KBP SDK 1.5.13

Release Notes

Revision History

Revision	Date	Change Description
KBP-SDK-1.5.13	07/12/20	Bug fixes and enhancements for OP2
KBP-SDK-1.5.12	03/30/20	Bug fixes and enhancements for OP2
KBP-SDK-1.5.11	11/26/19	Bug fixes and scale enhancements for OP2 and Jericho2 internal forwarding solution. Introduction of thread safety feature.
KBP-SDK-1.5.10	07/29/19	Bug fixes and enhancements for OP,OP2, Jericho and Jericho2 internal forwarding solution
KBP-SDK-1.5.9.1	05/07/19	Bug fix for OP2.
KBP-SDK-1.5.9	04/26/19	Bug fix(es) and enhancement(s) for OP and OP2.
KBP-SDK-1.5.8.2	04/17/19	Bug fix(es) and enhancement(s) for OP and OP2.
KBP-SDK-1.5.8.1	03/18/19	Bug fix(es) and enhancement(s) for OP2.
KBP-SDK-1.5.8	02/28/19	Bug fix and enhancements for OP2.
KBP-SDK-1.5.7.1	02/06/19	Bug fix for OP2.
KBP-SDK-1.5.7	01/21/19	Bug fixes and enhancements for OP2.
KBP-SDK-1.5.6.1	12/14/18	Bug fixes for OP2.
KBP-SDK-1.5.6	12/05/18	Bug fixes and enhancements for OP2 and Jericho2 internal forwarding solution.
KBP-SDK-1.5.5.1	10/12/18	Bug fix for OP2

KBP-SDK-1.5.5	09/24/18	Bug fixes for OP2 and Jericho2 internal forwarding solution. Support for 156.25 MHz Ref CLK
KBP-SDK-1.5.4.1	08/31/18	Bug fixes for OP2 and Jericho2 internal forwarding solution.
KBP-SDK-1.5.4	08/06/18	Bug fixes and enhancements for OP2 and Jericho2 internal forwarding solution, introducing statistics support.
KBP-SDK-1.5.3.2	07/19/18	Bug fixes for OP2.
KBP-SDK-1.5.3.1	06/25/18	Bug fixes for OP2.
KBP-SDK-1.5.3	06/01/18	Bug fixes and enhancements for OP2 and Jericho2 internal forwarding solution, introducing counter support.
KBP-SDK-1.5.2	03/29/18	Bug fixes for OP, OP2 and Jericho2 internal forwarding solution, LUT programming API for OP and OP2
KBP-SDK-1.5.1	02/23/18	Bug fixes for OP2 and Jericho2 internal forwarding solution
KBP-SDK-1.5.0	10/25/17	Preliminary support for Optimus Prime 2 and Jericho2 internal forwarding solution
KBP-SDK-1.4.17	02/02/18	AD manager optimization for v4/v6 mix, kbp_db_stats API enhancement, BER API enhancements, bug fixes.
KBP-SDK-1.4.16	11/03/17	API to reset the device, bug fixes.
KBP-SDK-1.4.15	10/06/17	Option to return error instead of abort in assert function for transport error, MDIO read for last MDIO write, and few bug fixes

Release Notes for KBP SDK Rev. 1.5.13

Description:

Compatibility with KBP Driver

- This version of the SDK is compatible with KBP driver version 2.12

KBP SDK 1.5.13 is a general availability release that includes fixes and enhancements for Optimus Prime 2. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.22 and KBP SDK 1.5.12.

KBP SDK 1.5.13:

1) [CS9303101] Enhancement to improve consistency of counter values read at a fixed periodic interval.

   The following API needs to be used to enable this enhancement.

   kbp_tap_db_set_property(tap_db, KBP_TAP_DB_PROP_RATE_COMPUTE, 1)

2) [CS9277297] Bug fix for issue in Stats DBs reporting more than expected value.

   The following API needs to be used to enable this fix.

   kbp_device_set_property(device, KBP_DEVICE_PROP_TAP_RESERVE_UDM, 1)

3) [KBPSDK-3019] [KBPSDK-3020] Bug fix related to result port assignment of counter enabled DBs.

4) Bug fix for issue in LUT programming for single port mode in OP.

5) General availability support for the 40M SKU of OP2

6) [SDK–209458] Enhancements in 0b AD feature to support return values in the range of 0-63 to ensure zeros on MSB 16 bits [23:8].

