

Standard Cryptographic LSI Specification
~ Countermeasures against Side Channel Attacks (65nm) ~

[Version 0.9]



August 27, 2010

**National Institute of Advanced Industrial Science and
Technology (AIST)
Research Center of Information Security (RCIS)**

1. OUTLINE.....	3
2. EXTERNAL SPECIFICATION	4
2.1. INPUT OUTPUT SIGNAL	4
2.2. PIN ASSIGNMENT	4
2.3. INPUT OUTPUT TIMING	9
2.4. INTERFACE REGISTERS.....	11
2.5. OPERATING PROCEDURE	14
3. DETAILED SPECIFICATION	16
3.1. OVERALL BLOCK DIAGRAM	16
3.2. HIERARCHICAL STRUCTURE	16
3.3. EXTERNAL INTERFACE CIRCUIT	18
3.4. INTERFACES FOR CRYPTOGRAPHIC ALGORITHM CORES	19
3.5. DETAIL OF INTERFACE REGISTERS	22
3.6. CLOCK TREE.....	30
3.7. RESET.....	31
3.8. SUPPLEMENTARY FUNCTIONS AND CONSIDERATIONS.....	32
4. THE ENVIRONMENT FOR RUN VERIFICATION BY LOGIC SIMULATION.....	37
4.1. THE OUTLINE OF RUN VERIFICATION MODULE	37
4.2. THE FUNCTION OF RUN VERIFICATION MODULE.....	37
4.3. THE VERIFICATION AND VERIFICATION RESULT OF RUN VERIFICATION MODULE	37
5. THE RESTRICTION OF LOGIC SYNTHESIS.....	39
6. PHYSICAL LAYOUT OF THE LSI.....	40
6.1. DESIGN ENVIRONMENT	40
6.2. RESULT OF SYNTHESIS	40
6.3. POWER SOURCE PLAN	42
6.4. MACRO LAYOUT.....	46
6.5. MAIN MODULE LAYOUT	47
6.6. REPORT ON CELL AREA	48
6.7. SIGNAL WIRING.....	49
6.8. ABOUT INSERTION OF DUMMY METAL	52
6.9. ON THE CHIP'S ORIENTATION MATCHING	53
6.10. THE RESULT OF VERIFYING POWER SEPARATION	56
7. IR-DROP VERIFICATION	57
8. X-TALK NOISE TEST	58
8.1. ABOUT X-TALK NOISE TEST.....	58
8.2. TEST RESULT.....	58
9. STA TEST.....	61
9.1. STA CONDITION	61
9.2. SUMMARY ON CLOCK GATING TIMING ERROR	61
9.3. MAXIMUM OPERATING SPEED	63
9.4. NOT ANNOTATED ANALYSIS.....	64
10. FORMAL VERIFICATION	65
11. LAYOUT TEST.....	67
11.1. DRC	67

11.2.	ANT	68
11.3.	DFM.....	69
11.4.	FL	70
11.5.	LVS	71
12.	THE SUMMARY OF THE RESULT OF EACH TEST	72
13.	CRYPTOGRAPHIC HARDWARE IPS	73
13.1.	AES0 (COMPOSITE FIELD S-BOX)	73
13.2.	AES1/AES2/AES3/AES4 (VARIETY OF S-BOXES).....	76
13.3.	AES5 (CTR MODE)	78
13.4.	AES6 (FA COUNTERMEASURE)	84
13.5.	AES7 (ROUND KEY PRE-CALCULATION)	88
13.6.	AES8 (MAO).....	90
13.7.	AES9 (MDPL)	91
13.8.	AES10 (THRESHOLD IMPLEMENTATION)	92
13.9.	AES11 (WDDL)	92
13.10.	AES12/AES13 (PSEUDO RSL).....	93
13.11.	CAMELLIA.....	93
13.12.	CAST-128	96
13.13.	DES.....	99
13.14.	ECC	102
13.15.	MISTY1	105
13.16.	RSA.....	108
13.17.	SEED	112
13.18.	TDES	115
13.19.	CLEFIA	118
REFERENCES.....	121

1. Outline

The *dedicated LSI with side channel attacks countermeasures applied standard cryptographic algorithms* (below, *cryptographic LSI*) is an LSI which implements public key cryptographic algorithm RSA, elliptic curve cryptographic (ECC) algorithm, and several other cryptographic algorithms with a goal, i.e., to perform the evaluation of various implementation attacks such as differential power analysis. The cryptographic LSI is manufactured by 65nm process of Fujitsu Microelectronics Corp. through E-shuttle corp., and is sealed (packaged) with 160pin Ceramic QFP.

