Camellia Hardware Macro Specification

Version	Update	Description
0.1	2007/09/16	Initial version is created
0.1.1	2007/09/21	Timing chart (Fig. 7) is fixed
0.2	2007/09/25	Translated

1. Overview

1.1 Hardware macro overview

The features of this Camellia hardware macro are summarized in Table 1. Only the ECB (Electronic Code Book) mode is supported, but the other modes such as CBC (Cipher Block Chaining) can be easily supported by using additional data buffers and a control circuit.

Table 1 CAST-128 hardware macro overview

Algorithm	Camellia
Data block size	128 bits
Key size	128 bits
Mode of operation	Electronic Code Book (ECB)
Source file name	Camellia.v
Description Language	Verilog-HDL
Top module name	Camellia
Throughput	128 bit / 23 clock
Round keys	On-the-fly

1.2 Algorithm overview

Camellia is a Feistel-type block cipher jointly developed by NTT (Nippon Telegraph and Telephone Corp.) and Mitsubishi Electric. Camellia supports 128-, 192-, and 256-bit keys. In the following, we describe 128-bit version. The detailed algorithm is described in the specification [1].

Fig. 1 shows the data randomization block including 18-round Feistel Network with functions F and FL/FL^{-1}. A 64-bit round function F consists of eight 8-bit S-boxes and an XOR network. Two 64-bit linear functions FL and FL^{-1} are given by AND, OR, XOR, and 1-bit rotation.

Round keys kw_1~kw_4, kl_1~kl_4, and k_1~k_{18} are used in initial/final key addition, F function, and FL/FL^{-1} functions. The round keys are generated from a secret key K_L and an intermediate key K_A according to Table 1. K_A is generated from K_L using Feistel Network and F-function as shown in Fig. 2,