7) Bug fix for crash in a scenario where core0 in OP2 has no DBs.

8) Bug fix for crash in a scenario with variable size AD.

9) Bug fix for an issue where ACL DBs in OP produce occasional lookup failures.

10) Limited support for hitbits feature on OP2.

11) KBP transport layer (XPT) source code is available as a separate package upon request.

12) The following API has been introduced to enable programming and reset of Tx and Rx interfaces on SerDes.

kbp_status kbp_device_interface_program_PHMF_FIFO(enum kbp_device_type type, struct kbp_device_config *config, uint64_t lane_bitmap, uint8_t reset);

kbp_status kbp_device_interface_PHMF_FIFO_toggle(enum kbp_device_type type, struct kbp_device_config *config, uint32_t flags);

kbp_status kbp_device_interface_TXPCS_soft_reset_toggle(enum kbp_device_type type, struct kbp_device_config *config);

kbp_status kbp_device_interface_tx_rx_toggle(struct kbp_device_config *config, uint32_t flags);

KBP SDK 1.5.12:

1) General availability support KBP_DEVICE_THREAD_SAFETY feature.

2) Resource management behavior has been changed to not disable DB access from a different core when a constraint related to instructions is hit. Instead an error code will now be returned indicating the scenario could not fit into the device.

3) The following APIs have been introduced to enable configuration on resources on KBP using a predetermined customized profile.

kbp_status kbp_device_load_resource_profile(struct kbp_device * device, char *resource_str);

kbp_status kbp_device_get_resource_profile(struct kbp_device *device, char *resource_str, uint32_t resource_str_size);

4) [CS9341543] [CS9287140] Bug fixes in ILKN SerDes diagnostics APIs.

5) New APIs related to Statistics interface diagnostics have been added.

kbp_status kbp_device_pm_interface_serdes_prbs(void *xpt_hdl, enum kbp_prbs_polynomial prbs_poly, uint32_t enable, uint16_t lane_bitmap);

kbp_status kbp_device_pm_interface_serdes_prbs_print(void *xpt_hdl, uint16_t lane_bitmap);

kbp_status kbp_device_pm_interface_serdes_eyescan(void *xpt, uint16_t lane_bmp);

kbp_status kbp_device_pm_interface_serdes_lanestate(void *xpt, uint16_t lane_bmp);

6) [CS8256160] kbp_pcie_init() API has been enhanced to detect and correct corrupt PCIe command register.

7) Limited support for the 40M SKU of OP2.

KBP SDK 1.5.11:

1) kbp_device_interface_init API has been enhanced to improve link stability, it may take a longer time to execute compared to previous version.

2) Error handling demo has been updated to showcase more types of errors and appropriate handling of the same.

3) Scale improvements in Jericho2 internal forwarding engine.

4) Bug fix for coherency issue in Jericho2 internal forwarding engine.

5) [CS8759607] A new device property KBP_DEVICE_THREAD_SAFETY has been added. This flag can be used in kbp_pcie_init API to enable a locking mechanism in the PCIe transport layer to allow multiple threads accessing the transport layer. A separate thread can now be used for updates to search Databases, for Error handling and for reading Stats Databases. This release contains preliminary support only for this feature.

6) A new database property KBP_PROP_MC_DB has been added. A new API **kbp_key_set_critical_field** can be used in conjunction with this property to optimize the database for certain data patterns typically seen in Multicast Databases.

7) Bug fixes and enhancements in op and tap diagnostic tools.

8) [CS9015531] Bug fix for issue in kbp_instruction_set_property API where per port result size constraint of 128b for OP device types wasn't being enforced.

9) Fix for issue with warmboot for stats databases.

10) Support added for 0b AD size in LPM databases for OP2 device type.

11) New device property KBP_DEVICE_PROP_HANDLE_INTERFACE_ERRORS added. This property can be used to disable handling of interface errors by kbp_device_fix_errors API.

12) kbp_status_errors structure has been enhanced to include additional error bits.

