CAST-128 Hardware Macro Specification

Version	Update	Description
0.1	2007/09/04	Initial version is created
0.2	2007/09/25	Timing chart (Fig. 7) is fixed

1. Overview

The features of this CAST-128 hardware macro are summarized in Table 1. Detailed algorithm is described in the RFC2144 [1] specification. 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	CAST-128
Data block size	64 bits
Key size	128 bits
Mode of operation	Electronic Code Book (ECB)
Source file name	CAST128.v
Description Language	Verilog-HDL
Top module name	CAST128
Throughput	128 bit / 16 clock
Round keys	Pre calculation & On-the-fly

2. I/O ports

I/O ports of the CAST-128 macro are summarized in Table 2.

Table 2 I/O ports

Port name	Direction	Width	Description
Kin	In	128	Key input
Din	In	64	Data input
Dout	Out	64	Data output
Krdy	In	1	When Krdy=1, a secret key is latched in an internal register, and initial key scheduling process is executed. If Drdy and Krdy assigned to be '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 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. 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 during encryption, decryption, or key generation process is executed. When this signal is 1, signals 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 Top Module

This hardware macro consists of sequencer and two data processing modules (a key scheduler and an encryption block) as shown in Fig. 1.

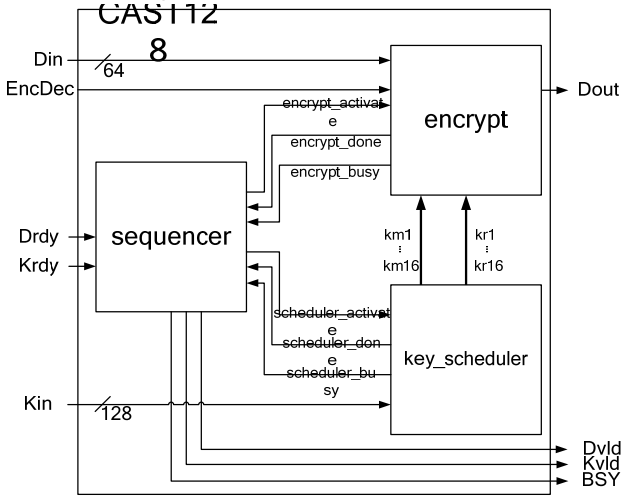

Fig.1 Block Diagram of top module CAST128

3.2 Key Scheduler

Fig. 2 shows the key scheduler, and the typical steps are described below. The first four steps are performed with S-boxes S5~S8 and XOR operations as follows.

$$\begin{cases} z0z1z2z3 = x0x1x2x3 \oplus S5[xD] \oplus S6[xF] \oplus S7[xC] \oplus S8[xE] \oplus S7[x8] & (1) \\ z4z5z6z7 = x8x9xAxB \oplus S5[z0] \oplus S6[z2] \oplus S7[z1] \oplus S8[z3] \oplus S8[xA] & (2) \\ z8z9zAzB = xCxDxExF \oplus S5[z7] \oplus S6[z6] \oplus S7[z5] \oplus S8[z4] \oplus S5[x9] & (3) \\ zCzDzEzF = x4x5x6x7 \oplus S5[zA] \oplus S6[z9] \oplus S7[zB] \oplus S8[z8] \oplus S6[xB] & (4) \end{cases}$$

When one equation is performed in one clock, two sets of S-boxes (S5~S8) are required, and thus we modified Equations (1)~(4) as follows, where only one set of S-boxes are used in each equation.

