GOST 28147-89 Specification

Introduction

The `gost_28147_89` module implements the GOST 28147-89 encryption algorithm, a Soviet and Russian government standard symmetric key block cipher. This module is designed to perform both encryption and decryption operations based on the specified mode. It supports the use of different S-boxes, which can be selected based on the configuration. The design adheres to the GOST R 34.11-94 standard for cryptographic hash functions, providing flexibility in cryptographic applications.

Architecture

The architecture of the `gost_28147_89` module is structured around a synchronous design with a single clock domain. The module consists of the following key components: - **Cipher Cycle Counter**: A 5-bit register `i` that counts the number of cipher cycles, ranging from 0 to 31. - **Key Storage**: An array `K` of eight 32-bit registers used to store the cipher key. - **Data Registers**: Two 32-bit registers `a` and `b` hold the most significant and least significant halves of the input data. - **S-box Transformation**: The module includes an S-box transformation function, which is conditionally selected based on the `select` input. - **Key Index Calculation**: The module computes the key index for encryption or decryption based on the current cycle count and mode. - **Data Transformation**: The data undergoes a series of transformations, including modular addition, S-box substitution, and bit rotation.

Interface

Signal	Width	In/Out	Description
clk	1	In	Clock signal for synchronous operation.
rst	1	In	Synchronous reset signal.
mode	1	In	Mode selection: 0 for encryption, 1 for decryption.
select	1	In	S-box selection: 0 for TestParameter S-boxes, 1 for CryptoPro S-boxes.
load	1	In	Load signal to input plaintext and start cipher cycles.
done	1	Out	Output signal indicating cipher text is ready.
kload	1	In	Load signal for cipher key.
key	256	In	Cipher key input.
pdata	64	In	Plaintext input.
cdata	64	Out	Ciphertext output.
i	5	-	Internal cipher cycle counter.
enc_index	3	-	Wire for encryption key index calculation.
dec_index	3	-	Wire for decryption key index calculation.
kindex	3	-	Wire for selected key index based on mode.
K	32x8	-	Register array for storing the cipher key.
b	32	-	Register for the most significant half of the input data.
a	32	-	Register for the least significant half of the input data.
state_addmod32	32	-	Wire for the result of modular addition.
state_sbox	32	-	Wire for the result of S-box substitution.
state_shift11	32	-	Wire for the result of 11-bit left rotation.
r_done	1	-	Register indicating readiness of output data.

Timing

The `gost_28147_89` module operates synchronously with the clock signal. The latency for the encryption or decryption process is determined by the 32 cycles required to complete the cipher operations. The `done` signal is asserted once the process is complete, indicating that the `cdata` output is valid and ready for reading. All operations are synchronized to the rising edge of the clock.

Usage

To use the `gost_28147_89` module, follow these steps: 1. **Reset**: Assert the `rst` signal to initialize the module. 2. **Load Key**: Provide the cipher key on the `key` input and assert the `kload` signal to load the key into the internal registers. 3. **Select Mode and S-box**: Set the `mode` input to 0 for encryption or 1 for decryption. Set the `select` input to choose the desired S-box configuration. 4. **Load Data**: Provide the plaintext on the `pdata` input and assert the `load` signal to start the cipher operation. 5. **Wait for Completion**: Monitor the `done` signal. Once asserted, the `cdata` output contains the valid ciphertext. 6. **Read Output**: Capture the `cdata` output once the `done` signal is high.

This module is designed for integration into larger cryptographic systems, providing a robust and flexible encryption/decryption solution compliant with the GOST 28147-89 standard.

Functional Description (Generated by funcgen)

Comprehensive Functional Description of Verilog Design Modules

Module: `gost_28147_89` (File: `gost28147-89.sv`)

Purpose

The `gost_28147_89` module implements the GOST 28147-89 symmetric key block cipher, primarily used for cryptographic applications like data encryption and decryption. This module can execute both encryption and decryption operations, selected via input control signals.

Parameters

- **None**: This module does not contain any parameterized values or constants accessible through parameters.

Ports

- **clk (input, 1-bit)**: The clock signal input for synchronizing operations in a sequential design.
- **rst (input, 1-bit)**: Synchronous reset signal that initializes internal states and registers.
- **mode (input, 1-bit)**: Determines the operation mode: `0` - encrypt, `1` - decrypt.
- **select (input, 1-bit)**: Chooses the S-box: `0` for TestParameter S-boxes and `1` for CryptoPro S-boxes when GOST_R_3411_BOTH is defined.
- **load (input, 1-bit)**: Triggers the loading of plain text and starts the cipher cycles.
- **done (output, 1-bit)**: Indicates the completion of the cipher operation.
- **kload (input, 1-bit)**: Initiates loading of the cipher key into the internal key storage.
- **key (input, 255:0-bit)**: The cipher key input for encryption or decryption, split into eight 32-bit words.
- **pdata (input, 63:0-bit)**: The plain text input to be processed.
- **cdata (output, 63:0-bit)**: The encrypted or decrypted cipher text output.

Internal Signals

- **i (reg, 4:0-bit)**: Serves as a cycle counter for cipher operations, incremented each clock cycle.
- **r_done (reg, 1-bit)**: Flag that becomes true when the cipher operations are completed.
- **enc_index, dec_index, kindex (wire, 2:0-bit each)**: Helper signals for determining the current key index in en/decryption processes.
- **K (reg array, 32:0)**: An array of registers holding segments of the cipher key.
- **a, b (reg, 31:0-bit each)**: Registers holding parts of the input data for intermediate processing.
- **state_addmod32, state_sbox, state_shift11 (wire, 31:0-bit each)**: Used for combinational transformations including modular addition, S-box substitution, and bitwise shifts.

Functionality

Sequential Logic

- **Cycle Counter (`i`)**: Reset or loaded initially; counts from 0 to 31 through subsequent clock cycles to determine cipher progress.
- **Key Storage (`K`)**: Loaded from the `key` input when `kload` is triggered, enabling its use in cipher calculations.
- **Data Registers (`a`, `b`)**: Loaded from `pdata` at the start and continuously updated based on current states through XOR operations and assignments.

Combinational Logic

- **Key Index Calculation (`kindex`)**: Depends on the `mode` and ensures correct key segments are chosen during cipher operations.
- **State Transformations**:
  - **state_addmod32**: Computes `a + K[kindex]`, essential for encryption round calculations.
  - **state_sbox**: Applies S-box transformation using pre-defined macros for substitution.
  - **state_shift11**: Left rotates the `state_sbox` result by 11 bits, part of the encryption algorithm's diffusion process.

Control Logic

The `done` signal, driven by `r_done`, is asserted when all 32 cipher cycles are completed.

Instantiations

- **Includes S-box Macros**: The file incorporates an external header `gost-sbox.vh` containing definitions or operations essential for the S-box transformations.

Inter-Module Connections

Currently, the description pertains to a single self-contained module (`gost_28147_89`). It does not instantiate other modules or rely on any parent module. All logic related to the GOST encryption/decryption is encapsulated within this module, relying on provided input signals and internal processing to produce the output cipher data.

In a broader system, ports such as `clk`, `rst`, `load`, and `kload` might be driven by a higher-level control module, while `key` and `pdata` would be provided by data input modules or interfaces. Similarly, the output `cdata` and `done` can be used by downstream processing or control modules to handle further data processing or state transitions.