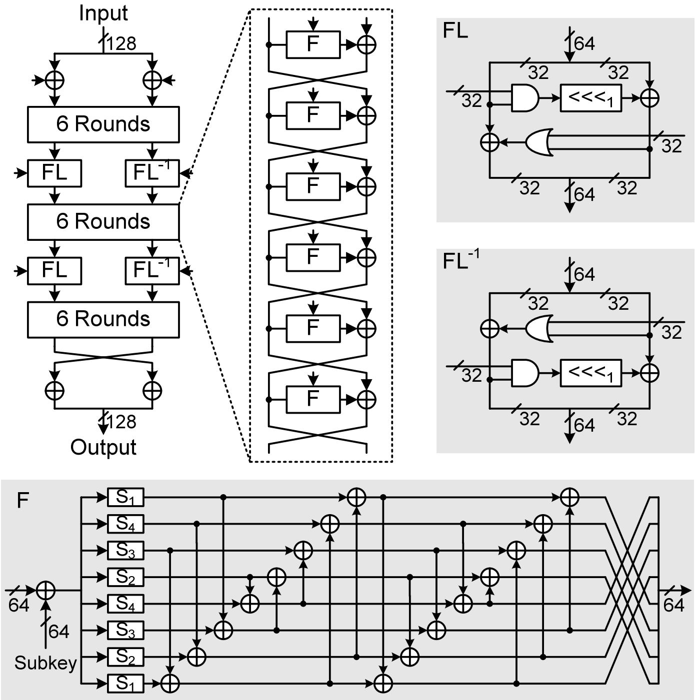

Fig. 1 Camellia encryption algorithm

Table 2 Generation rule for round keys

Initial XOR	$kw_{1(64)}$	$(K_L \lll_0)_{L(64)}$
	$kw_{2(64)}$	$(K_L \lll_0)_{R(64)}$
F (Round 1)	$k_{1(64)}$	$(K_A \lll_0)_{L(64)}$
F (Round 2)	$k_{2(64)}$	$(K_A \lll_0)_{R(64)}$
F (Round 3)	$k_{3(64)}$	$(K_L \lll_{15})_{L(64)}$
F (Round 4)	$k_{4(64)}$	$(K_L \lll_{15})_{R(64)}$
F (Round 5)	$k_{5(64)}$	$(K_A \lll_{15})_{L(64)}$
F (Round 6)	$k_{6(64)}$	$(K_A \lll_{15})_{R(64)}$
FL	$kl_{1(64)}$	$(K_A \lll_{30})_{L(64)}$
FL^{-1}	$kl_{2(64)}$	$(K_A \lll_{30})_{R(64)}$
F (Round 7)	$k_{7(64)}$	$(K_L \lll_{45})_{L(64)}$
F (Round 8)	$k_{8(64)}$	$(K_L \lll_{45})_{R(64)}$
F (Round 9)	$k_{9(64)}$	$(K_A \lll_{45})_{L(64)}$
F (Round 10)	$k_{10(64)}$	$(K_L \lll_{60})_{R(64)}$
F (Round 11)	$k_{11(64)}$	$(K_A \lll_{60})_{L(64)}$
F (Round 12)	$k_{12(64)}$	$(K_A \lll_{60})_{R(64)}$
FL	$kl_{3(64)}$	$(K_L \lll_{77})_{L(64)}$
FL^{-1}	$kl_{4(64)}$	$(K_L \lll_{77})_{R(64)}$
F (Round 13)	$k_{13(64)}$	$(K_L \lll_{94})_{L(64)}$
F (Round 14)	$k_{14(64)}$	$(K_L \lll_{94})_{R(64)}$
F (Round 15)	$k_{15(64)}$	$(K_A \lll_{94})_{L(64)}$
F (Round 16)	$k_{16(64)}$	$(K_A \lll_{94})_{R(64)}$
F (Round 17)	$k_{17(64)}$	$(K_L \lll_{111})_{L(64)}$
F (Round 18)	$k_{18(64)}$	$(K_L \lll_{111})_{R(64)}$
Final XOR	$kw_{3(64)}$	$(K_A \lll_{111})_{L(64)}$
	$kw_{4(64)}$	$(K_A \lll_{111})_{R(64)}$

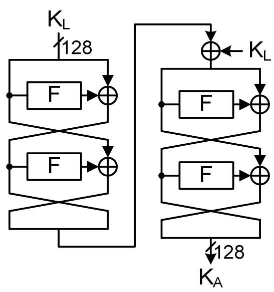

Fig. 2 Intermediate key generation.

2. I/O ports

I/O ports of the Camellia macro are summarized in Table 3.

Table 3 I/O ports

Port name	Direction	Width	Description
Kin	In	128	Key input
Din	In	128	Data input
Dout	Out	128	Data output
Krdy	In	1	When Krdy=1, a secret key is latched in an internal register, and the intermediate key generation process is executed. If Drdy and Krdy assigned to 1 at the same time, Krdy=1 has priority.
Drdy	In	1	When Drdy=1, a plaintext (or ciphertext) data is latched in an internal register and the encryption (or decryption) process is started.
EncDec	In	1	Encryption and decryption are executed when EncDec=0 and EncDec=1, respectively. Bit data on the port EncDec is stored in an internal register when encryption or decryption starts in response to Drdy=1.
RSTn	In	1	Reset signal. Sequencer logic and internal registers are reset when this signal is assigned to 0. The reset can be executed any time when the clock signal CLK is input, even if the enable signal EN=0.
EN	In	1	Enable signal. When EN=1, this macro is activated.
CLK	In	1	System clock. All registers are synchronized with the rising edge of this signal.
BSY	Out	1	Busy status flag. This signal is assigned to 1 while an encryption, decryption, or key generation process is executed. When this signal is 1, both Drdy and Krdy are ignored.
Kvld	Out	1	When round-key generation process is completed, this signal becomes 1 during the next one clock cycle, and then it goes 0. Soon after that, encryption and decryption processes are ready to start.
Dvld	Out	1	When encryption or decryption process is completed and cipher text or plain text are ready on the data output port Dout, this signal becomes 1 during the next one clock cycle, and then it goes 0.

3. Hardware Architecture

3.1 Datapath

A datapath of the Camellia macro is shown in Fig. 3. This macro executes 1-round operation in 1 clock cycle. A 128-bit block of plaintext are encrypted / decrypted in 16 clocks.

A secret key is contained in an internal register kl through a 128-bit port Kin in the key-scheduling. Then the intermediate key generation is started in the data randomization block. The obtained intermediate key is set to an internal register ka after 6 clocks. Round keys are generated from the values in kl and ka on the fly.

An input data (plaintext for encryption, ciphertext for decryption) is set to an internal register Dout_reg through a 128-bit port Din in the data-randomization block, An output data (ciphertext for encryption, plaintext for decryption) is obtained from a 128-bit port Dout.