KBP SDK 1.5.10:

1) ILA mode support for Optimus Prime 2.
2) Blackhawk FW version upgraded to D100_07.
3) Fix in kbp_db_stats API for Jericho internal forwarding solution.
4) [CS8109495][CS8448612] Fix for issue where DB Soft Errors are observed on OP2 Core-1 after device initialization.
5) Fix for coherency issue in Jericho2 internal forwarding solution.
6) Capacity improvements for Databases using per entry AD feature.
7) Fix for bug related to kbp_pcie_init API usage during warmboot for OP/OP2.
8) Fix for issue found when kbp_device_warmboot_save_and_continue API is called subsequent to a kbp_pcie_destroy API call.
9) Blackhawk FW version upgraded to D100_06.
10) [CS7925669] [KBPSDK-2632] - O3S bug fix in IFSR with BROADCAST_AT_XPT property set.
11) [CS8080941] [KBPSDK-2614] – Fix for capacity issue in Jericho+ running in Jericho compatibility mode.
12) Default TXFIR settings for ILKN interfaces running in NRZ mode updated with optimized values for typical designs. Customers are expected to analyze individual systems and determine TXFIR settings that work best for the same.
13) Bug fix for an issue in LPM where databases using result resolution were occasionally returning incorrect hits.
14) Bug fix in “op show clk” command to display core clock frequency.
15) Fix for bug causing slowdown in entry addition after kbp_instruction_search API is invoked.
16) kbp_pcie_init API has enhanced to detect link issues and return appropriate return codes.

17) kbp_device_selective_shutdown API has been added to selectively shutdown resources on KBP.

18) The following APIs have been introduced to support reading and writing device registers using PIO transactions. These APIs have limited support across different devices. Please contact Broadcom Support for guidance on how to use them.

kbp_status kbp_device_pio_register_write(struct kbp_device * device, uint32_t address, uint64_t data);
kbp_status kbp_device_pio_register_read(struct kbp_device * device, uint32_t address, uint64_t *o_data);

19) A new API has been introduced to allow users to add overlay keyfields on top of the master key.
kbp_status kbp_key_overlay_field(struct kbp_key *master_key, char *name, uint32_t width_1, enum kbp_key_field_type type, uint32_t offset_1) ;

20) Fix for PRBS issue.

21) New command added “op show lut_tbl” to dump LUT info.

22) *struct kbp_status_errors* has been updated with new fields to allow additional interrupt triggers, please check the structure definition for additional details.

23) kbp_device_fix_errors API has been enhanced to handle Interface related errors as well.

24) The following API has been enhanced to take an additional argument to give control to users on what error bits to be cleared. The new argument is sel_clr.
kbp_status kbp_device_clear_errors(struct kbp_device *device, struct kbp_status_errors *sel_clr)

KBP SDK 1.5.9.1:

1) Bug fix for AVS related issue on OP2 A0 silicon.

KBP SDK 1.5.9:

1) [KBPSDK-2502] struct kbp_entry_info has a new member ad_db, this can be used to determine the ad_db associated with the current entry.

2) New DB property KBP_PROP_XL_DB added. This property can be used to indicate that a DB is exceptionally large and optimize resource allocation and management for the same.

3) New DB property KBP_PROP_REPLICATE_DB added. This property can be used to indicate if a DB needs to replicate when being accessed from multiple device threads.

4) Scale improvements for LPM DBs.

5) Blackhawk Firmware version upgraded to D100_05.

6) kbp_device_dump() API has been enhanced to dump TAP information.

KBP SDK 1.5.8.2:

1) Bug fix for portability issue with bit field.

2) New DB property KBP_PROP_LOCALITY added. This property can be used to group DBs together in KBP and optimize storage.

3) Bug fix for LUT issue related to DUMMY fields.

4) Supports OP core clock programming for 750MHz when CPSEL pins are set to 720MHz operation.

5) kbp_pcie_set_property API updated to support user programmable signal number for MSI.

6) kbp_pcie_set_property API updated with MSI disable feature

7) Bug fix in kbp_device_read_die_temperature API & kbp_device_read_die_voltage API

8) ***struct kbp_status_errors*** has been updated with a new field ***crb_error_interrupt.*** Error interrupts trigger PCIe MSI when this field is set through kbp_device_enable_interrupt API

9) Fix for spurious error messages generated as part of AVS initialization.

10) Fix for AVS bug in op present in the following KBP SDK versions: 1.5.4.1, 1.5.5.1 and 1.5.6.1

KBP SDK 1.5.8.1:

1) [KBPSDK-2532] LUT issue related to padding.

2) KBP device driver version updated to 2.9 to support Linux kernel version >= 4.11

KBP SDK 1.5.8:

1) [KBPSDK-2520] AD entries exhausted after different amount of entries added between 1.5.6 and 1.5.7.

