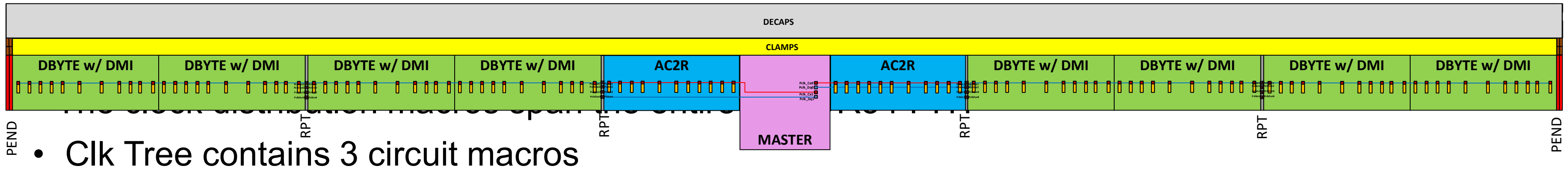
- Customer CKT
  - 4 x Samplers on PAD
  - VREF Generator with 4 programmable VREF values
  - Four Samplers are grouped into 2 channels
- Hard Macro Digital Impl
  - RXDATAFIFO provides data channel 0/1 to PUB for decision making
- PUB Soft-IP
  - Implements DFE Filtering Logic



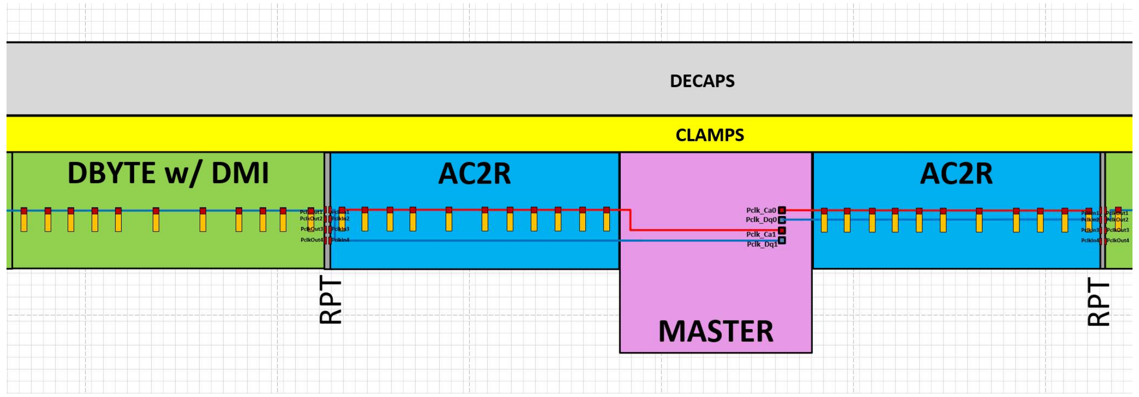
*** DFE Filtering Logic:** Decides which receiver data to use based on the state of the last data bit read.
The rules are:
- First bit in a stream always uses receiver 0
- If the first bit was a 0, use receiver 0 to sample the second bit; if the first bit was a 1, use receiver 1 to sample the second bit
- Similarly, if the second bit was a 0, use receiver 0 to sample the third bit; if the second bit was a 1, use receiver 1 to sample the third bit
- And so on…

CLK Tree

- Pclk_Ca0/1: Used to clock Command/Address SE/SEC/DIFF slices
- Pclk_Dq0/1: Used to clock DBYTE SE/DIFF slices



- Clk Tree contains 3 circuit macros
  1. PCLK_MASTER: Pclk Driver
  2. PCLK_RPT: Pclk Repeater
  3. PCLK_RX: Pclk Receiver



LPDDR5/4/4X PHY: System Clock Configuration

Fast frequency switching without PLL re-lock

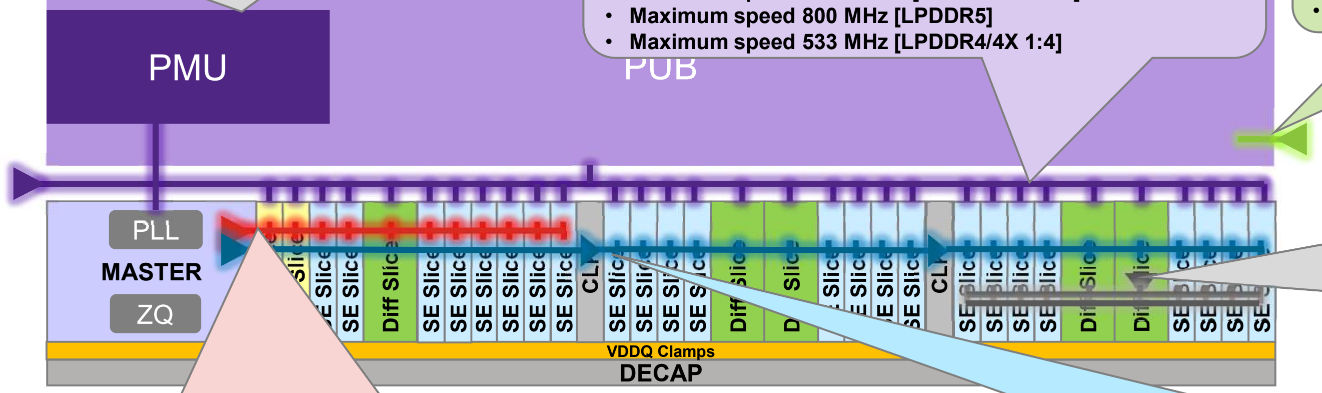
		LPDDR4/4X	LPDDR4/4X		LPDDR5	LPDDR5	Units
DFI	DFI freq ratio	2[3]	2[2]	4	4	2	1:2 or 1:4
	DfiClk freq	1066	533	533	800	800	MHz
PLL	Pllout ratio	4	8	8	8	8	4x or 8x
	pllin	1066	533	533	800	800	MHz
	pllout	4264	4264	4264	6400	6400	MHz
Clock Divider CK and CA	Clock divisor CA	1	2	1	2	2	1/1 or 1/2 or 1/4
	PclkCa	4264	2133	4264	3200[1]	3200[1]	MHz
Clock Divider DQ/DQS	Clock divisor DQ	1	2	1	1	2	1/1 or 1/2 or 1/4
	PclkDq	4264	2133	4264	6400	3200	MHz
DRAM interface rates	CK bit rate	4264	2133	4264	1600	1600	Mbps
	CK frequency	2132	1066	2132	800	800	MHz
	CA bit rate	2132	1066	2132	800	800	Mbps
	DQS bit rate	4264	2133	4264	6400	3200	Mbps
	DQS frequency	2132	1066	2132	3200	1600	MHz
	DQ bit rate	4264	2133	4264	6400	3200	Mbps

Note 1 : PUB must send same data every 2 PclkCa to effectively reduce the bit rate in ½. This avoids need of divde-by 4 logic.

Note 2 : Supported for date rates < 3200 Mbps. This option support fast-switching between DFI 1:2 and 1:4 without PLL re-lock.

Note 3 : This option supports full LPDDR4/4X data rates in DFI 1:2 mode. However, synthesis timing closure at 1066 Mhz DfiClk may be a challenge for controller and PUB. PHY provides CSR configurable (csrDxPipeEn) pipeline registers to help close PHY timing for > 800 MHz DfiClk.

LPDDR5/4/4X PHY: HardIP Clock Distribution



PMUCLK
- Gated portions of DFICLK tree
- Runs at either DFICLK or DFICLK/2 using even/odd control of clock gater
- 533 MHz Max (divided) Speed in 1:2 mode
- 800 MHz Max Speed in 1:4 mode

DfiClk
- Skew balanced with CTS
- All Timing paths to HardIP on DFICLK
    - No other timing from hardIP (no more Ring-Osc)
- Maximum speed 1066 MHz [LPDDR4/4X 1:2]
- Maximum speed 800 MHz [LPDDR5]
- Maximum speed 533 MHz [LPDDR4/4X 1:4]

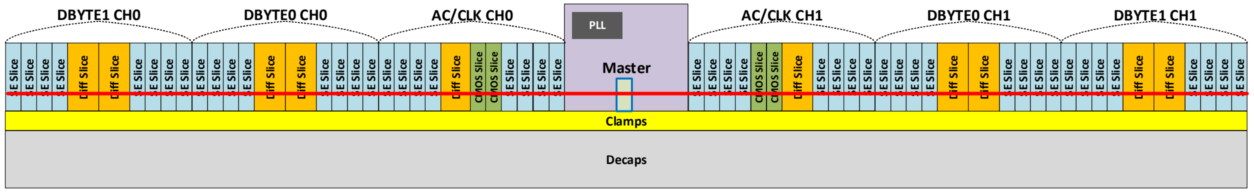
APBCLK
- For PUB APB→DFI Logic only
    - APB requests are synchronized to DFICLK internally in the PUB
- No architectural speed limitations

ByteStrobes
- 2133 MHz Max [LPDDR4]
- 3200 MHz Max [LPDDR5]
- Driven from RDQS_T/C Slice

Pclk_Ca0/1
- Driven From Master
    - 3200 MHz Max speed [LPDDR5]
    - 4267 MHz Max speed [LPDDR4/4x 1:2]
    - Optionally rebuffered
- This clock is stopped when DFI{0,1}_LP_CTRL_REQ & DFI_DRAM_CLK_DISABLE=1
- One channel PHY pictured above
    - Two independently controlled copies when DFI1_EXISTS