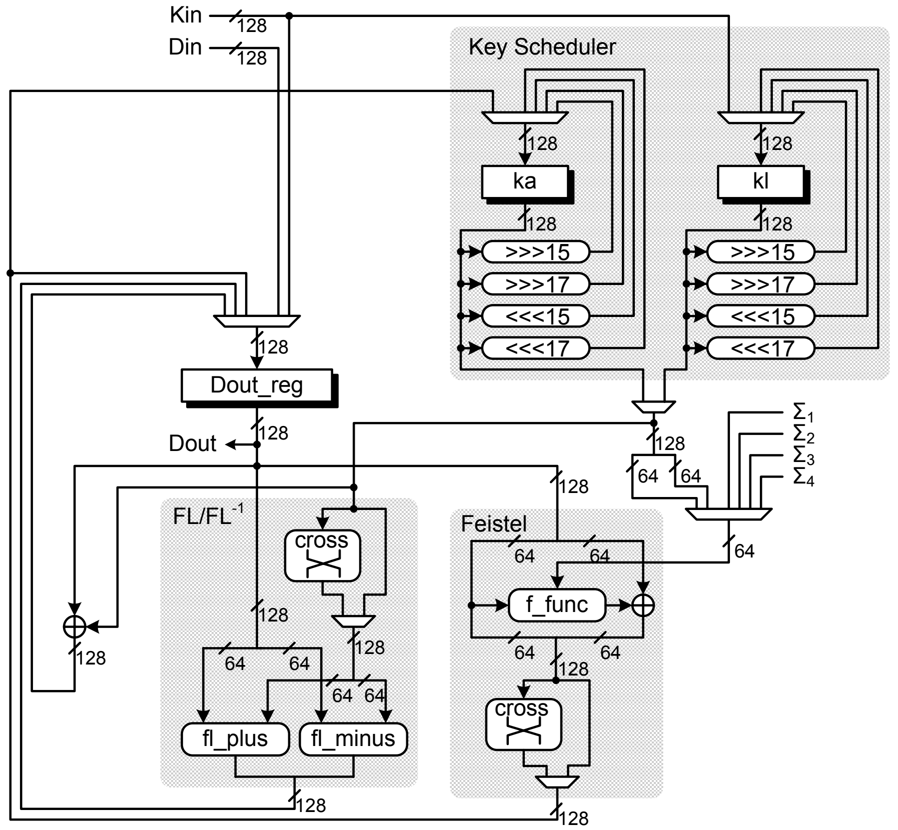

Fig.3 Datapath

3.2 State Diagram

The state diagram of the Camellia sequencer and its description are shown in Fig. 4 and Table 4, respectively.