The number of implemented cryptographic algorithms is 10. For AES, since there are 14 different forms of implementation, in total, there are 23 cryptographic algorithm cores being loaded.

AES (key length: 128bit)

0. S-Box implementation→composite, encryption/decryption support
 1. S-Box implementation→case statement, encryption only support
 2. S-Box implementation→AND-XOR implementation (1-Stage), encryption only support
 3. S-Box implementation→AND-XOR implementation (3-Stage), encryption only support
 4. S-Box implementation→composite, encryption only support
 5. CTR mode support pipeline implementation
 6. Implementation for fault analysis resistance evaluation
 7. Implementation with precomputation of round key
 8. Implementation with countermeasures against DPA (Masked AND Operation)
 9. Implementation with countermeasures against DPA (MDPL)
 10. Implementation with countermeasures against DPA (Threshold Implementation)
 11. Implementation with countermeasures against DPA (WDDL)
 12. Implementation with countermeasures against DPA (Pseudo RSL)
 13. Implementation with countermeasures against DPA (For evaluating the effect of pseudo RSL)
- Camellia (key length: 128) encryption/decryption support
 - SEED: encryption/decryption support
 - MISTY1: encryption/decryption support
 - Triple-DES: 3Key, encryption/decryption support
 - DES: encryption/decryption support
 - CAST128: encryption/decryption support
 - RSA: 1024bit modular exponentiation
 - ECC: key length is 64bit. Point scalar multiplication on the field with characteristic 2.
 - CLEFIA: encryption/decryption support

The main functions:

- Running cryptographic algorithm
- Equivalence to the controlling FPGA of Cryptographic Evaluation Board FPGA (below, SASEBO) developed at FY 2006 and the interfaces.
- Trigger signal output for power information sampling, etc. (trigger signal output is deterrable).
- For evaluation of fault analysis, outputting the pre-setup intermediate value and intermediate key of algorithm process (support only for AES core no.6 above).
- For evaluation of fault analysis, outputting the intermediate value and the intermediate key at the time when the fault occurs (support only for AES core no.6 above).
- Support for self-running mode, i.e., for each 0.3 seconds, automatically continues the cryptographic process (support only for AES core no.0 above).

2. External Specification

Below is the external specification of the cryptographic LSI.

2.1. Input Output Signal

Input output signals of the cryptographic LSI are shown in Table 2-1.

Table 2-1 Input Output Signals

Classification (Total)	Signal Name	Number	Significance	Direction (LSI's view)	Usage, Notes
System (11)	CLKA	1	--	IN	For 24MHz clock input of LSI internal circuit. Exactly as same as CLKB, or higher frequency clock input.
	CLKB	1	--	IN	Clock for LSI interface circuit.
	HRST_N	1	L	IN	SSignal Reset generated by the reset circuit on board. Asynchronous reset input.
	LEDO[1:0]	2	L	OUT	Output for LED drive (NC pin)
	SWIN[3:0]	4	--	IN	Input for switch (NC pin)
	PHIN[1:0]	2	--	IN	Input for pin header (NC pin)
Bus Control (4)	WR_N	1	L	IN	Write instruction
	RD_N	1	L	IN	Read instruction
	RSV0	1	--	IN	(NC pin)
	RSV1	1	--	IN	(NC pin)
Bus Address (16)	A[15:0]	16	--	IN	
Bus Data (32)	DI[15:0]	16	--	IN	Input data
	DO[15:0]	16	--	OUT	Output data
For Evaluation (13)	START_N	1	L	OUT	Start the target process
	END_N	1	L	OUT	End the target process
	(TRIG0)	1	--	OUT	(NC pin)
	(TRIG1)	1	--	OUT	(NC pin)
	EXEC	1	H	OUT	Processing the target process
	STATE[4:0]	5	--	OUT	Show the selected IP
	MON[3:0]	4	--	OUT	For internal monitor (detail unspecified)
Total		77			