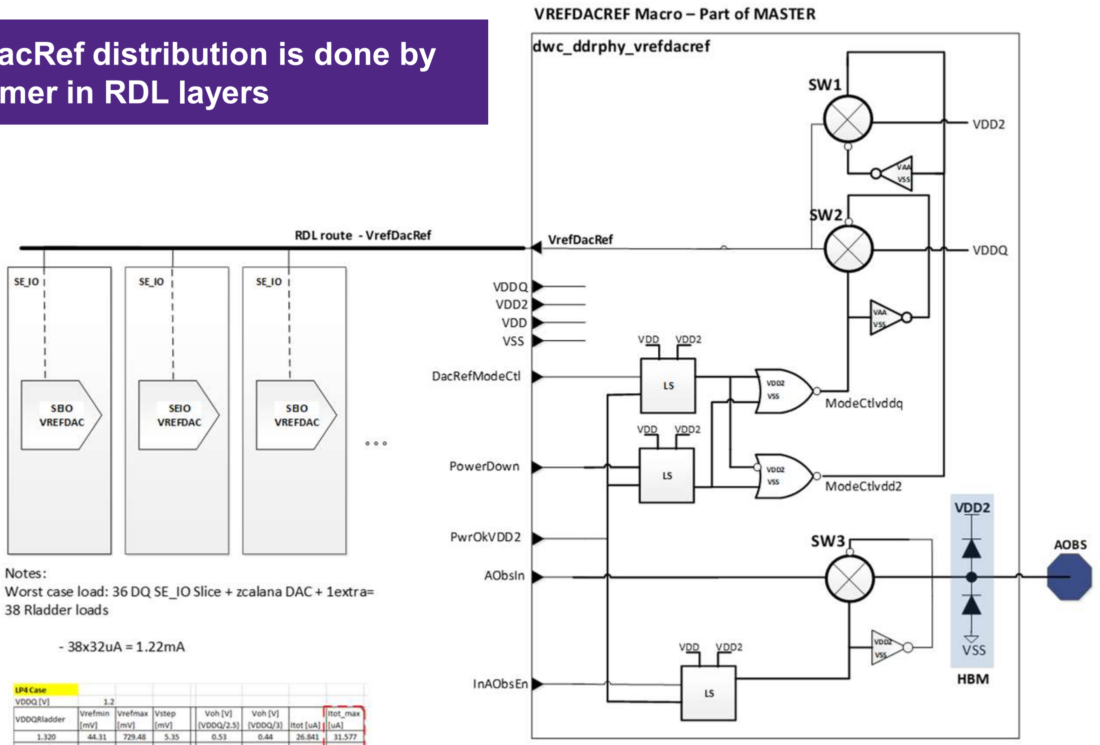
Pclk_Dq0/1
- Driven From Master
    - 6400 MHz Max speed [LPDDR5]
    - 4267 MHz Max speed [LPDDR4/4x]
    - Optionally rebuffered
- This clock is stopped when DFI{0,1}_LP_DATA_REQ
- One channel PHY pictured above
    - Two independently controlled copies when DFI1_EXISTS

VREFDACREF



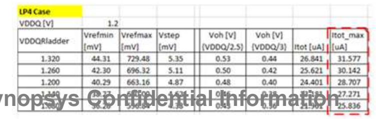
- Generates “VrefDacRef”
  - All the DAC’s in DBYTE SE IO, DIFF IO and ZCALANA are powered by VrefDacRef
  - Power Saving: All the DAC’s in AC SE IO are powered by VDD
- Analog observability path to off-chip
  - PLL analog signal monitoring

VrefDacRef distribution is done by customer in RDL layers



Notes:
Worst case load: 36 DQ SE_IO Slice + zcalana DAC + 1extra= 38 Rladder loads

- 38x32uA = 1.22mA



LP4 Case							
VDDQ [V]	1.2						
VDDQRladder	Vrefmin [mV]	Vrefmax [mV]	Vstep [mV]	Voh [V] (VDDQ/2.5)	Voh [V] (VDDQ/3)	Itot [uA]	Itot_max [uA]
1.320	44.31	729.48	5.35	0.53	0.44	26.841	31.577
1.260	42.30	696.32	5.11	0.50	0.42	25.621	30.142
1.200	40.29	663.16	4.87	0.48	0.40	24.401	28.707
1.140	38.27	630.00	4.62	0.46	0.38	23.181	27.271
1.080	36.26	596.84	4.38	0.43	0.36	21.961	25.836

LPDDR5/4/4X PHY Area and Power

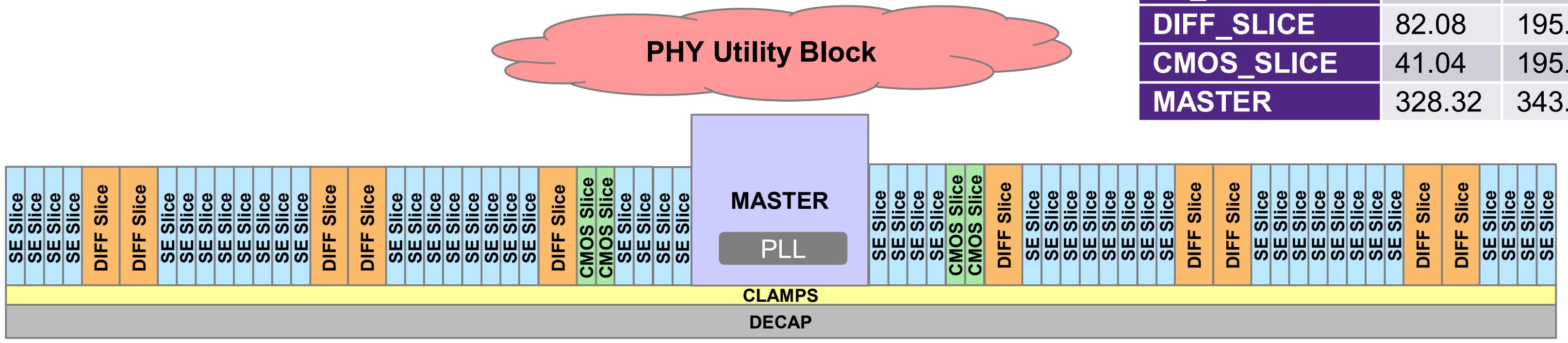
LPDDR5/4/4X PHY Area

TSMC N7 / N7+

- Configuration:
  - 32b PHY; which is two 16b channels
    - i.e., one Master hard-IP and one PUB shared by two PHYs to save area
  - Support for LPDDR5/4/4X
    - WCK and RDQS differential signals for a byte of DQ bits in LPDDR5 mode
  - 2 ranks

- Area
  - Hard-IP (including clamps): ~0.75mm^2
  - PUB: ~1.2 million NAND2s
  - 96KB SRAM (~0.2mm^2)
- Decap: Depends on MIM cap usage and system power distribution network

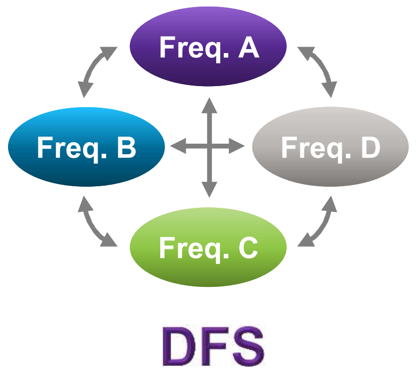
Hard IP (7nm)	X (µm)	Y (µm)
SE_SLICE	41.04	195.12
DIFF_SLICE	82.08	195.12
CMOS_SLICE	41.04	195.12
MASTER	328.32	343.2



DW LPDDR5/4/4X PHY Impedance Controls

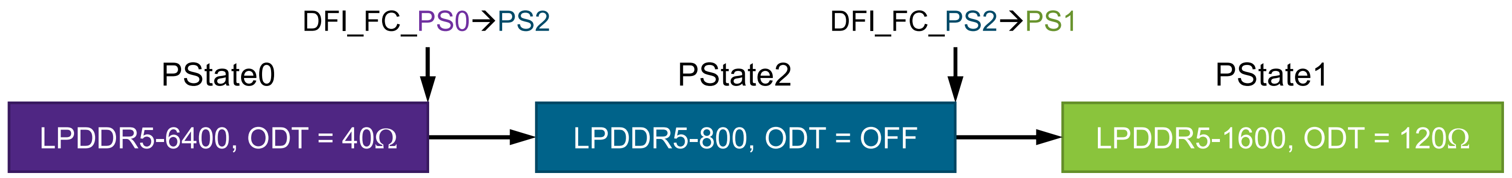
- Driver impedance controls:
  - {30, 40, 60, 120} Ω
- PHY ODT impedance controls
  - {40, 60, 120, OFF} Ω
  - Note: In LPDDR4 mode ODT off not recommended if DRAM can pull bus above VDD
  - Automatic transition between writes and ODT in PHY hard IP
- PHY impedance compensation state machine
  - User programmable compensation duty-cycle/update-interval (for power savings)
- External 120 Ω precision resistor required for driver and ODT impedance compensation

LPDDR5/4/4X PHY DVFS



- LPDDR5/4/4X PHY has multiple active operating states – “PState[0:14]”
  - Each PState can have an independent:
    - Operating frequency, Core voltage, IO voltage
    - PState[7:13] must have data rates half of PState[0:6] respectively.
    - Feature set optimization [e.g. ODT on/off, driver impedance value, preamble settings, etc.]
  - All PStates trained @ boot time only
    - 2D training available in all PStates
  - Fast frequency switching between PState[m] and PState[m+7], where m= 0 to 6
    - Frequency changes supported in-band through DFI_frequency_change protocol
    - Fully hardware accelerates in PHY, no user-software intervention required when switching states
  - All configuration/trained-state stored in the PHY
    - Configuration stored in ECC-protected SRAM, copied to registers during frequency switch by PHY HW engine
      - Saves power and area in PUB and improves timing by reducing the total number of registers in the PUB

Example:



LPDDR5/4/4X PHY Low Power Features

- Other configurable Power Savings features
  - PLL bypassing for low speeds
    - Required below 667Mbps
  - Receivers in low power standby state when not reading
    - Unused instances, in mission mode, powered off
    - Differential receivers powered off if not in LPDDR4 or LPDDR5 strobe mode
  - Programmable PHY-side ODT strength
  - Support for power down and self refresh per rank
  - DM/DBI tri-stated when not needed
  - LPDDR5 Strobe mode disable (expected to be allowed at low speed)

LPDDR5/4/4X PHY Low Power Features

DFS

- LPDDR5 DRAM support for DFS
  - Bank organization can change, improving utilization at 3200 Mbps and below
  - Other analog timings looser at lower speed to enable DRAMs to use less power
  - 1:2 clocking available at 3200 Mbps, enabling similar latency to 6400 Mbps 1:4 operation
    - 6400 Mbps requires 1:4 operation (CK = 800 MHz : WCK = 3200 MHz)
  - All clocks/strobes can operate in Single-Ended mode <= 1600 Mbps
  - RDQS can be disabled completely <= 1600 Mbps
  - “DVFSC” allows DRAM to use a lower core voltage
  - “DVFSQ” allows user to ramp IO voltage down to 0.3 V
    - Requires unterminated operation and ZQ calibration to be disabled

LPDDR5/4/4X PHY Low Power States (1/2)

PHY Low Power State	Exit Latency	PHY Low Power Actions	DRAM Commands Available	Comments
DFI_LPDATA	T_{lp_wakeup} = 2 DFICLKs	PCLK_DQ clock gated PUB DX clock gated Receivers are in stand-by	All commands (available other than data bearing transactions)	
DFI_LPCTRL	T_{lp_wakeup} = 2 DFICLKs	DFI_LPDATA + CA slices idle MEMCLK running PCLK_CA running	DES Maintain Power-down Maintain Self-Refresh Maintain SR+PD	
DFI_LPCTRL_1	T_{lp_wakeup} = 4 DFICLKs	DFI_LPCTRL + MEMCLK stopped PCLK_CA clock gated	DES Maintain Power-down Maintain Self-Refresh Maintain SR+PD	Best Dynamic Power savings with uMCTL5 and LPDDR4 DRAMs

LPDDR5/4/4X PHY Low Power States (2/2)

PHY Low Power State	Exit Latency	PHY Low Power Actions	DRAM Commands Available	Comments
LP2 – Fast Standby	3µs + Retraining time	DFI_LPCTRL_1 + DFICLK stopped ZQ/LCDL Calibration off PLL in Standby	Maintain SR+PD	Best SOC-Directed Power Savings option Re-uses DFI-frequency protocol and uMCTL2 HWFFC
LP3 – Standby	LP2 exit + 10µS (PLL pwron)	LP2 + PLL in Power-down	Maintain SR+PD	Extra power savings compared to LP2 even if VDD is not removed
PHY IO Retention	LP3 exit + Register Restore time	LP3 + VDD gated off VDDQ for DBYTEs optionally power gated off	Maintain SR+PD	CKE*/CS* driven to zero, MEMRESET held at previous state. All other Pins are tri-stated.

LPDDR5/4/4X PHY Power

LPDDR5 mode in TSMC N7/N7+ Power Estimate

- LPDDR5 mode, 6400Mbps, 2 channels, 16 bits/channel, 2 rank system, 40Ω TX, 60Ω Term, TT, 25C, and EQ off

Rail	Voltage (V)	Read Power (mW)	Write Power (mW)	Idle Power (mW)	LP_DATA Power (mW)	LP_CTRL Power (mW)	LP2 Power (mW)	PHY IO Ret. Power (mW)
VDD	0.75	275	356	146	88	8.5	1.0	0
VDDQ	0.5	21	79	3	3	0.1	0.1	0.1
VAA_VDD2	1.05	26	26	9	9	9.4	0	0
Total Power		**322**	**462**	**158**	**100**	**18.0**	**1.1**	**0.1**

- Power is hard-IP blocks only w/ leakage, does not include power of PUB/uMCTL2
- Toggle rate is 25%. Power represents current drawn from the PHY's supplies
- All numbers are estimates and subject to change

LPDDR5/4/4X PHY Power

LPDDR4X mode in TSMC N7 Power Estimate

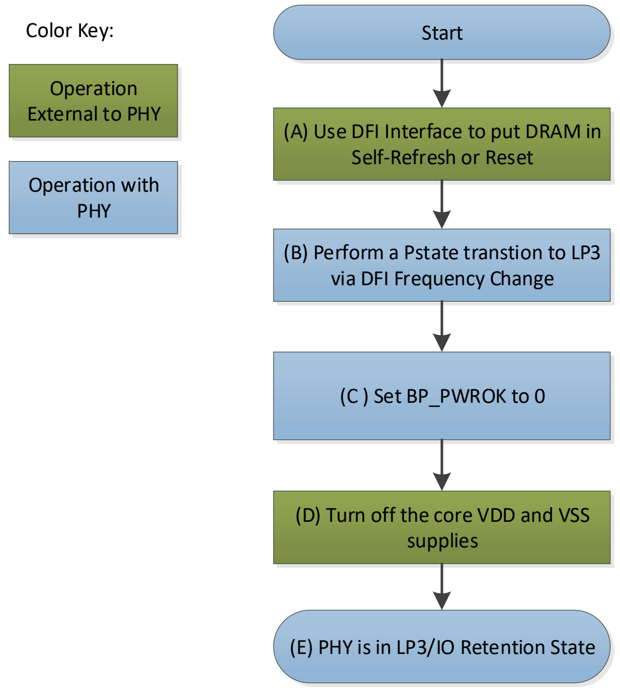
- LPDDR4X mode, 4267Mbps, same data byte config as area slide (32 DQ), 1 rank system, 40Ω TX, 60Ω Term, TT, 25C, and EQ off

Supply Rail	Voltage (V)	Read Power (mW)	Write Power (mW)	Idle Power (mW)
VDD	0.75	225	283	135
VDDQ	0.6	27	109	13
VAA_VDD2	1.1	10	10	10
Total Power		**262**	**402**	**158**

- Power is hard-IP blocks only w/ leakage, does not include power of PUB/uMCTL2
- Toggle rate is 50%. Power represents current drawn from the PHY's supplies
- All numbers are preliminary estimates and subject to change

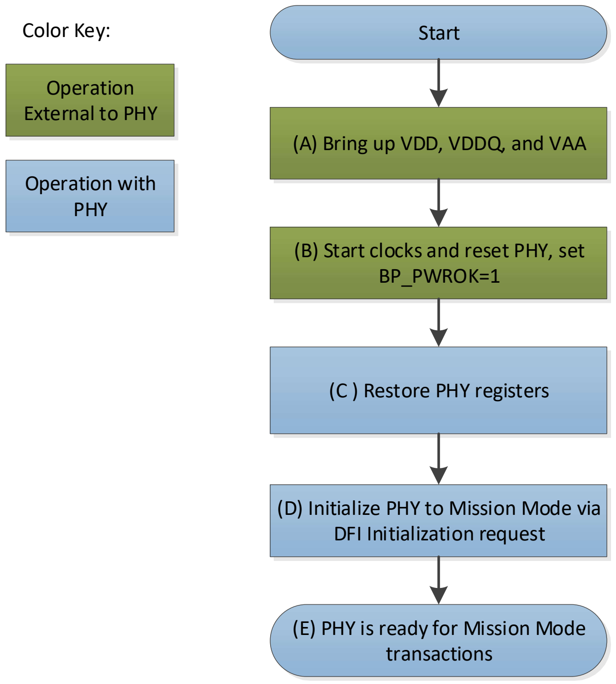
LPDDR5/4/4X PHY: LP3/Retention Entry Sequence

- Use the DFI interface to put the DRAM into Self-Refresh Power Down.
- Transition the PHY to the LP3 / IO Retention state by using a DFI Frequency Change operation.
- Set the BP_PWROK signal to 0 or Assert Reset input to 1.
  - This will enable the data retention feature on the CKE and MEMRESET_L pins.
  - CKE will be held at 0; MEMRESET_L will be held at previous state (0 or 1).
  - Remaining PHY input signals must be valid, inactive states for at least 10ns after the BP_PWRK asserts 1 -> 0.
  - The PHY input clocks are not required to toggle when BP_PWROK=0.
- Turn off the core VDD and VDDQ supplies. These supplies can be turned off in any order. The VDD2H power supply must remain active during LP3/IO Retention.
- All PHY input values are now don't cares


Color Key:
Operation External to PHY
Operation with PHY
Start
(A) Use DFI Interface to put DRAM in Self-Refresh or Reset
(B) Perform a Pstate transtion to LP3 via DFI Frequency Change
(C) Set BP_PWROK to 0
(D) Turn off the core VDD and VSS supplies
(E) PHY is in LP3/IO Retention State

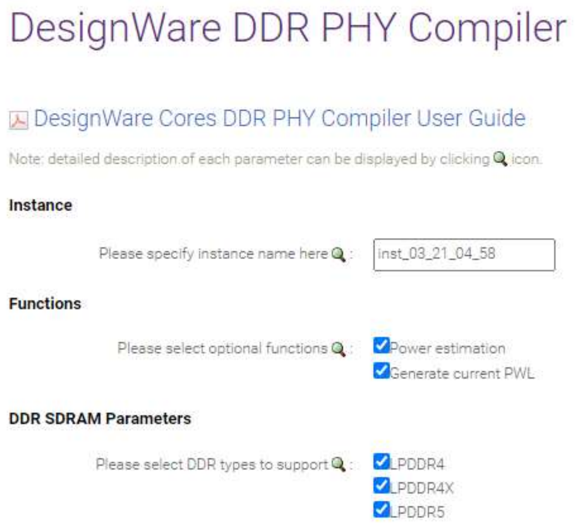
LPDDR5/4/4X PHY: LP3/Retention Exit Sequence