2) LPM scale improvements.

3) AVS support for OP2.

KBP SDK 1.5.8 also introduces the following APIs.

1) ***kbp_status kbp_device_inject_errors(struct kbp_device *device, struct kbp_status_errors *err, uint32_t emulate);***

   This API injects/emulates an error in the device, non-emulation mode supported only for DBA/UDA/UIT memory

2) ***kbp_status kbp_device_get_errors(struct kbp_device *device, struct kbp_status_errors *o_errors);***

   This API can be used to get the current error status in the device.

3) ***kbp_status kbp_device_enable_interrupt(struct kbp_device *device, struct kbp_status_errors *en_err);***

   This API can be used to set the masks for the errors that will trigger interrupt. Each call will overwrite the previous values.

4) ***kbp_status kbp_device_clear_errors(struct kbp_device *device);***

   This API can be used to clear the errors in the device.

5) ***kbp_status kbp_device_get_error_list(struct kbp_device *device, uint64_t *error_buffer, uint32_t buffer_size, uint32_t *filled_count);***

   This API can be used to get the list of error codes for the errors addressed during the last kbp_device_fix_errors() call.

6) ***kbp_status kbp_device_interface_serdes_tap_settings(enum kbp_device_type type, struct kbp_device_config *config, uint32_t lane_bmp, struct kbp_serdes_tap_settings *tap_settings);***

   This API can be used to set TXFIR settings for ILKN/Search interface.

7) ***kbp_status kbp_device_pm_interface_serdes_tap_settings(void *xpt, uint16_t lane_bmp, struct kbp_serdes_tap_settings *tap_settings);***

   This API can be used to set TXFIR settings for PM/Stats interface.

KBP SDK 1.5.7.1:

KBP SDK 1.5.7.1 is a limited availability release that includes additional fixes for Optimus Prime 2. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.19 and KBP SDK 1.5.7.

KBP SDK 1.5.7.1 has the following fixes and enhancements:

1) Fix for spurious error messages from kbp_device_interface_init API.

KBP SDK 1.5.7:

KBP SDK 1.5.7 is a limited availability release that includes additional fixes and enhancements for Optimus Prime 2. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.19 and KBP SDK 1.5.6.1.

KBP SDK 1.5.7 has the following fixes and enhancements:

1) Optimus Prime 2 B0 support.
2) AVS support for Optimus Prime 2.
3) [KBPSDK-2497] ACL counters combined with Stats are not working.
4) [KBPSDK-2506] OP2 fails to reestablish the stat ports.
5) The following API has been enhanced to take custom TXFIR settings as input

   kbp_status kbp_device_pm_interface_init(int unit, void *xpt_hdl, int32_t port_speed, int32_t port_bmp, int32_t is_nrz, int32_t link_training, struct kbp_serdes_tap_settings *user_emph_vals)

KBP SDK 1.5.6.1:

KBP SDK 1.5.6.1 is a limited availability release that includes additional fixes for Optimus Prime 2. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.19 and KBP SDK 1.5.6.

KBP SDK 1.5.6.1 has the following fixes:

6) Bug fix issue with Counters.

7) Bug fix for issue in Warmboot for Statistics DBs.

KBP SDK 1.5.6:

KBP SDK 1.5.6 is a limited availability release that includes additional fixes for Optimus Prime 2 and Jericho2 internal forwarding solution. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.19 and KBP SDK 1.5.5.1

KBP SDK 1.5.6 has the following fixes and enhancements:

1) Bug fix for issue in NetRoute multi-access databases on Optimus Prime 2.

2) Bug fix for issue with Dummy DBs in SMT mode.

3) Improved scale for v6 databases in Jericho2 internal forwarding solution.

KBP SDK 1.5.6 also introduces API to inject parity errors into the KBP.

1) ***kbp_status kbp_device_inject_errors(struct kbp_device *device);***

   This API can be used to inject a soft error into the KBP. This API facilitates testing the error handling capabilities of the KBP and is expected to be used in conjunction with kbp_device_fix_errors API.

KBP SDK 1.5.6 also introduces two APIs to initialize and configure field value for TAP databases.