2.2. Pin Assignment

The pin assignment of cryptographic LSI is shown in Table 2-2, and the illustration of top-view's pin assignment is shown in Figure 2-1. The cryptographic LSI developed in FY 2008 used the manufacturing rule 130nm and 90nm, and thus the die size was large and it was possible to make both the number of die-pads and the number of package's pins equal to 160. This time, since the manufacturing rule is set to 65nm, the number of die-pads is reduced to 136. Due to this, in order to maintain the compatibility with the already developed cryptographic LSI with 160-pin package, we use the N.C pins of the cryptographic LSI developed in FY 2008 and do not connect several pins of

power source to the die, treating them as N.C pins.

Moreover, since we try reducing the noise to increase the accuracy of measurement of the electric power and the electromagnetic wave, the cryptographic LSI is structured such that VDD/VSS of the internal LSI and the input output buffer are separated.

Table 2-2 Cryptographic LSI Pin Assignment (1/4)

Pin NO	Signal Name	I/O	I/F Voltage	output	I/O Buffer	Function
1	VSS*					core GND
2	VSS*					core GND
3	VSS IO					I/O GND
4	(SWIN[3])					N.C
5	(SWIN[2])					N.C
6	(SWIN[1])					N.C
7	(SWIN[0])					N.C
8	VDE					I/O 3.3V
9	(PHIN[1])					N.C
10	(PHIN[0])					N.C
11	N.C					N.C
12	N.C					N.C
13	VSS IO					I/O GND
14	N.C					N.C
15	N.C					N.C
16	N.C					N.C
17	N.C					N.C
18	N.C					N.C
19	VDE					I/O 3.3V
20	VDD					core 1.2V
21	VSS					core GND
22	VSS IO					I/O GND
23	N.C					N.C
24	N.C					N.C
25	N.C					N.C
26	(RSV1)					N.C
27	(RSV0)					N.C
28	VSS IO					I/O GND
29	A[15]	I	3.3V		IOCB2EITNNMXA02	Address Bus
30	A[14]	I	3.3V		IOCB2EITNNMXA02	Address Bus
31	A[13]	I	3.3V		IOCB2EITNNMXA02	Address Bus
32	A[12]	I	3.3V		IOCB2EITNNMXA02	Address Bus
33	VDE					I/O 3.3V
34	A[11]	I	3.3V		IOCB2EITNNMXA02	Address Bus
35	A[10]	I	3.3V		IOCB2EITNNMXA02	Address Bus
36	A[9]	I	3.3V		IOCB2EITNNMXA02	Address Bus
37	A[8]	I	3.3V		IOCB2EITNNMXA02	Address Bus
38	VSS IO					I/O GND
39	VSS*					core GND
40	VSS*					core GND

• The symbol “()” in「Signal Name」is for future extension, it is the N.C pin in the cryptographic LSI

Table 2-2 Cryptographic LSI Pin Assignment (2/4)

Pin NO	Signal Name	I/O	I/F Voltage	output	I/O Buffer	Function
41	VDD					core 1.2V (N.C in the die)
42	VDE					I/O 3.3V (N.C in the die)
43	A[7]	I	3.3V		IOCB2EITNNMXA02	Address Bus
44	A[6]	I	3.3V		IOCB2EITNNMXA02	Address Bus
45	A[5]	I	3.3V		IOCB2EITNNMXA02	Address Bus
46	A[4]	I	3.3V		IOCB2EITNNMXA02	Address Bus
47	VSS IO					I/O GND
48	VDD					core 1.2V
49	A[3]	I	3.3V		IOCB2EITNNMXA02	Address Bus
50	A[2]	I	3.3V		IOCB2EITNNMXA02	Address Bus
51	A[1]	I	3.3V		IOCB2EITNNMXA02	Address Bus
52	A[0]	I	3.3V		IOCB2EITNNMXA02	Address Bus
53	VDE					I/O 3.3V
54	VSS					core GND
55	VSS IO					I/O GND
56	CLKB	I	3.3V		IOCB2EITSNMXA02	Clock.Schmitt
57	VSS IO					I/O GND
58	CLKA	I	3.3V		IOCB2EITSNMXA02	Clock.Schmitt
59	VSS IO					I/O GND
60	VDD					core 1.2V
61	N.C					core GND
62	VSS					I/O GND
63	HRST_N	I	3.3V		IOCB2EITSNMXA02	Reset .Schmitt
64	N.C					I/O GND
65	WR_N	I	3.3V		IOCB2EITNNMXA02	Write Instruction
66	RD_N	I	3.3V		IOCB2EITNNMXA02	Read Instruction
67	VDE					I/O 3.3V
68	VSS					core GND
69	DO[15]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
70	DO[14]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
71	DO[13]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
72	DO[12]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
73	VSS IO					I/O GND
74	VDD					core 1.2V
75	DO[11]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
76	DO[10]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
77	DO[9]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
78	DO[8]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
79	VDE					I/O 3.3V (N.C in the die)
80	VDD					core 1.2V (N.C in the die)