- The VDD2H power supply will already be active, restore the core VDD and VDDQ power supplies similar to Step A in “PHY Initialization”
  - Ensure all VDD-level signals are stable
- Start clocks and reset the PHY similar to Step B in “PHY Initialization”.
  - Assert BP_PWROK input.
    - BP_PWROK input should be asserted any time after Reset input is de-asserted.
    - The csrPwrOkDly can be asserted before or after assertion of BP_PWROK input.
    - The {TxBypassModeEn*, WRSTN} signals may be undefined at VDD power-on, but must be driven LOW at least 10ns prior to the asserting edge of BP_PWROK.
- Restore all PHY registers.
  - Refer to PUB databook for details
- Initialize the PHY to mission mode through a DFI initialization request. This step is similar to Step J of “PHY Initialization”
- The PHY is ready for mission-mode transactions


Color Key:
Operation External to PHY
Operation with PHY
Start
(A) Bring up VDD, VDDQ, and VAA
(B) Start clocks and reset PHY, set BP_PWROK=1
(C) Restore PHY registers
(D) Initialize PHY to Mission Mode via DFI Initialization request
(E) PHY is ready for Mission Mode transactions

LPDDR5/4/4X PHY Power Estimate

- Please use PHY compiler for power estimates specific to your PHY configuration, foundry and process node.
- https://www.synopsys.com/dw/ddrphy.php



DesignWare DDR PHY Compiler

DesignWare Cores DDR PHY Compiler User Guide

Note: detailed description of each parameter can be displayed by clicking icon.

Instance

Please specify instance name here : inst_03_21_04_58

Functions

Please select optional functions : [✓ Power estimation] [✓ Generate current PWL]

DDR SDRAM Parameters

Please select DDR types to support : [✓ LPDDR4] [✓ LPDDR4X] [✓ LPDDR5]

LPDDR5/4/4X Automotive Features

LPDDR5/4/4X PHY: Automotive Features

- Anti-valent implementation for safety critical Interrupts
- Triple Mode Redundant (TMR) for critical control signals
- Increased DFT coverage
  - Greater than 99% stuck-at
  - Greater than 90% at-speed (TDF)
- ISO 26262 ASIL B Ready
- AEC Q100 qualified Automotive Grade 2 (AP)

LPDDR5/4/4X PHY: Safety Critical Interrupts

- PHY has following 3 interrupt sources which are safety critical.
  - ACSM SRAM Parity Error
  - PState DMA Parity Parity Error
  - RxFIFO Check Error
- Each of the above interrupts has been implemented using anti-valent flip-flops for reliability.
- Each of the above interrupts has 2 bit “fault[1:0]” outputs available as PHY_TOP pin for SOC to monitor the status.
  - 2’b10 :: no fault state
  - Anything else :: some fault.
- Each of the above interrupts can’t be masked. However, they have 2 CSR controls:
  - Clear :: when asserted, forces the “fault[1:0]” output pins as 2’b10.
  - Override :: when asserted, forces the fault[1:0] output pins as 2’b01.
    - This is for testing purposes.

LPDDR5/4/4X PHY: Pins & Status

- PHY_TOP has 6 outputs pins: PhyInt_fault[5:0]
  - PhyInt_fault[1:0] : Active High unmask-able ACSM Parity Error anti-valent fault ; 2'b10 :: no fault.
  - PhyInt_fault[3:2] : Active High unmask-able PState DMA Parity Error anti-valent fault ; 2'b10 :: no fault.
  - PhyInt_fault[5:4] : Active High unmask-able RxFIFO Check Error anti-valent fault ; 2'b10 :: no fault.

- No Status CSR provided for PhyInt_Fault[5:0].

Triple Mode Redundancy (TMR) for Critical Control Signals

- Triple Mode Redundancy (TMR) insertion is to be included while **implementing the PUB components of the IP** for flip-flops that perform critical control functions, including enabling/disabling the microcontroller subsystem.
  - This ensures that these critical functions continue to operate as intended, even in the presence of single-bit soft error
  - The full list of flops to be implemented with TMR is as follows:
    - PUBMODE[0].HwtMemSrc
    - MicroContMuxSel[0]. MicroContMuxSel
    - ContextToMicro[0]. ContextToMicro
    - MicroReset[3].ResetToMicro
    - MicroReset[0].StallToMicro
    - SequencerOverride (all bits)
    - PieVecCfg (all bits)
    - PieInitVecSel (all bits)

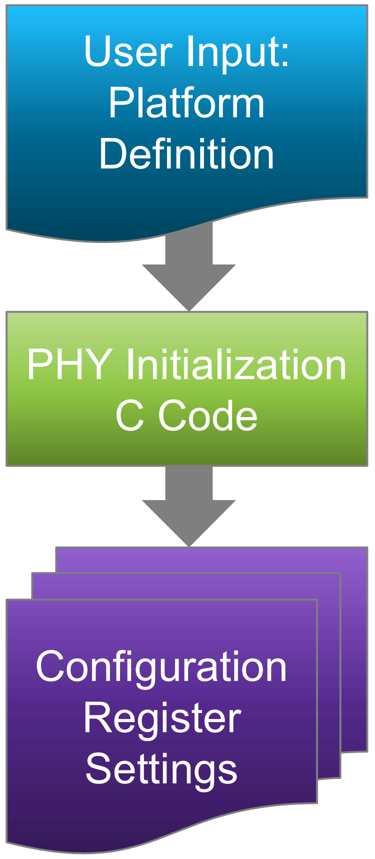
LBIST Support

- What is LBIST ?
  - In-system self-test to meet functional safety requirement as per ISO 26262.
  - Utilizes scan chains and a small amount of logic added by a DFT tool such as Synopsys DFTMax.
  - Uses pseudo-random pattern generator (PRPG) to create scan data, and a multiple-input signature register (MISR) to compare the design response to an expected value.
    - At the end of the test, if the actual signature matches the expected signature, a PASS signal will be asserted.
- LBIST Controller must be inserted in SOC while **synthesizing PUB component of the IP**.
  - PHY define DWC_DDRPHY_LBIST_EN should be defined.
  - PHY provides additional ports required for enabling LBIST functions.
- LBIST simulation can be performed at 2 levels:
  - With only PUB scan chains
  - With PUB + HardIP scan chains
- An appnote describing LBIST implementation flow will be provided.

LPDDR5/4/4X PHY Firmware

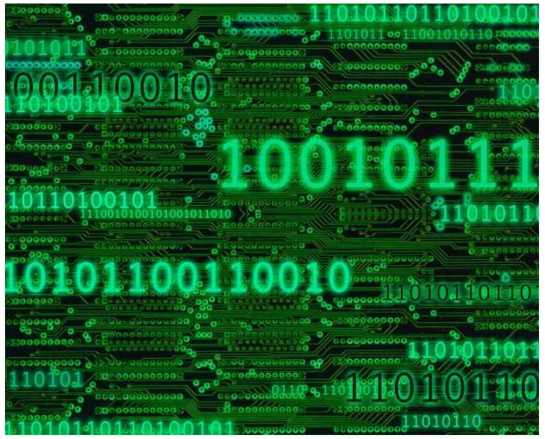
LPDDR5/4/4X PHY Register Programming Automation

- PHY Initialization C Code (“PHYINIT”)
- Precisely sets CSRs and firmware message block parameters based on platform definition
- Delivered in machine-usable formats to help get the simulations running sooner
- Synopsys internal verification uses same initialization process in our testbench environment
- Allows Synopsys to update the initialization process to customers in a consistent way that does not require significant integration work in the consuming SOC


User Input: Platform Definition
PHY Initialization C Code
Configuration Register Settings

LPDDR5/4/4X PHY: Firmware Overview

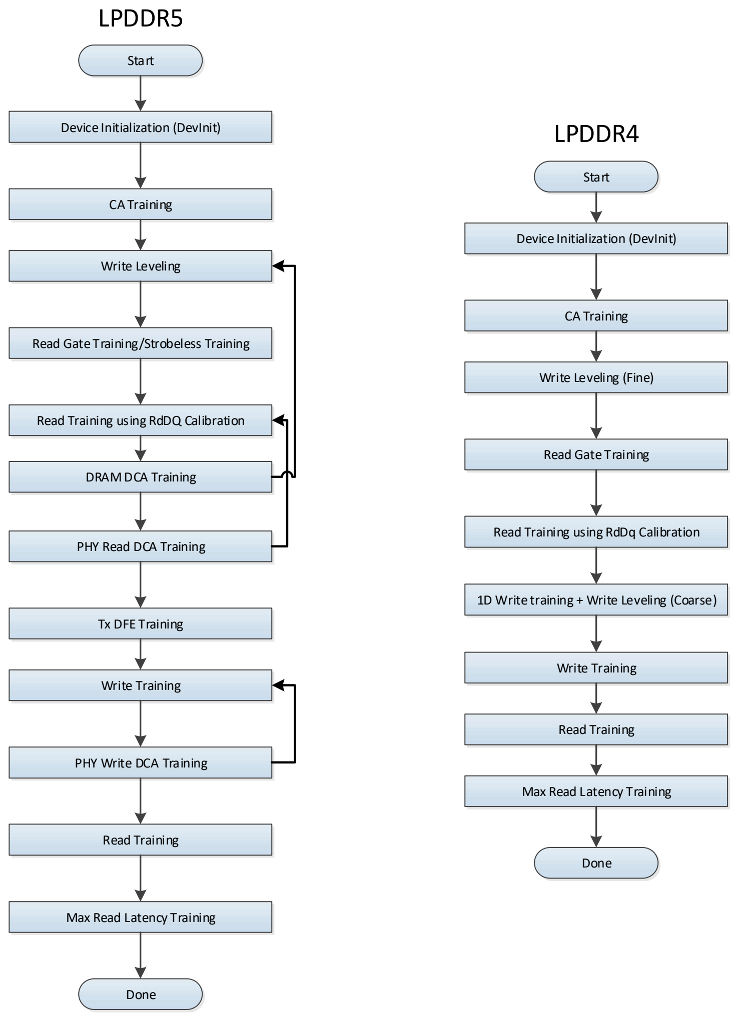
- The following firmware supports the LPDDR5/4/4X PHY
  - Combined 1D+2D training firmware
    - LPDDR4 / LPDDR4X
    - LPDDR5
    - Single combined 1D+2D image eases customer use model
    - 2D training (including DFE) enabled in all PStates
  - Quickboot firmware option
    - Restores training results from initial bootup
    - Enables quick retraining, and dram initialization
  - ATE test firmware
  - Diagnostic firmware (“diags”)
    - Includes 2D eye mapping utility
    - Includes Write Leveling margin utility (LPDDR4/4X)
    - Includes Read Gate margin utility
- Same proven Firmware access/programming mechanism as previous generation
  - Improved documentation, user-guides, troubleshooting manual