1) ***kbp_status kbp_tap_db_value_init(struct kbp_tap_db *db, struct kbp_tap_db_value **value);***

2) ***kbp_status kbp_tap_db_value_set_field(struct kbp_tap_db_value *value, char *name, uint32_t width_1);***

KBP SDK 1.5.5.1:

KBP SDK 1.5.5.1 is a limited availability release that includes additional fixes for Optimus Prime 2 and Jericho2 internal forwarding solution. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.19 and KBP SDK 1.5.5

KBP SDK 1.5.5.1 has the following fix:

1) Bug fix for Optimus Prime 2 clock programming in Big Endian platforms.

KBP SDK 1.5.5:

KBP SDK 1.5.5 is a limited availability release that includes additional fixes for Optimus Prime 2 and Jericho2 internal forwarding solution. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.19 and KBP SDK 1.5.4.1

KBP SDK 1.5.5 introduces support for 156.25 MHz SerDes reference clock for Optimus Prime 2. It also includes Warmboot support for Jericho2 internal forwarding solution.

KBP SDK 1.5.5 has the following fixes:

1) CS6104616: Fix for issue with KBP init sequence getting stuck for OP2.

KBP SDK 1.5.4.1:

KBP SDK 1.5.4 is a preview release that includes additional fixes for Optimus Prime 2 and Jericho2 internal forwarding solution. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.18 and KBP SDK 1.5.4

KBP SDK 1.5.4.1 has the following fix:

1) Fix for issue in kbp_instruction_set_opcode in SMT mode.

2) Fix for assert in JR2 KAPS.

KBP SDK 1.5.4:

KBP SDK 1.5.4 is a preview release that includes additional fixes and enhancements for Optimus Prime 2 and Jericho2 internal forwarding solution. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.18 and KBP SDK 1.5.3.2. Previous versions of KBP silicon (NLA12K) are fully supported by this SDK.

KBP SDK 1.5.4 also introduces support for Statistics. A reference application has been added in the SDK package to showcase the usage of APIs related to Statistics.

KBP SDK 1.5.3.2:

KBP SDK 1.5.3.2 is a preview release that includes additional fixes for Optimus Prime 2. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.18 and KBP SDK 1.5.3.1. Previous versions of KBP silicon (NLA12K) are fully supported by this SDK.

KBP SDK 1.5.3.2 has the following fix:

1) Fix for issue in kbp_instruction_set_opcode API.

KBP SDK 1.5.3.1:

KBP SDK 1.5.3.1 has the following fix:

1) [KBPSDK-2445] kbp_device_interface_init fails for Optimus Prime 2.

KBP SDK 1.5.3:

KBP SDK 1.5.3 is a preview release that includes additional fixes and enhancements for Optimus Prime 2 and Jericho2 internal forwarding solution. This release also supports Optimus Prime and contains all the issues resolved as of KBP SDK 1.4.18 and KBP SDK 1.5.2. Previous versions of KBP silicon (NLA12K) are fully supported by this SDK.

KBP SDK 1.5.3 also introduces support for Counters. A reference application has been added in the SDK package to showcase the usage of APIs related to Counters.

KBP SDK 1.5.2:

KBP SDK 1.5.2 also discontinues support of compare3 instructions for Optimus Prime and Optimus Prime 2. Compare3 is still supported for NLA12K devices.

KBP SDK 1.5.2 also introduces APIs to configure LUT through KBP SDK for Optimus Prime and Optimus Prime 2.

1. ***kbp_status kbp_instruction_set_opcode(struct kbp_instruction *instruction, int32_t opcode);***

   This API allows user to set the LUT opcode for the instruction. It should be called after instruction install and the same opcode will be used to do LUT write from the KBP SDK.

   “instruction” : Instruction handle

   “opcode” : LUT Opcode to be programmed for instruction

Internally discovered bugs through testing on Optimus Prime silicon have been resolved. Please refer to the Optimus Prime Silicon Errata sheet for additional details on the silicon status.

This release also supports crash recovery for NLA12K and Jericho internal forwarding solution. All features of NLA12K and Jericho internal forwarding solution are GA quality and fully supported.

This version of the SDK should not be used for capacity or update rate evaluations for Optimus Prime 2 or Jericho2 internal forwarding solution. The release is mainly a preview for feature in Optimus Prime 2, and to demonstrate portability of existing applications already using KBP SDK APIs.