•The symbol “()” in “Signal Name” is for future extension, it is the N.C pin in the cryptographic LSI.

Table 2-2 Cryptographic LSI Pin Assignment (3/4)

Pin NO	Signal Name	I/O	I/F Voltage	output	I/O Buffer	Function
81	VSS*					core GND
82	VSS*					core GND
83	VSS IO					I/O GND
84	DO[7]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
85	DO[6]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
86	DO[5]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
87	DO[4]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
88	VDE					I/O 3.3V
89	DO[3]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
90	DO[2]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
91	DO[1]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
92	DO[0]	O	3.3V	8mA	IOCB2EOT2X8NA02	Output Data
93	VSS IO					I/O GND
94	N.C					N.C
95	N.C					N.C
96	N.C					N.C
97	N.C					N.C
98	N.C					N.C
99	VDE					I/O 3.3V
100	VDD					core 1.2V
101	VSS					core GND
102	VSS IO					I/O GND
103	N.C					N.C
104	N.C					N.C
105	N.C					N.C
106	N.C					N.C
107	N.C					N.C
108	VSS IO					I/O GND
109	DI[0]	I	3.3V		IOCB2EITNNMXA02	Input Data
110	DI[1]	I	3.3V		IOCB2EITNNMXA02	Input Data
111	DI[2]	I	3.3V		IOCB2EITNNMXA02	Input Data
112	DI[3]	I	3.3V		IOCB2EITNNMXA02	Input Data
113	VDE					I/O 3.3V
114	DI[4]	I	3.3V		IOCB2EITNNMXA02	Input Data
115	DI[5]	I	3.3V		IOCB2EITNNMXA02	Input Data
116	DI[6]	I	3.3V		IOCB2EITNNMXA02	Input Data
117	DI[7]	I	3.3V		IOCB2EITNNMXA02	Input Data
118	VSS IO					I/O GND
119	VSS*					core GND
120	VSS*					core GND

•The symbol “()” in「Signal Name」is for future extension, it is the N.C pin in the cryptographic LSI.

Table 2-2 Cryptographic LSI Pin Assignment (4/4)