$$\begin{cases} z0z1z2z3 = x0x1x2x3 \oplus S7[x8] & (5) \\ z4z5z6z7 = x8x9xAxB \oplus S8[xA] & (6) \\ z8z9zAzB = xCxDxExF \oplus S5[x9] & (7) \\ zCzDzEzF = x4x5x6x7 \oplus S6[xB] & (8) \end{cases}$$

$$\begin{cases} z0z1z2z3 = z0z1z2z3 \oplus S5[xD] \oplus S6[xF] \oplus S7[xC] \oplus S8[xE] & (9) \\ z4z5z6z7 = z8z9zAzB \oplus S5[z0] \oplus S6[z2] \oplus S7[z1] \oplus S8[z3] & (10) \\ z8z9zAzB = zCzDzEzF \oplus S5[z7] \oplus S6[z6] \oplus S7[z5] \oplus S8[z4] & (11) \\ zCzDzEzF = z4z5z6z7 \oplus S5[zA] \oplus S6[z9] \oplus S7[zB] \oplus S8[z8] & (12) \end{cases}$$

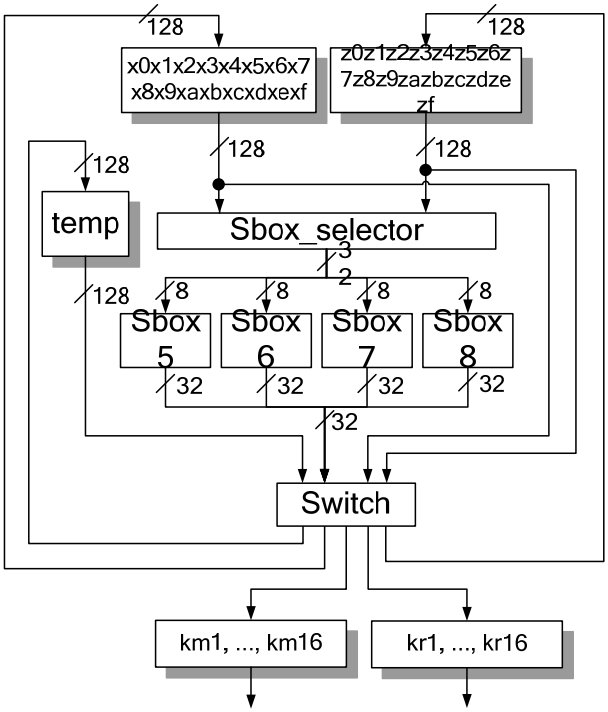

Fig.2 Datapath architecture for key scheduler

Fig. 3 shows the activated datapath of the key scheduler when Equations (5)~(8) are performed. The four equations can be executed in the same clock because they do not use the same sbox.. Four

Equations (9)~(12) are executed one by one in each cycle. The activated datapath to perform Equation (9) is shown in Fig. 4. The following steps are executed similary by changing selection of (i) registers feeding data to Sbox, (ii) registers with which Sbox outputs are XORed, and (ii) registers to be updated. The selections for (i) and (ii) are controlled by "sbox_selector" and those for (iii) are done by "sequencer."

