ARCHITECTURE SPECIFICATION

Compact 128 bit blockcypher CLEFIA Implementation

Type-III Compact CLEFIA Implementation on FPGA

GNU LGPL Licence

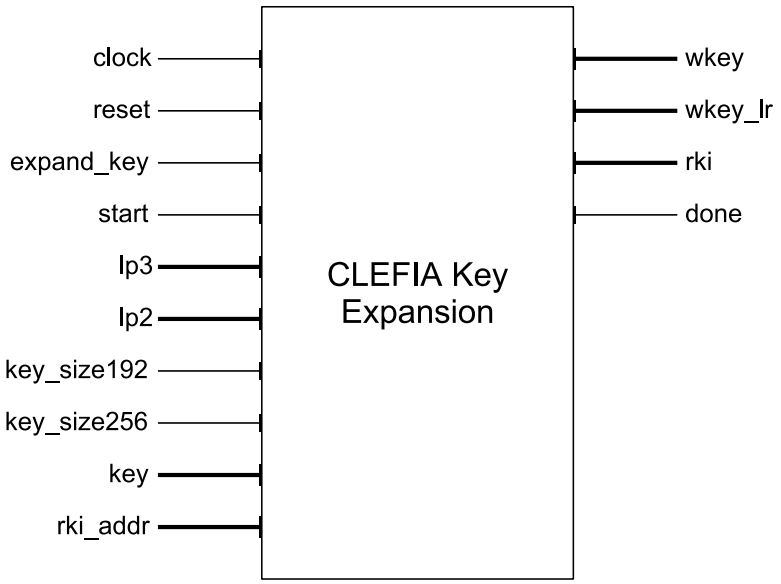
1. Overview

CLEFIA encryption algorithm is a novel symmetrical block ciphering algorithm proposed and developed by SONY Corporation focused on Digital Rights Management (DRM) purposes. This algorithm improves the security of encryption with the use of techniques such as Diffusion Switch Mechanisms, consisting of multiple diffusion matrices in a predetermined order, to ensure immunity against differential and linear attacks, and the use of Whitening Keys, combining data with portions of the key before the first round and after the last round.

The CLEFIA algorithm works with a key size of 128, 192, or 256 bits. It consists of a key scheduling phase and an input data block transformation phase computed over multiple rounds, employing a relatively homogeneous algorithm. The data path of CLEFIA Type-III is composed of a two 4-branch Feistel structure computed for several rounds, defined as $GFN_{8,n}$. However, this architecture places the two structures into the same design through pipeline and proper data scheduling.

The following architecture explores the resources provided by FPGA devices to provide a compact architecture capable of perform CLEFIA key expansion for 128, 192 and 256-bit keys. This design uses FPGA embbeded elements such as RAM memory (LUT) to provide an storage mechanism for generated round keys. This document describes the CLEFIA Key Expansion architecture.

2. Block Diagram



3. Signal Descriptions

Name	Length	Direction	Description
clock	1	Input	system main clock.
reset	1	Input	system main reset signal.
expand_ key	1	Input	control system expansion key operation (1 = enable, 0 = disable).
start	1	Input	process start of operation signal.
lp2	32	Input	Generated L key part (1).
lp3	32	Input	Generated L key part (2).
key_ size192	1	Input	use a 192 bit key size.
key_ size256	1	Input	use a 256 bit key size.
key	256	Input	key used for expansion and whitening.
rki_addr	6	Input	Round Key block memory address.
wkey	128	Output	Whitening key for cyphering.
wkey_lr	256	Output	Whitening key left and right part used for key expansion.
rki	32	Output	Round key output.
done	1	Output	end of operation signal.