Pin NO	Signal Name	I/O	I/F Voltage	output	I/O Buffer	Function
121	VDD*					core 1.2V
122	VDE*					I/O 3.3V
123	DI[8]	I	3.3V		IOCB2EITNNMXA02	Input Data
124	DI[9]	I	3.3V		IOCB2EITNNMXA02	Input Data
125	DI[10]	I	3.3V		IOCB2EITNNMXA02	Input Data
126	DI[11]	I	3.3V		IOCB2EITNNMXA02	Input Data
127	VSS_IO					I/O GND
128	VDD					core 1.2V
129	DI[12]	I	3.3V		IOCB2EITNNMXA02	Input Data
130	DI[13]	I	3.3V		IOCB2EITNNMXA02	Input Data
131	DI[14]	I	3.3V		IOCB2EITNNMXA02	Input Data
132	DI[15]	I	3.3V		IOCB2EITNNMXA02	Input Data
133	VDE					I/O 3.3V
134	VSS					core GND
135	(LED[0])					N.C
136	(LED[1])					N.C
137	END_N	O	3.3V	8mA	IOCB2EOT2X8NA02	Target Process End
138	START_N	O	3.3V	8mA	IOCB2EOT2X8NA02	Target Process Start
139	VSS_IO					I/O GND
140	VDD					core 1.2V
141	VSS					core GND
142	VSS_IO					I/O GND
143	STATE[0]	O	3.3V	8mA	IOCB2EOT2X8NA02	Show Selected IP
144	STATE[1]	O	3.3V	8mA	IOCB2EOT2X8NA02	Show Selected IP
145	STATE[2]	O	3.3V	8mA	IOCB2EOT2X8NA02	Show Selected IP
146	STATE[3]	O	3.3V	8mA	IOCB2EOT2X8NA02	Show Selected IP
147	VDE					I/O 3.3V
148	VSS					core GND
149	STATE[4]	O	3.3V	8mA	IOCB2EOT2X8NA02	Show Selected IP
150	(MON[1])					N.C
151	(MON[2])					N.C
152	(MON[3])					N.C
153	VSS_IO					I/O GND
154	VDD					core 1.2V
155	EXEC	O	3.3V	8mA	IOCB2EOT2X8NA02	Target Process in process
156	N.C					N.C
157	N.C					N.C
158	N.C					N.C
159	VDE*					I/O 3.3V
160	VDD*					core 1.2V

•The symbol “()” in「Signal Name」is for future extension, it is the N.C pin in the cryptographic LSI.