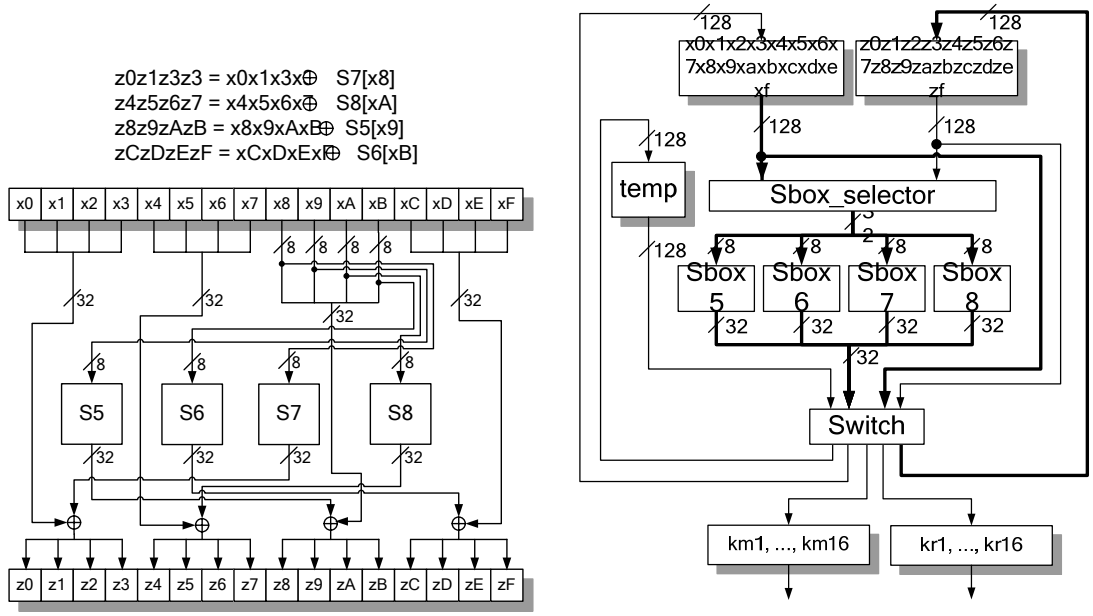

Fig.3 Operation of key scheduler

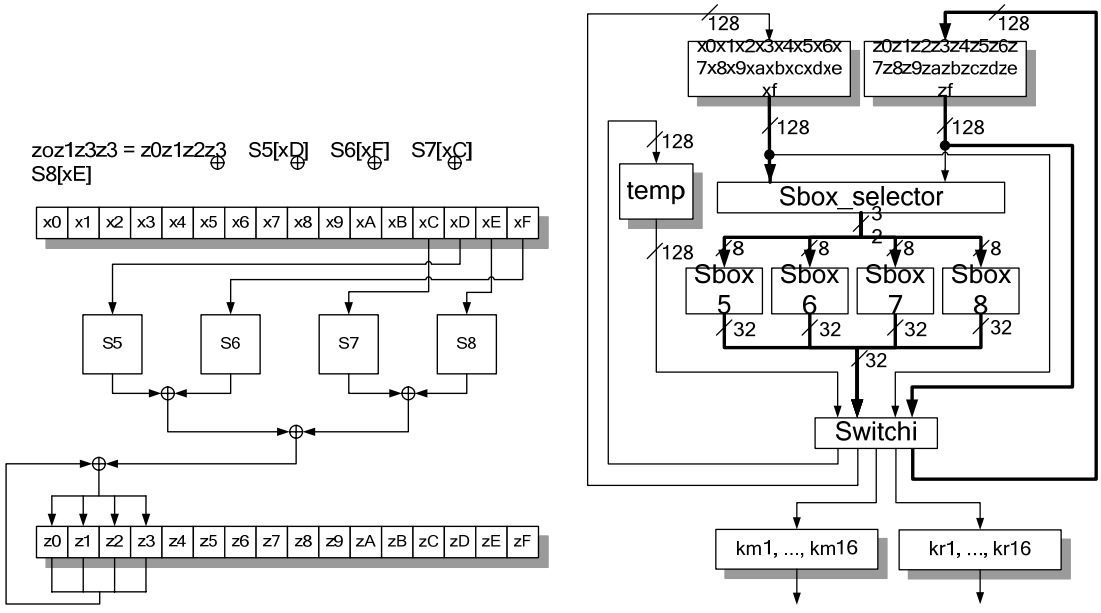

Fig.4 Operation of key scheduler

3.3 Encryption Block

The data path architecture of the encryption block is shown in Fig. 5. The CAST-128 hardware macro The Fiestel-network cipher CAST-128 switches the following three round functions according

to Table 3.

$$\text{Type 1:} \quad \begin{cases} I = ((Kmi \ + \ D) \ <<< \ Kri) \\ f = ((S1[Ia] \oplus S2[Ib]) - S3[Ic]) + S4[Id] \end{cases} \tag{13}$$

$$\text{Type 2:} \quad \begin{cases} I = ((Kmi \oplus D) <<< Kri) \\ f = ((S1[Ia] - S2[Ib]) + S3[Ic]) \oplus S4[Id] \end{cases} \tag{14}$$

$$\text{Type 3:} \quad \begin{cases} I = ((Kmi - D) <<< Kri) \\ f = ((S1[Ia] + S2[Ib]) \oplus S3[Ic]) - S4[Id] \end{cases} \tag{15}$$