This SDK includes –

- APIs to initialize the device, create instructions and keys, and create and manage databases.
- Multiple Reference applications that showcase use of Optimus Prime 2, Optimus Prime and NLA12K.
- A C-Model for NLA12K, Optimus Prime and Optimus Prime 2. The model is transaction, not cycle accurate.
- The SDK can be used in Blackhole mode without using the C-Model or real device. Pass the transport layer handle as null in kbp_device_init().
- Dynamic resource management of databases for NLA12K, Optimus Prime and Optimus Prime 2
- Support for managing multiple KBP's on one line card
- Warmboot and crash recovery

Optimus Prime 2:

- Any existing application coded for Optimus Prime is expected to work with Optimus Prime 2 C-Model
- All aspects of configuring and managing the device
- NetACL, including associated data.
- NetRoute with associated data
- Support for Massively parallel ACLs, Power controlled ACLs with associated data.
- Single cycle lookup of multi access LPM databases.
- Resolution between lookup results of databases. Typical use is for resolving the result of a match between a public and private forwarding database.
- Hit bits and Aging
- Counters and Statistics

Optimus Prime:

- Any existing application coded for NLA12K is expected to work with Optimus Prime C-Model
- All aspects of configuring and managing the device
- NetACL, including associated data.
- NetRoute with associated data
- Support for Massively parallel ACLs, Power controlled ACLs with associated data.
- Single cycle lookup of multi access LPM databases.
- Resolution between lookup results of databases. Typical use is for resolving the result of a match between a public and private forwarding database.
- Hit bits and Aging
- Warmboot for all database types

NLA12K:

- GA Support for NetRoute, Massively parallel ACLs, Power-controlled ACLs, and Associated data for ACLs, SMT and Cascade.
- Warmboot and crash recovery support for all database types
- Parity error correction (In field soft repair) for entries in DBA and correction of algorithmic errors reported by device

Devices Supported:

- NL12000, NLA12000, NLA88650, ERC120, SD5452A. SDK has been tested with Rev A2 and Rev B0 and B1 Silicon
- BCM52311, BCM15K. SDK has been verified against A0 and A1 silicon
- BCM16K. SDK has been verified against A0 and B0 silicon
- Jericho 2 BCM88690, Jericho 2C BCM88800, Qumran 2C BCM88850

Related documents: (available at https://support.broadcom.com) :

- Device Datasheets: NLA12000-DS116-R-KBP, NLA88650-DS107-R-KBP, 41CSERC120-DS115-R-KBP , 48CSSD5452A-DS115-R-KBP, 1CSCSCO-DS116-RDS-KBP 15000-DS126, 52311-DS128, 16000-DS118
- KBP SDK Reference Manual, KBP-SDK-RM100 or newer

Known Limitations / Issues:

The following features of the Optimus Prime 2 chip are currently not supported

- Crash Recovery
- Capacity and update rates are not optimal for NetACL databases
- The KBP SDK cannot recover if a fatal error occurs when transporting messages to the KBP

- Link training for 56G SerDes is unstable, manual TXFIR settings might be required.

The following features of the Optimus Prime chip are currently not supported

- Capacity and update rates are not optimal for NetACL databases
- The KBP SDK cannot recover if a fatal error occurs when transporting messages to the KBP

Optimus Prime C-Model Limitations:

The C-Model is a transaction accurate, not cycle accurate model. Model does not support-

- Network / interface specific functionality such as ILA protocol, Network Byte Order, packet errors.
- ECC, Parity scan / protection, Parity Errors.
- MDIO, JTAG Boundary Tag Registers
- Response to Error Detection, Test Pattern Generator / Checker
- Multiple instances of C-Model may not be instantiated at the same time

Optimus Prime 2 C-Model Limitations:

The C-Model is a transaction accurate, not cycle accurate model. Model does not support-

- Network / interface specific functionality such as ILA protocol, Network Byte Order, packet errors.
- ECC, Parity scan / protection, Parity Errors.
- MDIO, JTAG Boundary Tag Registers
- Response to Error Detection, Test Pattern Generator / Checker
- Multiple instances of C-Model may not be instantiated at the same time

NLA12K C-Model Limitations:

The C-Model is a transaction accurate, not cycle accurate model. Model does not support-

- Network / interface specific functionality such as ILA protocol, Network Byte Order, packet errors.
- ECC, Parity scan / protection, Parity Errors.
- MDIO, Cascade of processors, JTAG Boundary Tag Registers
- Upper limit on total Associated Data in responses
- Low Power Mode (see Dynamic Power Control section in Datasheet)

Response to Error Detection, Test Pattern Generator / Checker