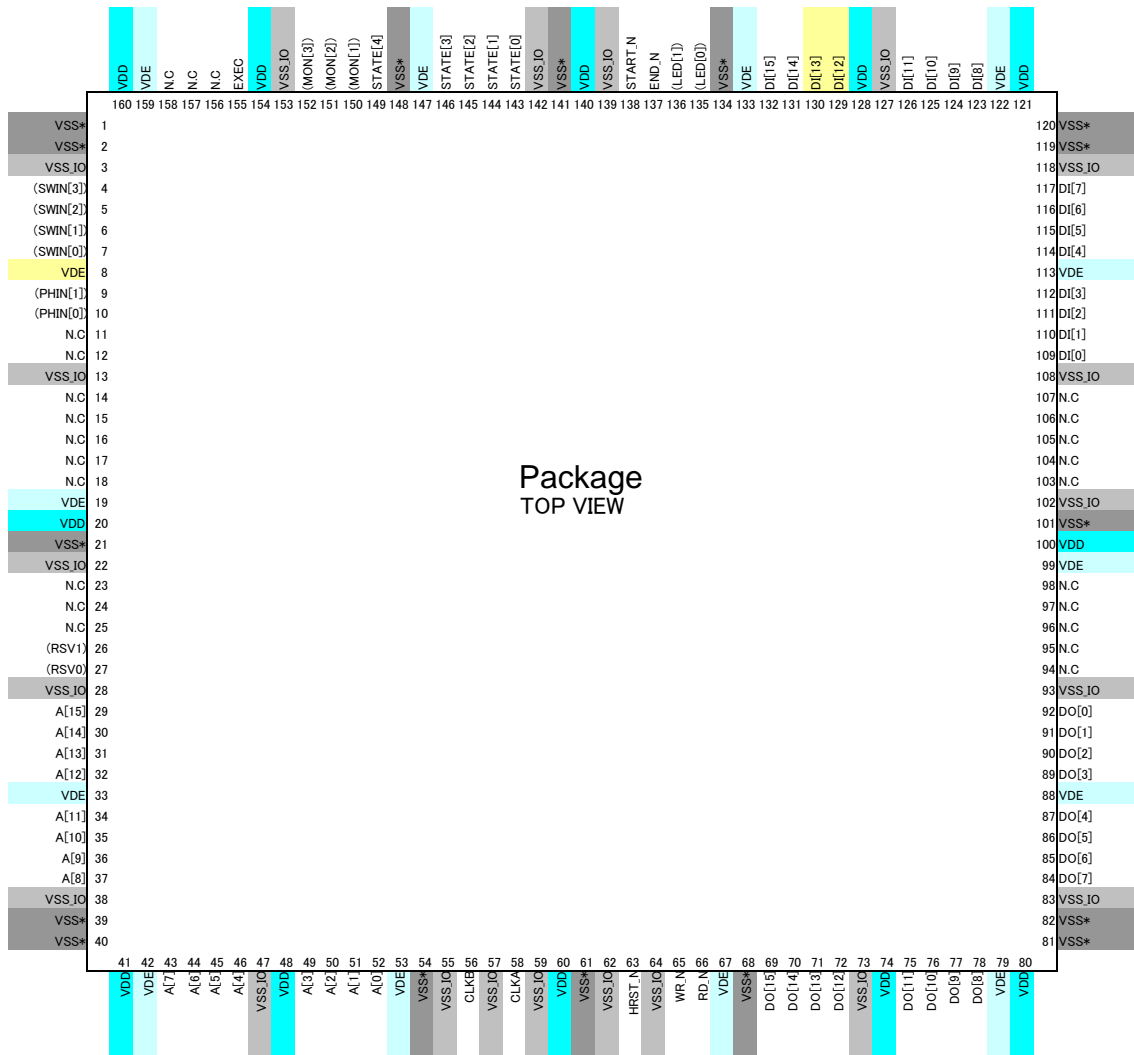


Figure 2-1 Illustration of the Pin Assignment of Cryptographic LSI

2.3. Input Output Timing

The timing of input and output of cryptographic LSI is shown in Figure 2-2 to Figure 2-4.