High Level Boot-Time Training Flow

How the DRAM is trained

1. DevInit – allows the system to come out of reset and send MRs at boot frequency
2. CA Training – trains CA pins so DRAM commands can be sent
3. Write Leveling – trains DQS fine for LPDDR4 and WCK fine and coarse for LPDDR5
4. Read Gate / Strobeless – trains Rx receiver to find read burst start
5. Read Training RdDQ – trains Vref DQ and delay to do reads with fixed patterns
6. DRAM DCA (LPDDR5) – optimize duty cycle for DQ on DRAM
7. PHY read DCA – optimize duty cycle for DQ reads on PHY
8. Write Leveling Coarse (LPDDR4) – train DQS coarse to align with DQ for writes, and also preliminary TX DQ training
9. Tx DFE – optimize DRAM write DFE
10. Write Training – optimize DQ delays and DRAM Vref for writes
11. PHY Write DCA - optimize duty cycle for DQ writes on PHY
12. Read Training – optimize DQ delays and PHY Vref for reads
13. MRL training – optimize DFI max real latency


LPDDR5
Start
Device Initialization (DevInit)
CA Training
Write Leveling
Read Gate Training/Strobeless Training
Read Training using RdDQ Calibration
DRAM DCA Training
PHY Read DCA Training
Tx DFE Training
Write Training
PHY Write DCA Training
Read Training
Max Read Latency Training
Done
LPDDR4
Start
Device Initialization (DevInit)
CA Training
Write Leveling (Fine)
Read Gate Training
Read Training using RdDq Calibration
1D Write training + Write Leveling (Coarse)
Write Training
Read Training
Max Read Latency Training
Done

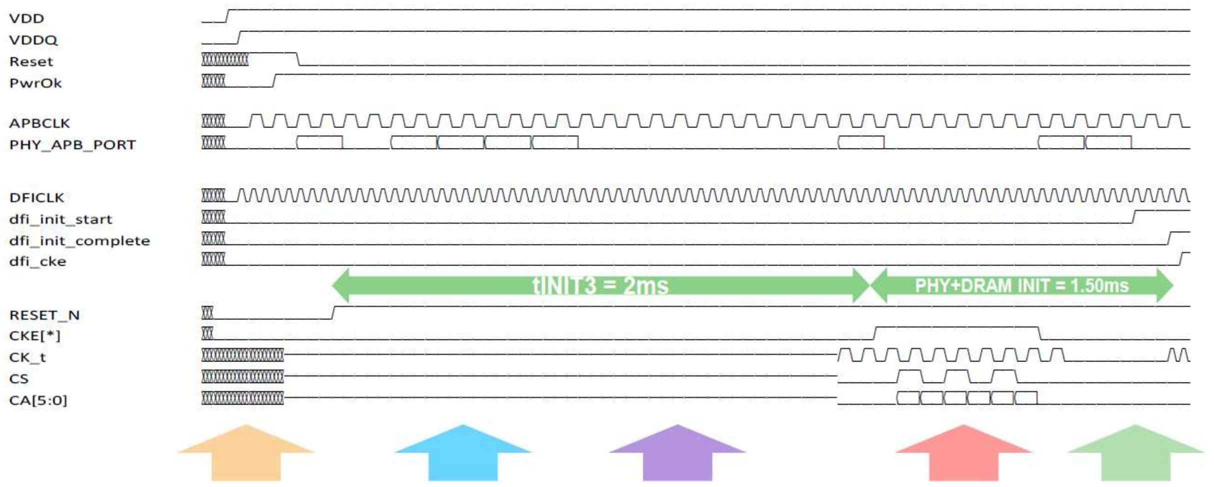
Quickboot + Retraining

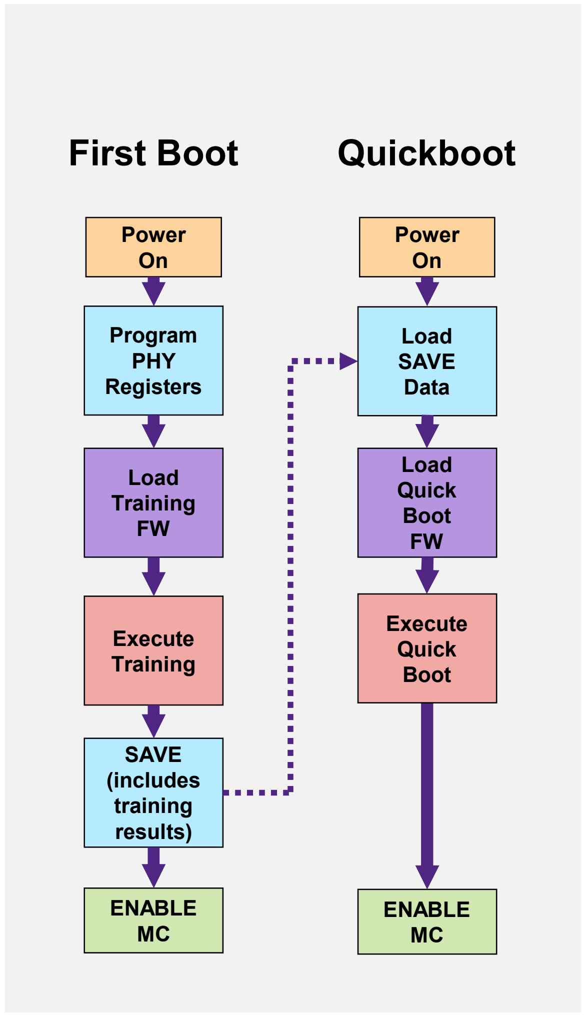
- For some applications, full robust training duration may be too long - in such cases, the recommend flow is
  - Run the full boot training firmware once at initial bootup and save the training results
  - For subsequent bootups, run Quickboot firmware, duration ≈ 3.5ms
- The “Quickboot + Periodic Retraining” flow is used by
  - Multiple LPDDR4X multiPHY and LPDDR4 multiPHY V2 customers
  - Multiple LPDDR5/4/4X PHY customers, including automotive applications
- “Quickboot + Retraining” approach has been verified in Synopsys lab

DW LPDDR5/4/4X PHY Quickboot

Reduces training times to ~3.5ms

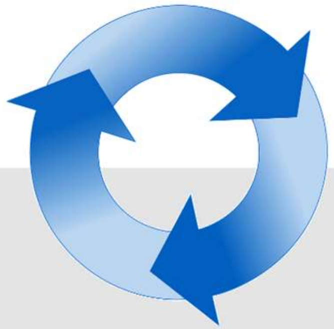
- Full training is only required at first boot
  - PHY state can be SAVED after training
- Fast initialization supported through Quickboot firmware sequence
  - DRAM initialization @ LPDDR4/5 boot frequency
  - Compensation of LPDDR4/4X parameters {tDQSCK, tDQS2DQ}
  - Compensation of LPDDR5 parameters {tWCKDQI, tWCKDQO}





DW LPDDR5/4/4X PHY Periodic Retraining (PPT)

Required to compensate for PHY delay line and SDRAM delay shift



- Periodic Retraining (aka “PPT”) compensates following drifts
  - Drift in PHY IO impedance
  - Drift in READ interface (mostly in the PHY)
    - Drift from DRAM tDQSCK/tWCKDQO, PHY internal delays in read strobe path and any delay in package/routing
    - Retraining (PPT) can compensate for +/- 5UI drift
      - This range is large enough to cover full voltage, temperature and aging ranges
    - Retraining is performed during LP3/Retention exit as well as during mission mode periodically
  - Drift in WRITE interface (mostly in the SDRAMs)
    - Retraining (PPT) compensates for full range of drift specified in JEDEC specification for tDQS2DQ/tWCKDQI
    - Retraining uses SDRAM’s built-in oscillator to determine amount of drift
    - Retraining is performed during LP3/Retention exit as well as during periodic mission mode
- Periodic DFI update cycles compensate drift in PHY’s internal delay line delays
- PPT is always required irrespective of whether system is using full boot or Quickboot

New & Notable in LPDDR54

- 1D and 2D images will now be condensed into one image due to increases in IMEM and DMEM
- LP5 Interface has changed with inclusion of WCK (Write CK generated in PHY)
- WCK must be synchronized using special clock startup sequence (static, toggle, fast)
- WCK can be set to always on mode
- Training can run on up to 15 pstates
  - In 15 Pstate mode all csr writes are to HW pstate 0
- Support single-ended CK/WCK/RDQS per Pstate
- Support DFI in 1:2 or 1:4 mode
- We always train LP4 in 1:4 mode
- Coarse = 1UI
- Fine = 1/64 UI
- Debug messaging is the same as V2 PHY
- Message block provides user input and output

