BROADCOM

Product Brief

SDKLT: Logical Table-Based Switch Development Kit

Easy-to-Use, High Performance, Flexible, and Open

Key Features

- Data-driven modular architecture
- Table-based programming
- Logical tables APIs
- Consistent and simple set of APIs
- Reliable Warmboot and ISSU
- Asynchronous API operations
- Batched or Atomic transactions
- Transaction rollback
- API execution replay for debug

Key Benefits

- Ease-of-use
- Rapid development time
- Operational flexibility
- Faster configuration and boot time
- Ability to batch entry operations
- Monitoring of resource usage
- Packet I/O performance
- Easy to bug with relevant debug information

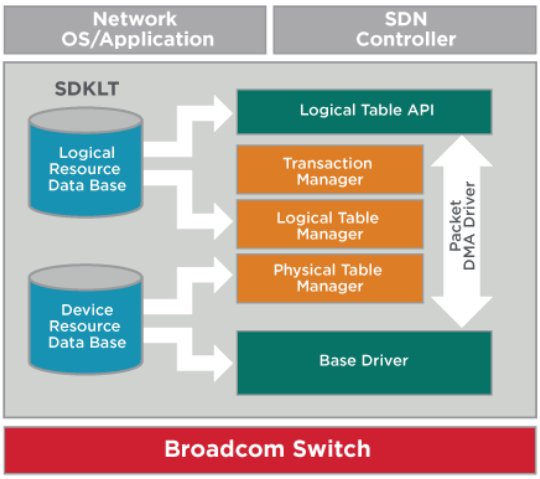
Description

Switch Development Kit Logical Tables (SDKLT) is purpose-built to support a data-driven programming model for next generation data centers and service providers.

The SDKLT supports table-based programming using Logical Table APIs, which support the new generation of switch programming models.

- Flexibility of the user to implement the device functions.
- Visibility into physical device resource usage.
- Ability to update multiple physical tables in a single transaction.

SDKLT Architectural Diagram



SDKLT Architecture

The SDKLT architecture shown in the figure above consists of several functional blocks that handle the operations associated with the Logical Table APIs as they are executed.

- Logical Table APIs (LT-APIs): Simple APIs enable access to logical tables.
- Transaction Manager (TRM): Parses logical table APIs and facilitates them as specified to the Logical Table Manager (LTM) or cache. It helps support sync, async APIs and batched or atomic transactions.
- Logical Table Manager (LTM): Manages logical table entry operations, and communicates with the Physical Table Manager.
- Physical Table Manager (PTM): Receives requests from the LTM, converts them into device-specific register/memory format.
- Base driver: Contains the driver for the internal I/O to program the chip.
- Packet DMA driver: Supports CMIC packet I/O.

SDKLT Architecture (continued)

- Logical Resource Data Base (LRDB): Contains all the data necessary for logical tables and their mappings.
- Device Resource Data Base (DRDB): Contains information about device memories, registers, and embedded processor elements.

End-User Experience

- Simplified and reduced set of consistent APIs.
- APIs enable structured programming and code reuse.
- Common terminology between the hardware and software.
- Improved debug with diag-shell access to all logical and physical tables, event logging, playback, and history
- Common error handlers with relevant information.

Usage Flexibility

- LT-APIs provide flexible access to any logical table.
- LT-APIs remain the same regardless of the tables or device.

The resource manager allows the user to query per-table capacity.

Transactions enable multiple table-entry operations to be combined or batched into a single transaction. Transactions can be atomic (all or none) or simple batches of table operations. Table operation ordering is preserved within a transaction.

Applications can register to receive asynchronous notification for any table update. Reliable In-Service Software Upgrade (ISSU) recovery is inherent in the database-driven programming model. The recovery of data structures after Warmboot is a very reliable, robust, warm-boot operation.

Performance Optimization

Improved table update performance is enabled by software-modeled access to most physical tables and pipelined posted writes to the hardware. Lockless and lightweight SDK packet processing result in high packet I/O performance. Both asynchronous and synchronous API transactions can be used resulting in operational performance.

Data-Driven Modular Architecture

Support for devices and features is provided through a modular infrastructure. The devices and features reside in data files contained within the LRD and DRD. Features are implemented in a modular fashion across devices.

APIs are simple, uniform table management primitives across all devices. Table-based APIs are offered to manage devices and features. This results in faster configuration and boot time.

Summary

The logical table-based SDKLT is a new approach in configuring switch devices that are architected around a data-driven programming model. It provides a complete solution for a scalable and modular development environment that benefits ease-of-use, programming flexibility, and performance optimization.

SDKLT-PB100 01.29.18