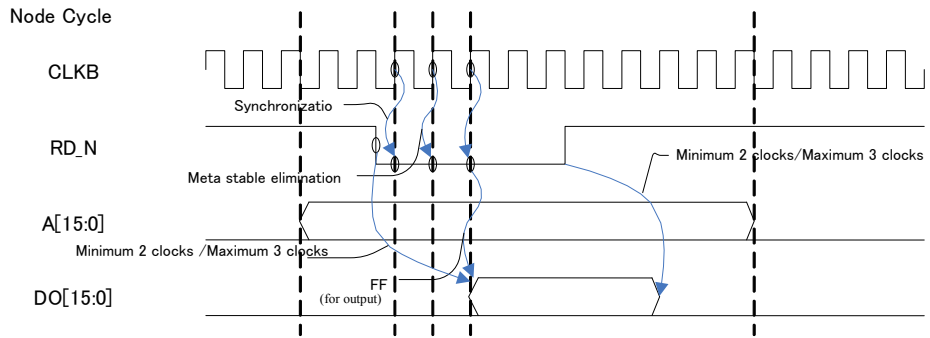


Figure 2-2 Read Cycle Timing Chart

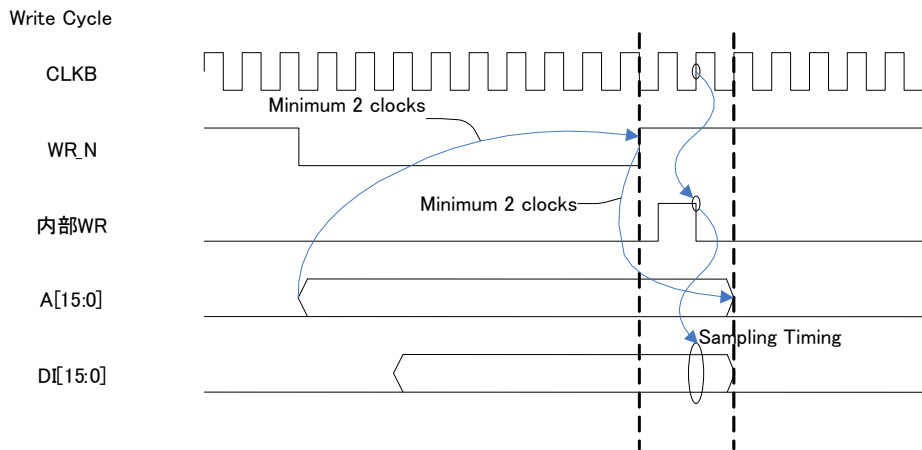


Figure 2-3 Write Cycle Timing Chart

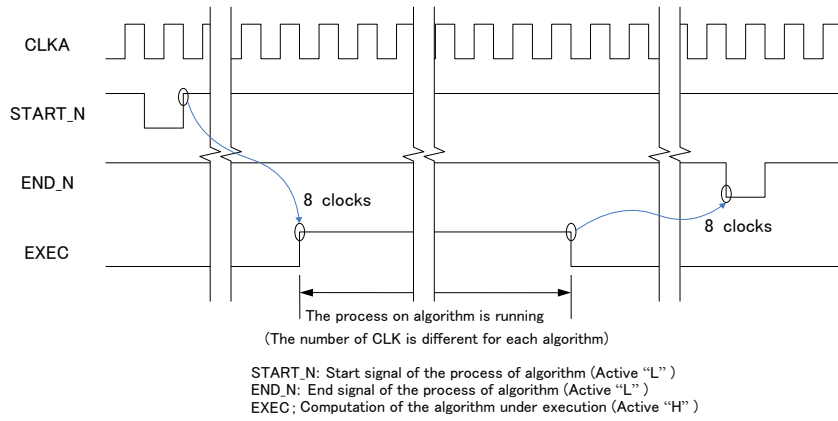


Figure 2-4 Timing Signal for Evaluation

2.4. Interface Registers

The list of interface registers and address maps of the cryptographic LSI is shown in Table 2-3.

Table 2-3 Interface Register (1/2)

Classification	Address	Register Name	Abbreviation	R/W	Functions, etc.	
System Control	0x0000	(reserved)		--		
	0x0002	Control Register	CONT	R/W	Start process instruction (W) / End process notification (R) Key generate instruction (W) / End process notification (R) Reset control of cryptographic IP (W)	
	0x0004	IP Select Register 0	IPSEL0	R/W	Specify cryptographic IP to run	
	0x0006	IP Select Register 1	IPSEL1	R/W	Specify cryptographic IP to run	
	0x0008	Output Select Register 0	OUTSEL0	R/W	Specify cryptographic IP to output data	
	0x000A	Output Select Register 1	OUTSEL1	R/W	Specify cryptographic IP to output data	
	0x000C	Mode Register	MODE	R/W	Specify running mode, key length, encryption/decryption, etc.	
	0x000E	Round Select Register	RSEL	R/W	Specify the round number to save intermediate value	
	0x0010	Test Register 1	TEST1	R	Custom core run control 1	
	0x0012	Test Register 2	TEST2	R	Custom core run control 2	
	⋮	⋮	⋮	⋮	⋮	
	0x00FF	(reserved)				
Symmetric Key Cryptography	Secret Key (→cryptographic LSI)	0x0100	Key Register 0	KEY0	W	Key for symmetric encryption (Most significant 16 bits)
		0x0102	Key Register 1	KEY1	W	Key for symmetric encryption (16 bits following KEY0)
		⋮	⋮	⋮	⋮	⋮
		0x010E	Key Register 7	KEY7	W	Key for symmetric encryption (Least significant 16 bits)
	IV (→cryptographic LSI)	0x0110	IV Data Register 0	IV0	W	Input IV data (Most significant 16 bits)
		0x0112	IV Data Register 1	IV1	W	Input IV data (16 bits following IV0)
		⋮	⋮	⋮	⋮	⋮
		0x011E	IV Data Register 7	IV7	W	Input IV data (Least significant 16 bits)
	Input Text (→cryptographic LSI)	0x0120	Input Text Register 0	ITEXT0	W	Input text data (Most significant 16 bits)
		0x0122	Input Text Register 1	ITEXT1	W	Input text data (16 bits following ITEXT0)
		⋮	⋮	⋮	⋮	⋮
		0x015E	Input Text Register 31	ITEXT31	W	Input text data (Least significant 16 bits)
	Random data (→cryptographic LSI)	0x0160	Random Data Register 0	RAND0	W	Input random data (Most significant 16 bits)
		0x0162	Random Data Register 1	RAND1	W	Input random data (16 bits following RAND0)
		⋮	⋮	⋮	⋮	⋮
		0x016E	Random Data Register 7	RAND7	W	Input random data (Least significant 16 bits)
	(reserved)	⋮	⋮	⋮	⋮	⋮
	0x017E	(reserved)				
	Output Text (←Cryptographic LSI)	0x0180	Output Text Register 0	OTEXT0	R	Output text data (Most significant 16 bits)
		0x0182	Output Text Register 1	OTEXT1	R	Output text data (16 bits following OTEXT0)
		⋮	⋮	⋮	⋮	⋮
		0x01BE	Output Text Register	OTEXT31	R	Output text data (Least significant 16 bits)