Table. 3 Scheduling of round functions

Round Function	Used rounds
Type 1	1, 4, 7, 10, 13, 16
Type 2	2, 5, 8, 11, 14
Type 3	3, 6, 9, 12, 15

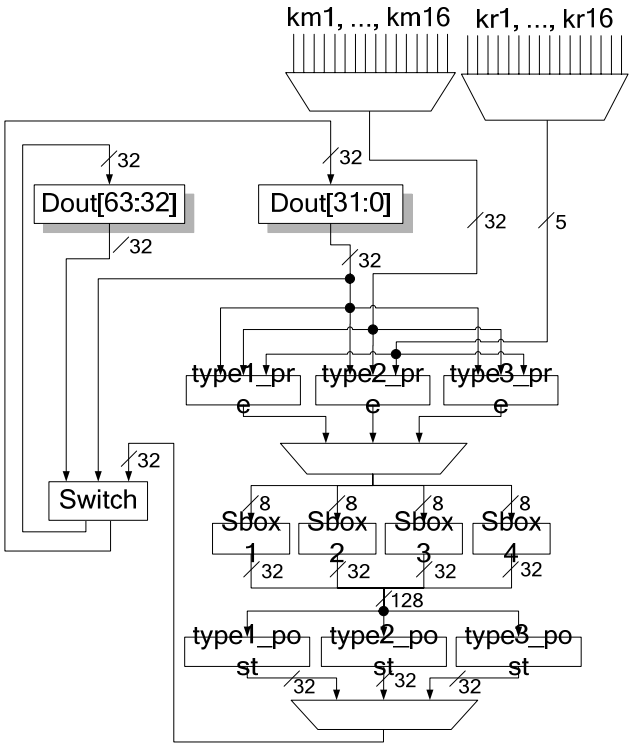

Fig. 5 Datapath for encryption block

In order to reduce hardware resources, blocks calculating 'I' and 'f' in Equations (13)~(15) are designed separately , and S-boxes are shared between three round function. In Fig. 6, the former process of 'I' is performed by the blocks type1_pre, type2_pre, and type3_pre, and the latter process 'f' is done by the blocks type1_post, type2_post, and type3_post. The thick lines in Fig. 6 show the activated datapath when the round function Type 1 is executed.

3.4 State Diagram

The state diagram of the CAST-128 sequencer and its description are shown in Fig. 6 and Table 4, respectively.

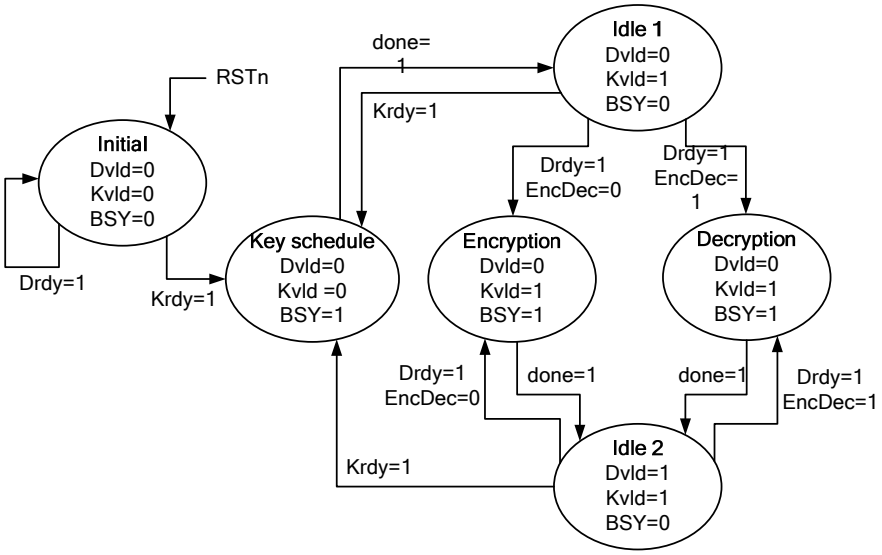

Fig. 6 State diagram of sequencer logic

Table 4 Sequencer States

State	Description
Initial	Initial state
Key schedule	Key scheduling for round key generation is executed.
Encryption	Encryption is performed
Decryption	Decryption is performed
Idle 1	Idle state with invalid data on the port Dout.
Idle 2	Idle state with valid data on the port Dout.