	Ratio (DFI:CK: DATA_STROBE)	dfiClk (MHz)	memClk (CK) (MHz)	WCk/DQS (MHz)	Data Rate (Mbps)
Lp5	1:1:2	800	800	1600	3200
Lp5	1:1:4	800	800	3200	6400
Lp4	1:2:2	1067	2133	2133	4267
Lp4	1:4:4	533	2133	2133	4267

LPDDR5/4/4X PHY: Initialization Training Summary

- A single FW image has all the following training steps

Training Steps	Description	Per Bit	Per Byte	Per Rank	Per Pstate
Command Bus Training (CBT) – Address delay	Command/Address delay to center clock in middle of data eye	✓		Rank 0 only	✓
Command Bus Training (CBT) – Vref	VREF optimization for input receivers	✓		✓	✓
Read DQS Gate	RxEn delay		✓	✓	✓
Duty Cycle Adjuster (DCA) Training (LPDDR5 mode only)	For WCK duty cycle correction using DRAM Duty Cycle Monitor (DCM) as well as PHY internal DCA circuit	✓		✓	✓
WCK2CK Leveling (a.k.a., Write Leveling in LPDDR4 mode)	To compensate CK-to-WCK timing skew	✓		✓	✓
Write Training	TxDQ/DMI delay, VREF and DRAM DFE	✓		✓	✓
Read Training	RxDQ/DMI delay, VREF and PHY DFE	✓		✓	✓
Link ECC Training (LPDDR5 mode only)	TxDQS (Link ECC) delay		✓	✓	✓
Read Latency Training	To find the smallest read latency that works for all RD FIFOS in the PHY, limiting system memory latency		✓	✓	✓

LPDDR5/4/4X PHY: Periodic Phase Training (PPT)

- Periodic Training is for DRAM drift compensation due to Voltage/Temperature (VT) variations in the DRAM
- **LPDDR4/4X Mode**:
  - Write DQ to DQS Training
  - Read Gate Training
- **LPDDR5 Mode:**

Training Steps	Description	Per Bit	Per Byte	Per Rank	Per Pstate
tWCK2DQO Drift [LPDDR5 Read PPT]	• For detecting the drift in DRAM tWCK2DQO parameter and adjusting the Read DQS Gate delays dynamically (RxEN delay)		✓	✓	✓
tWCK2DQI Drift [LPDDR5 Write PPT]	• For detecting the drift in DRAMs tWCK2DQI parameter and adjusting the WDQ delays dynamically (WDQ delay)	✓		✓	✓

LPDDR5/4/4X PHY: Options for Initiating Periodic Phase Training

- 3 options for initiating PPT
  - DFI compliant options
  - Proprietary option (with SNPS LPDDR5/4/4X PHY & Controller), less bandwidth loss

PPT Options	Description	Duration
PHY initiated through a programmable timer	• Upon timer expiration, PHY request MC to relinquish DFI and Memory interface using DFI Master Interface • PHY starts DRAM WOSC/ROSC, reads the results, calculate the drift and updates the WDQ/RXEn LCDL	~2000 DFICLK
Controller initiated through DFI Frequency change interface	• Controller can trigger PHY to perform training using Frequency Change interface • PHY request MC to relinquish DFI and Memory interface using DFI Master Interface • PHY starts DRAM WOSC/ROSC, reads the results, calculate the drift and updates the WDQ/RXEn LCDL	~2000 DFICLK
SNPS Controller assisted Drift Tracking through DFI MRR snoop interface	• Controller starts the WOSC/ROSC oscillators on its own through MPC commands in the middle of normal traffic • Controller reads the Oscillator count values and sends a hint to the PHY to snoop the DRAM read responses • PHY calculate the drift and updates the WDQ/RXEn LCDL during the next DFI Update event	**~30 DFICLK**

LPDDR5/4/4X PHY: ATE Test Firmware

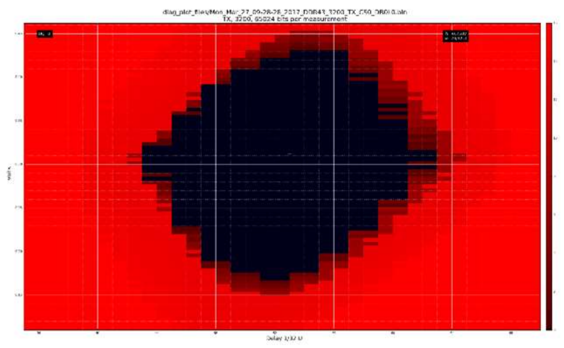
- Firmware image provided to perform functional testing of the PHY on Automated Test Equipment (ATE)
- Allows fire and forget testing of the LPDDR PHY
  - LPDDR PHY can be tested in parallel with other portions of the SOC allowing LPDDR PHY test time to be hidden
  - Load the firmware, start the tests, and come back later to get the results
- Functional tests include:

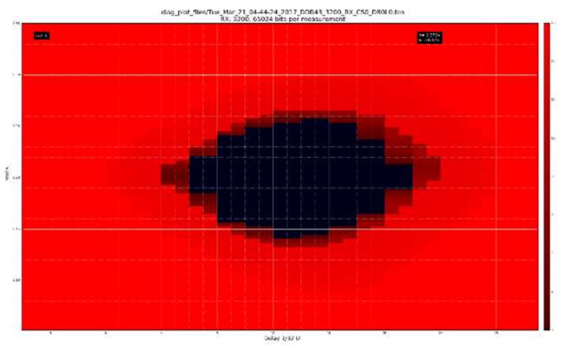
• Impedance Calibration	• CA Loopback 1D/2D Eye
• PLL Lock	• Data Loopback 1D/2D Eye
• LCDL Linearity	• Core or Pad loopback option

LPDDR5/4/4X PHY: Diagnostic Firmware

- LPDDR5/4/4X 2D eye mapping diagnostic firmware is available
- Allows measuring 2D eye at both DRAM and host receivers
- Allows quick debug of channels to identify S/I issues in systems
- Shows eyes in SiP/POP when probing is impossible

Example Output

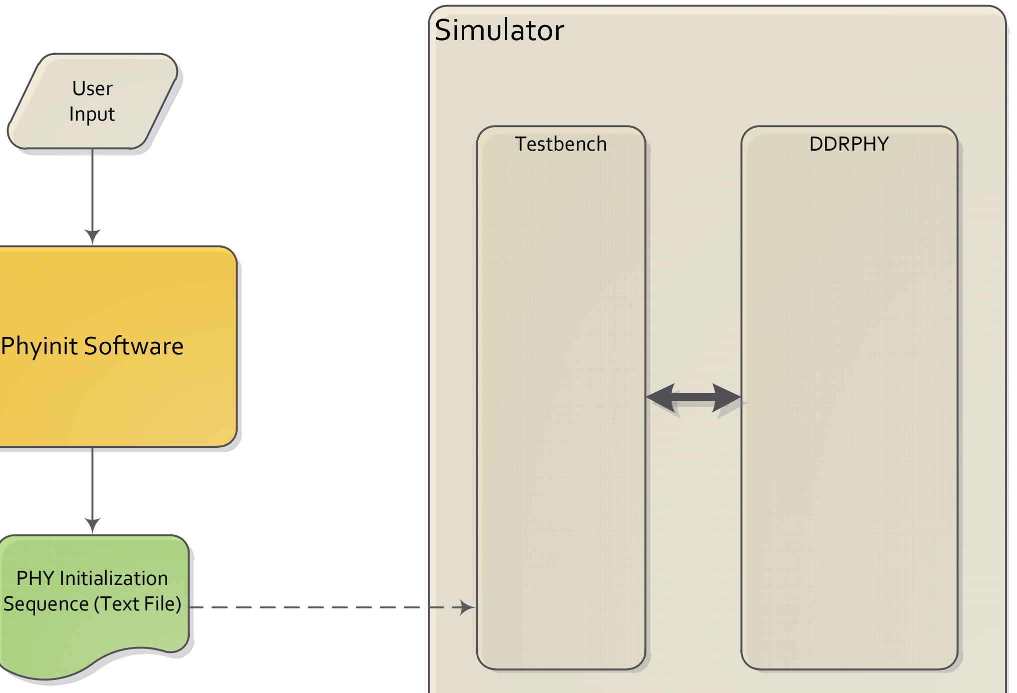

LPDDR4 2D Eye @ DRAM RX


LPDDR4 2D Eye @ PHY RX

PHYINIT

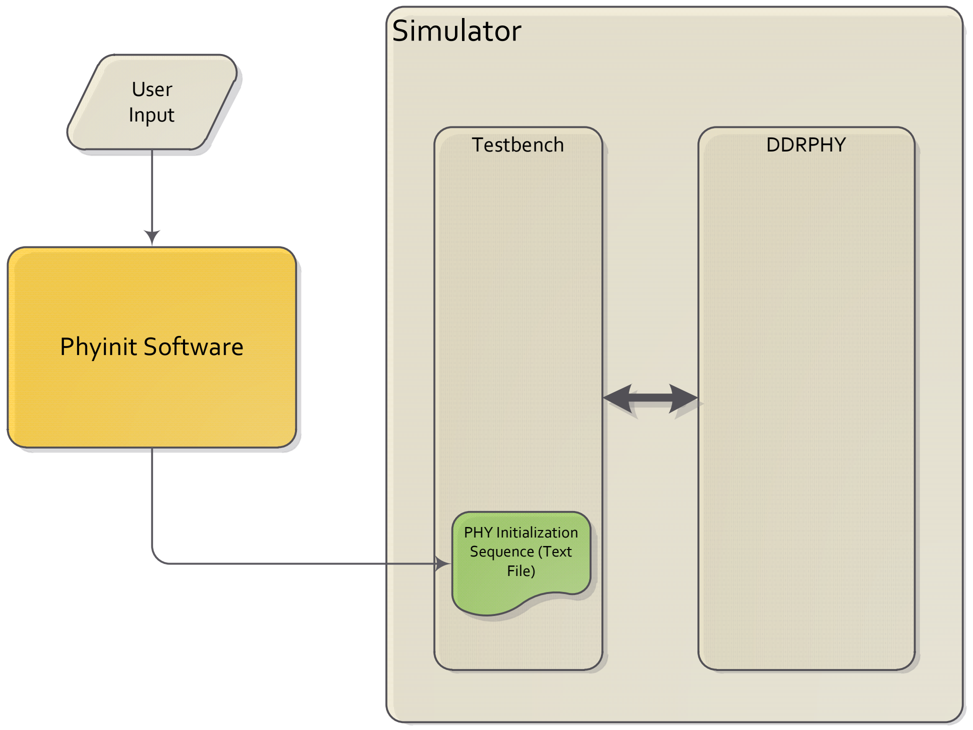
What is Phyinit?