4. Architecture Descriptions

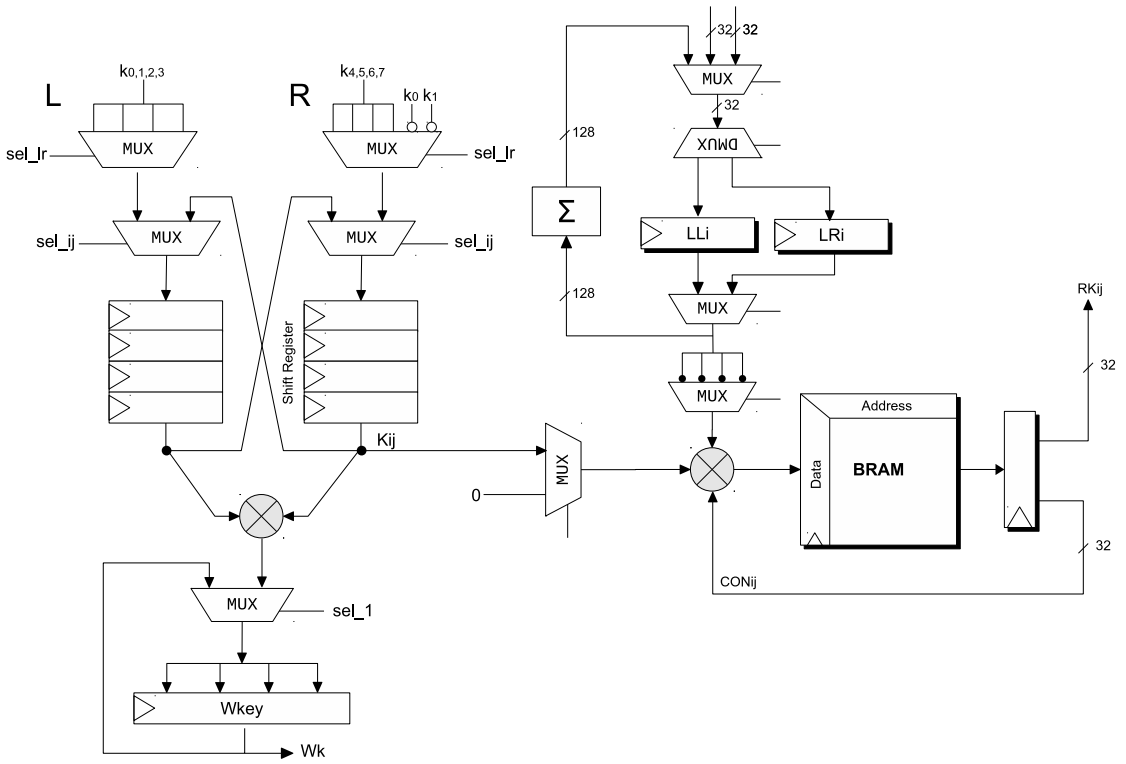
4.1. Basic Operation and Protocols

The key scheduling process is divided into two steps. In the first step the 128 or 256-bit L key value is computed using the the same GFN structure as the ciphering calculation. In the second step the round keys are generated using the L keys and shared key values.

A compact 32-bit data flow can be obtained, resulting in the structure depicted in this documentation. Both the needed constant values and the generated round keys are stored in the same BRAM. This BRAM operates both as a RAM, storing and reading the round keys from the lower part of this memory, and as a ROM, where the constant values are initialized in the upper part of this memory. Note that the address input of the round keys output can be either the round key, when data is being ciphered, or the constant address, when the GFN hardware is being used to generate the L key value.

The architecture can be depicted into two parts. The first one is the Whitening Key generation, accordingly to the key size (128, 192 or 256-bit). This block is responsible for the realization of the Left and Right keys. The values are also divided into 32-bit data. The second is the L key (Left and Right) management and sigma function realization. Given the stored constants, the L and R shared keys and the generated L keys, round keys are generated and stored into the shared BRAM.

4.2. Block Datapath



5. Round Key Memory Block

The Round Key memory Block RAM (BRAM) stores the cyphering and key expansion round keys and the needed constants. The first memory blocks are allocated to cyphering round keys, generated through key expansion. The following blocks are composed by 128 positions due the addressing simplification and controlling. The first portion are the round keys used for key expansion, followed by the needed constants.

The figure bellow describe the Round Key BRAM memory organization.

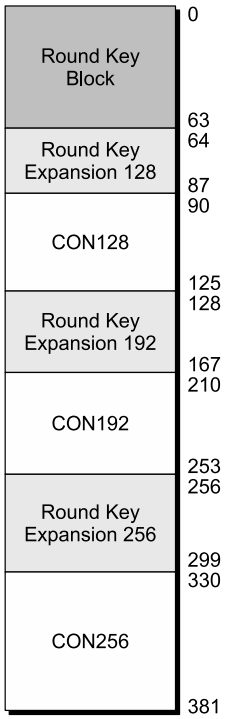
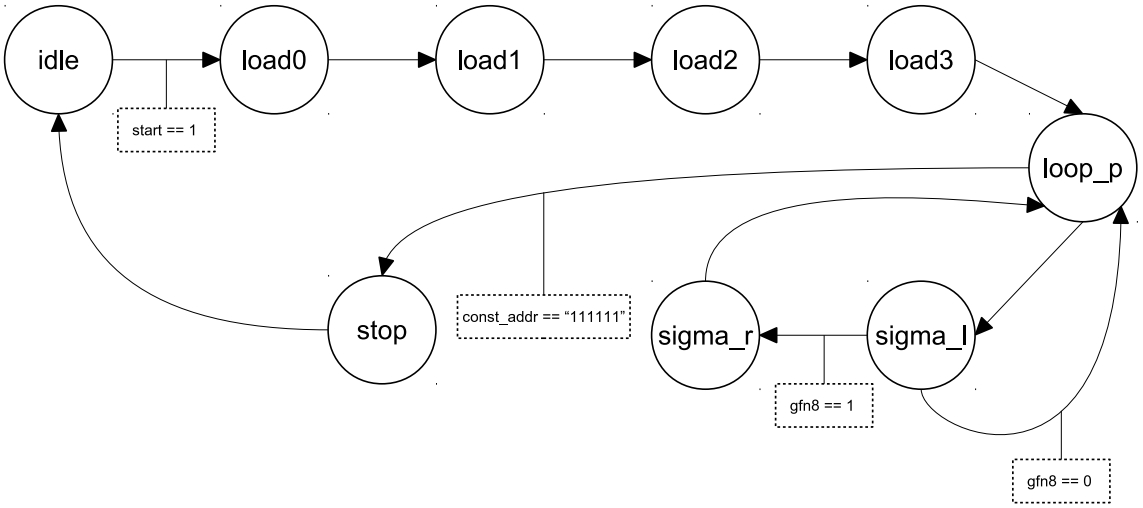


Figure 1: Round Key memory organization.

The table bellow presents a summary of constants an round keys usage.

Key Size	RKeys	Constants
128-bits	24	36
192-bits	40	44
256-bits	44	52

5.1. State Machine



6. Timing

6.1. LKey Generation

Revision History

Date	Description	Owner
06/11/2013	Document conception	João Carlos
07/11/2013	• Design component external interface; • Design basic datapaph; • Describe cyphering state machine; • Place and system design overview;	João Carlos
28/11/2013	• Timing for 128-bit key cyphering; • Timing for 192-bit key cyphering; • Timing for 256-bit key cyphering;	João Carlos