4. Timing Chart

Fig. 7 shows the timing chart of the key scheduling and encryption process of the CAST-128 macro in the minimum cycles for the control signals. The operations are performed as follows.

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

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

CLK3~130: The key scheduling takes 128 clocks, and thus the flags Kvld and BSY are set to 1 and 0 in CLK130, respectively. The sequencer goes to the idling state “Idle 1.”

CLK131: The signal Drdy is set to 1, and the 64-bit plaintext is stored into an internal register.

CLK132: The encryption process is started in accordance with EncDec=0, and BSY is set to 1.

CLK133~148: The encryption takes 16 clocks, and thus it is completed in CLK148. The ciphertext is output from the 64-bit port Dout. The flag BSY goes 0, and Dvld goes to 1. The sequencer

state is set to “Idle2.”

CLK149: By asserting Drdy=1, another operation is started. The 64-bit input (ciphertext) is stored into an internal register. The decryption process is started in accordance with EncDec=1. The flag BSY is set to 1, and Dvld goes 0.

CLK150~165: The decryption also takes 16 clocks. and thus it is completed in CLK165. The output data (plaintext) is output from the 64-bit port Dout. The flag BSY goes 0, and Dvld is set to 1.

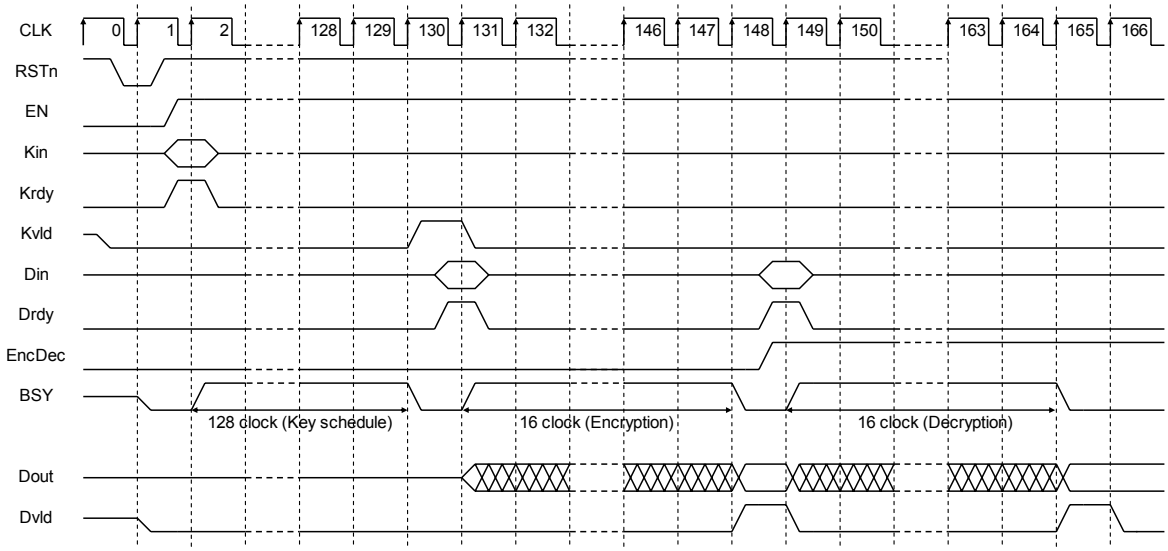

Fig. 7 Timing Chart

5. Reference

[1] C. Adams, “The CAST-128 Encryption Algorithm,” RFC2144 (Informational), May 1997.