- A Software to produce the sequence of events for PHY initialization specific to customer's configuration
- Relieves customer from understanding **Most** the CSR programming to prepare PHY for mission mode
- Phyinit is written in C
- User enter PHY configuration and parameters in a text file
- Phyinit output a text file containing sequence of APB writes to initialize PHY



Usage Model

- User runs Phyinit as a Standalone software and get a text output
- Output text file is Verilog syntax compatible
- User can either:
  - Write own script to parse output text and feed sequence into their testbench
  - Include output text directly in their testbench



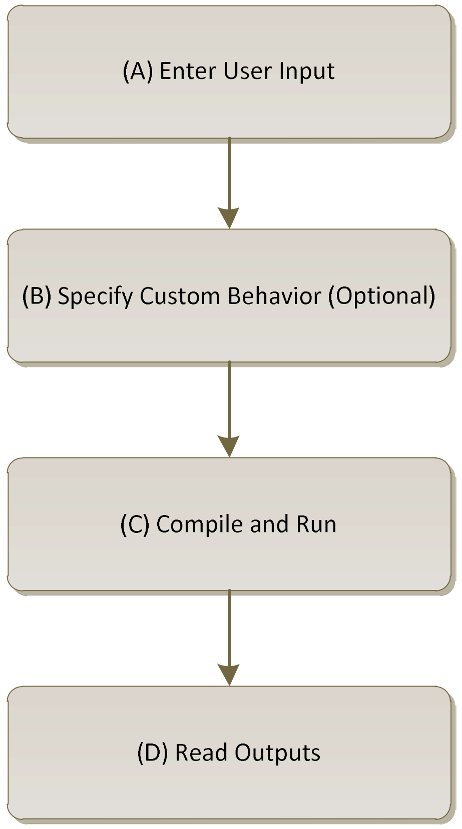
LPDDR54 Major Changes

- LPDDR54 Supports two Modes of configuration:
  - 2 PSTATE Mode
  - 15 PSTATE Mode
    - Requires external PS SRAM to store PHY register data.
- The initialization sequences are significantly different between two modes.
  - Detailed diagram no longer provided in PUB databook and moved the HTML generated diagram.
- PhyInit abstracts out the sequence differences to provide customers with consistent setup.
- Frequency and State Changes in 15 PState
  - New frequency encoding and limitations documented in PUB.
  - In P15 mode custom register programming must be identical for all Pstates.
    - I.e. if user programs register A in Px, it must also be programed in all PSTATES.

Phyinit Directory Structure

```
<release path>/software
         |
         |-README # Quick instructions to run PhyInit
         |-<protocol>/ # Includes all source code and output for given protocol
                  |
                  |-Makefile # Includes targets to compile and run PhyInit for this protocols
                  |-obj/ # Contains compiled C objects of each PhyInit function
                  |-output/ # Contains text files output by PhyInit
                  |-src/ # Contains all non-editable source code
                  |-userCustom/ # Contains all user customizable source code
                  |-doc/ # Contains documentation for PhyInit source code
                  |-example/ # Example implementation of userCustom functions.
```

Running Phyinit



(A) Enter User Input

A. User edit the file `<protocol>/userCustom/dwc_ddrphy_phyinit_setDefault.c`. See Running Phyinit Section in HTML documentation for details. :

1. user_input_basic - these are inputs that the user must enter information specific to their configuration. The user must replace the default values in the field with their specific settings.
2. user_input_advanced - these are optional inputs. The user can choose to enter new values or use the default values.
3. user_input_sim - these are inputs that the user shall enter if execution of training firmware skipped. If training firmware is executed, the user can leave these fields at their default value.
4. Firmware Message Block these are inputs to the message block required to execute the training firmware. PhyInit populates some of the variables based on user_input_basic and user_input_advanced. See dwc_ddrphy_phyinit_calcMb() source code for details. These settings can still be change inside dwc_ddrphy_phyinit_setDefault() if desired.

B. Technology specific register programming.

– These registers are listed in section Technology Dependent Register Programming and must be programmed in dwc_ddrphy_phyinit_userCustom_customPreTrain().

(B) Specify Custom Behavior

- Two places for user to override Phyinit behavior:
  1. Before training - `dwc_ddrphy_phyinit_userCustom_customPreTrain()`
     - Override message block values calculated or set default by Phyinit
     - Override any CSR programming done by `dwc_ddrphy_phyinit_C_initPhyConfig()`
     - E.g. changing SequenceCtrl in message block
     - E.g. program CSRs for AC/Dbyte swizzle
     - E.g. change value of PclkPtrInitVal otherwise set by Phyinit

  2. After training - `dwc_ddrphy_phyinit_userCustom_customPostTrain()`
     - Override any CSR programmed by training firmware or `dwc_ddrphy_phyinit_progCsrSkipTrain()`
     - E.g. change values of delay CSRs to compensate for board delays

(C) Compile and Run

- To create all output.txt files just call make :

```
make FIRMWARE_PATH=<path to Training firmware release folder>
```

- `make` Options:

Option	Values	Description
RET_EN	[1,0]	Generates a retention exit sequence txt file and includes additional register reads during the initialization sequence to save PHY state. See “IO Retention Save/Restore” in PhyInit HTML documentation for details.

(D) Reading Phyinit Output

- Multiple output text files will be created, depending on protocol:

1. dwc_ddrphy_phyinit_out_<protocol>**_skiptrain**.txt -- this file contains the PHY initialization sequence for skipping training firmware. This is useful for shortening simulation time.
2. dwc_ddrphy_phyinit_out_<protocol>**_train1d**.txt – this file contains the PHY initialization sequence that runs 1D training firmware
3. dwc_ddrphy_phyinit_out_<protocol>**_train1d2d**.txt – applicable only for ddr4* and lpddr4, this file contains the PHY initialization sequence that runs 1D and 2D training firmware
4. dwc_ddrphy_phyinit_out_<protocol>_***_retention_exit**.txt - this file contains the IO restore retention exit sequence

- Each file contains sequence of APB writes, expressed in dwc_ddrphy_apb_wr(<address>, <data>);
- For steps where actions other than APB writes is necessary, such as step A, a function call is printed: dwc_ddrphy_phyinit_userCustom_A_bringupPower();
- Comments throughout the file to explain the steps and indicate which CSR is written

Example Output Text

dwc_ddrphy_phyinit_out_<protocol>_train1d.txt

```
//##############################################################
//
// (C) Initialize PHY Configuration
//
// Load the required PHY configuration registers for the appropriate mode and memory configuration
//
//##############################################################

// [phyinit_C_initPhyConfig] Start of dwc_ddrphy_phyinit_C_initPhyConfig()
dwc_ddrphy_apb_wr(32'h200c5,16'h19); // DWC_DDRPHYA_MASTER0_PllCtrl2_p0
// [phyinit_C_initPhyConfig] Pstate=0,  Memclk=1600MHz, Programming PllCtrl2 to 19 based on DfiClk frequency = 800.
dwc_ddrphy_apb_wr(32'h1200c5,16'ha); // DWC_DDRPHYA_MASTER0_PllCtrl2_p1
// [phyinit_C_initPhyConfig] Pstate=1,  Memclk=1067MHz, Programming PllCtrl2 to a based on DfiClk frequency = 533.
dwc_ddrphy_apb_wr(32'h2200c5,16'hb); // DWC_DDRPHYA_MASTER0_PllCtrl2_p2
// [phyinit_C_initPhyConfig] Pstate=2,  Memclk=933MHz, Programming PllCtrl2 to b based on DfiClk frequency = 466.
dwc_ddrphy_apb_wr(32'h3200c5,16'hb); // DWC_DDRPHYA_MASTER0_PllCtrl2_p3
// [phyinit_C_initPhyConfig] Pstate=3,  Memclk=800MHz, Programming PllCtrl2 to b based on DfiClk frequency = 400.
```

Example Output Text (Cont’d)

dwc_ddrphy_phyinit_out_<protocol>_train1d.txt

```
//########################################################################
//
// (E) Set the PHY input clocks to the desired frequency 800
//
// Set the PHY input clocks to the desired operating frequency. Before proceeding to the next step,
// the clock should be stable at the new frequency. For more information on clocking requirements, see
"Clocks" on page <XXX>.
//
//########################################################################

dwc_ddrphy_phyinit_userCustom_E_setDfiClk (800);
```