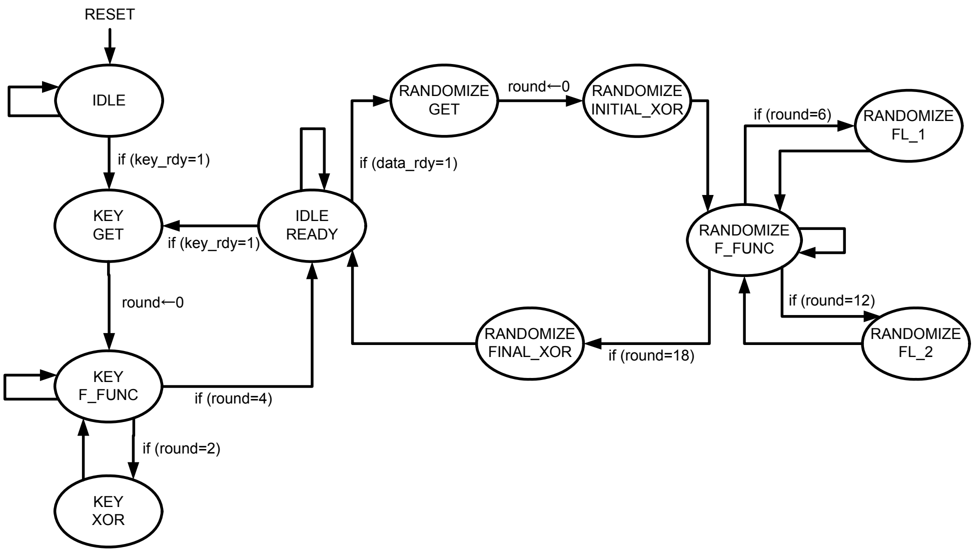

Fig. 4 State diagram of sequencer

Table 4 State diagram of sequencer logic

State	Description
IDLE	Initial state. Only key input is accepted.
IDLE_READY	Idling state where intermediate-key generation is finished. Both data input and key input are accepted.
KEY_GET	States for intermediate-key generation.
KEY_F_FUNC	
KEY_XOR	
RANDOMIZE_GET	States for data randomization (in encryption or decryption).
RANDOMIZE_INITIAL_XOR	
RANDOMIZE_F_FUNC	
RANDOMIZE_FL_1	
RANDOMIZE_FL_2	
RANDOMIZE_FINAL_XOR	

4. Timing Chart

Fig. 5 shows the timing chart of the key scheduling, encryption, and decryption process for the Camellia macro in the minimum cycles for the control signals. The operation are performed as follows.

CLK1: The sequencer logic is initialized by resetting RSTn to 0.

CLK2: By asserting Krdy=1, the 128-bit secret key on Kin is stored to an internal register. Soon after that, the key scheduling process is started, and BSY is set to 1.

CLK3~CLK8: The key scheduling process takes 6 clocks, and thus Kvld and BSY are set to 1 and 0 in CLK8, respectively. The sequencer goes to the idling state "IDLE_READY."

CLK9: By asserting Drdy=1, the 128-bit input (plaintext) and the control signal EncDec are stored into internal registers. The encryption process is started in accordance with EncDec=0, and BSY is set to 1.

CLK10~32: The encryption takes 23 clocks, and thus it is completed in CLK32. The output data (ciphertext) is output from Dout and Dvld is set to 1 only in the 23rd clock (i.e., CLK32). The sequencer is set to "IDLE_READY," and BSY goes to 0 in CLK32.

CLK33: By asserting Drdy=1, the next operation is started. The 128-bit input (ciphertext) and the control signal EncDec are stored into internal registers. The decryption process is started in accordance with EncDec=1, and BSY is set to 1.

CLK34~57: The decryption also takes 23 clocks. and thus it is completed in CLK57. The output data (plaintext) is output from Dout and Dvld is set to 1 only in the 23rd clock (i.e., CLK56). The sequencer is set to "IDLE_READY," and BSY goes 0 in CLK57.

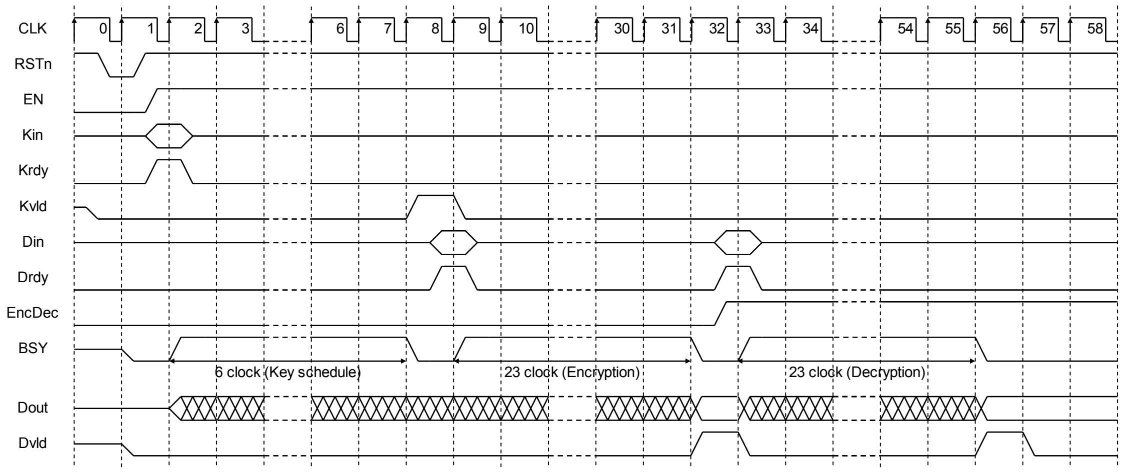

Fig. 5 Timing Chart

5. Reference

[1] K.Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita, "Specification of Camellia – a 128 -bit Block Cipher," Sep. 2001, http://info.isl.ntt.co.jp/crypt/camellia/dl/01espec.pdf