IO Retention Sequence

- PhyInit can be configured to :
  - include retention register save process in the PHY Initialization flow
  - generate a retention exit sequence txt file with exact registers to restore
- Requirements:
  - user needs to implement “dwc_ddrphy_phyinit_io_read16()” to read and save registers values
  - to enable add “RET_EN=1” to the PhyInit make command
    - e.g. make FIRMWARE_PATH=/synopsys/dwc_ddrphy/fw/rel **RET_EN=1**
- Output:
  - APB read commands are issued in the PHY init output text files
  - *_retention_exit.txt files are created which implement retention exit sequence for the given PHY configurations
  - user must modify *_retention_exit.txt to replace all 16’h0 values with saved values

Example Output Text with RetEn=1

dwc_ddrphy_phyinit_out_<protocol>_train1d.txt

```
// //##############################################################
// //
// // Customer Save Retention Registers
// //
// // This function can be used to implement saving of PHY registers to be
// // restored on retention exit. the following list of register reads can
// // be used to compile the exact list of registers.
// //
// //##############################################################

dwc_ddrphy_apb_wr(32'hd0000,16'h0); // DWC_DDRPHYA_APBONLY0_MicroContMuxSel
dwc_ddrphy_apb_wr(32'hc0080,16'h3); // DWC_DDRPHYA_DRTUB0_UcclkHclkEnables
// dwc_ddrphy_apb_rd(32'h1005f);
// dwc_ddrphy_apb_rd(32'h1015f);
// dwc_ddrphy_apb_rd(32'h1105f);
// dwc_ddrphy_apb_rd(32'h1115f);
// dwc_ddrphy_apb_rd(32'h1205f);
// dwc_ddrphy_apb_rd(32'h1215f);
// dwc_ddrphy_apb_rd(32'h1305f);
// dwc_ddrphy_apb_rd(32'h1315f);
// dwc_ddrphy_apb_rd(32'h1405f);
```

Example Retention Exit Text File

dwc_ddrphy_phyinit_out_<protocol>_train1d_retention_exit.txt

```
// [dwc_ddrphy_phyinit_restore_sequence] Write the MicroContMuxSel CSR to 0x0 to allow access to the internal CSRs
dwc_ddrphy_apb_wr(32'hd0000,16'h0); // DWC_DDRPHYA_APBONLY0_MicroContMuxSel
// [dwc_ddrphy_phyinit_restore_sequence] Write the UcclkHclkEnables CSR to 0x3 to enable all the clocks so the reads can complete
dwc_ddrphy_apb_wr(32'hc0080,16'h3); // DWC_DDRPHYA_DRTUB0_UcclkHclkEnables
//  Issue register writes to restore registers values.
//  --------------------------------------------------
//  - IMPORTANT -
//  - The register values printed in this txt file are always 0x0. The
//    user must replace them with actual registers values from the save sequence. The
//    save sequence can be found in the output.txt file associated with the particular init sequence.
//    The save sequence issues APB read commands to read and save register values.
//  - refer to the init sequence output file for details.
//  ------------------------------------------------------------------------------

dwc_ddrphy_apb_wr(32'h1005f,16'h0); // DWC_DDRPHYA_DBYTE0_TxSlewRate_b0_p0
dwc_ddrphy_apb_wr(32'h1015f,16'h0); // DWC_DDRPHYA_DBYTE0_TxSlewRate_b1_p0
dwc_ddrphy_apb_wr(32'h1105f,16'h0); // DWC_DDRPHYA_DBYTE1_TxSlewRate_b0_p0
dwc_ddrphy_apb_wr(32'h1115f,16'h0); // DWC_DDRPHYA_DBYTE1_TxSlewRate_b1_p0
dwc_ddrphy_apb_wr(32'h1205f,16'h0); // DWC_DDRPHYA_DBYTE2_TxSlewRate_b0_p0
dwc_ddrphy_apb_wr(32'h1215f,16'h0); // DWC_DDRPHYA_DBYTE2_TxSlewRate_b1_p0
dwc_ddrphy_apb_wr(32'h1305f,16'h0); // DWC_DDRPHYA_DBYTE3_TxSlewRate_b0_p0
dwc_ddrphy_apb_wr(32'h1315f,16'h0); // DWC_DDRPHYA_DBYTE3_TxSlewRate_b1_p0
```

Registers not included in PhyInit

- For any register not programmed, the customer is required to include it in either of these functions:
  - dwc_ddrphy_phyinit_userCustom_customPostTrain.c
  - dwc_ddrphy_phyinit_userCustom_customPreTrain.c
- Technology Dependent Register Programming
- Very simple Register programming where PhyInit cannot simply user experience.
  - Example : Lane Swizzle Dq0LnSel;
- Debug or Performance counter registers
  - Example AnalogDebug
- STAR workarounds
  - Some STAR workarounds may instruct to program or override a register setting.
- SOTW or otherwise specified non-default register settings.

Users may need to override PhyInit Logic

Examples where override is necessary

- Lets take PhyInterruptEnable register.
  - PhyEccEn is automatically set based on basic user inputs.
  - The rest of the fields cannot be directly concluded.
  - Only few customers may want to enable Interrupts.
  - PhyInit leaves interrupts fields disabled.
  - Customers that want an interrupt enabled, must overwright PhyInterruptEnable.
  - either CustomPre() or customPost() functions

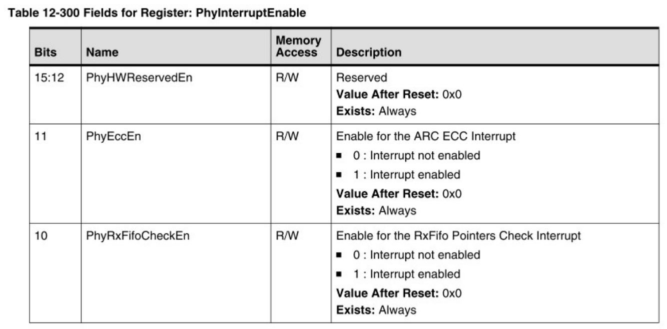


Table 12-300 Fields for Register: PhyInterruptEnable

Bits	Name	Memory Access	Description
15:12	PhyHWReservedEn	R/W	Reserved **Value After Reset:** 0x0 **Exists:** Always
11	PhyEccEn	R/W	Enable for the ARC ECC Interrupt ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always
10	PhyRxFifoCheckEn	R/W	Enable for the RxFifo Pointers Check Interrupt ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always

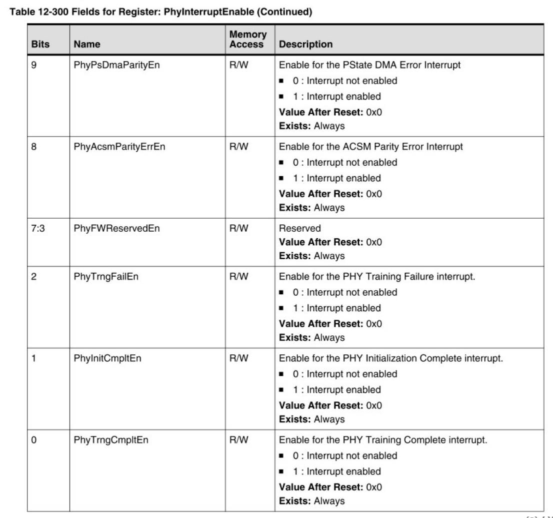


Table 12-300 Fields for Register: PhyInterruptEnable (Continued)

Bits	Name	Memory Access	Description
9	PhyPsDmaParityEn	R/W	Enable for the PState DMA Error Interrupt ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always
8	PhyAcsmParityErrEn	R/W	Enable for the ACSM Parity Error Interrupt ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always
7:3	PhyFWReservedEn	R/W	Reserved **Value After Reset:** 0x0 **Exists:** Always
2	PhyTrngFailEn	R/W	Enable for the PHY Training Failure interrupt. ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always
1	PhyInitCmpltEn	R/W	Enable for the PHY Initialization Complete interrupt. ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always
0	PhyTrngCmpltEn	R/W	Enable for the PHY Training Complete interrupt. ■ 0 : Interrupt not enabled ■ 1 : Interrupt enabled **Value After Reset:** 0x0 **Exists:** Always

Customer testbench for LPDDR54

Synopsys Confidential Information

How to use CTB to run RTL simulation

- Environment setup
  - Download and unpack the Synopsys DFI and memory VIP in <VIP unpack directory>
  - Install the Synopsys DFI and memory VIP in <VIP install directory> (refer to Synopsys_VC_VIP_Memory_Portfolio_README.txt)
  - Set environment variable for VIP

    % *setenv* DESIGNWARE_HOME <VIP unpack directory>

    % *setenv* CTB_VIP_HOME <VIP install directory>
  - Source the boot environment to set up directory paths

    % cd synopsys/<ip_name>/Latest/ctb/Latest/sim

    % source bootenv

- Run test

  % runtc <command_options>

  (% runtc cfg=lp54cs1dq18ch1 tc=demo_basic dram=lpddr4 skip_train=1 freq0=333 freq_ratio0=1 rank=1 pstates=1 dfi_mode=1 hard_macro=B)

Test Cases

Testcase	Descriptions	Flow
demo_basic	test scenario for basic mission mode write/read transactions	(1) Initialize PHY. (2) Send mission mode write followed by read transaction. (3) Check write/read data.
demo_lp	test scenario for low power states ✓ DFI lower power ✓ frequency change ✓ LP3/IO Retention	(1) Initialize PHY. (2) Send mission mode write followed by read transaction. (3) Check write/read data. (4) Enter and exit PHY low power state(“lp_mode=< 1:dfi_lp_data/2:dfi_lp_ctl/3:dfi_lp_ctl + DRAMCLKSTOP/4:frequency change with LP2/5:LP3/6:IO retention >”). (5) Send mission mode write followed by read transaction. (6) Check write/read data.
demo_ate		

SYNOPSYS®
Silicon to Software™

Thank You