		31			bits)
Intermediate value data (←cryptographic LSI)	0x01C0	Intermediate Value Register 0	RDATA0	R	Intermediate value data (Most significant 16 bits)
	0x01C2	Intermediate Value Register 1	RDATA1	R	Intermediate value data (16 bits following RDATA0)
	:	:	:	:	:
	0x01CE	Intermediate Value Register 7	RDATA7	R	
Intermediate key data (←cryptographic LSI)	0x01D0	Intermediate Key Register 0	RKEY0	R	Intermediate key data (Most significant 16 bits)
	0x01D2	Intermediate Key Register 1	RKEY1	R	Intermediate key data (RKEY016 bits following)
	:	:	:	:	:
	0x01DE	Intermediate Key Register 7	RKEY7	R	Intermediate key data (Least significant 16 bits)
(reserved)	:	(reserved)			
	0x01FE				

Table 2-3. Interface Registers (2/2)

Classification	Addresses	Register Name	Abbreviation	R/W	Functions, etc.	
Public Key Cryptography	Exponent (←cryptographic LSI) (*1)	0x0200	Exponent Register 0	EXP00	W	Exponent(Most significant 16 bits)
		0x0202	Exponent Register 1	EXP01	W	Exponent (16 bits following EXP00)
		⋮	⋮	⋮	⋮	⋮
		0x023E	Exponent Register 31	EXP1F	W	Exponent(Least significant 16 bits)
		0x02FE	(reserved)			
	Modulus (→cryptographic LSI) (*2)	0x0300	Modulus Register 0	MOD00	W	Modulus(Most significant 16 bits)
		0x0302	Modulus Register 1	MOD01	W	Modulus(16 bits following MOD00)
		⋮	⋮	⋮	⋮	⋮
		0x033E	Modulus Register 31	MOD1F	W	Modulus(Least significant 16 bits)
		0x037E	(reserved)			
	Pre-computation result input (→cryptographic LSI) (*3)	0x0380	Pre-Computation Result Register 0	PREDAT00	W	Pre-computation result (Most significant 16 bits)
		0x0382	Pre-Computation Result Register 1	PREDAT01	W	Pre-computation result (16 bits following PREDAT00)
		⋮	⋮	⋮	⋮	⋮
		0x039E	Pre-Computation Result Register 16	PREDAT0F	W	Pre-computation result (Least significant 16 bits)
		0x03FE	(reserved)			
	Input Data (→cryptographic LSI) (*4)	0x0400	Input Data Register 0	IDATA00	W	Input Data(Most significant 16 bits)
		0x0402	Input Data Register 1	IDATA01	W	Input Data(16 bits following IDATA00)
		⋮	⋮	⋮	⋮	⋮
		0x043E	Input Data Register 31	IDATA1F	W	Input Data(Least significant 16 bits)
		0x04FE	(reserved)			
	Output Data (←cryptographic LSI) (*5)	0x0500	Output Data Register 0	ODATA00	R	Output Data(Most significant 16 bits)
		0x0502	Output Data Register 1	ODATA01	R	Output Data(16 bits following ODATA00)
		⋮	⋮	⋮	⋮	⋮
		0x053E	Output Data Register 31	ODATA1F	R	Output Data(Least significant 16 bits)
		0x05FE	(reserved)			
	(empty)	0x0600 ⋮ 0xFFFE				
	Chip Information (0xFFFF0 ~0xFFFFF)	0xFFFF0	(reserved)			
0xFFFFC		Version Register	VER	R		
0xFFFFE		(reserved)		--		

(*1) For the case of ECC, the registers for keys and random numbers are in the following range
0x0200-0x0206 Exponent Register 0-3 EXP00-03 Key Register 64bit
0x0208-0x020e Exponent Register 4-7 EXP04-07 Random Register 64bit

(*2) For the case of ECC, below becomes the x-coordinate of the ECC initial point register, and its

range is as follows

0x0300-0x0316 Modulus Register 0-11 MOD00-0B x-coordinate register 192bit

(*3) For the case of ECC, below becomes the Z-coordinate register and its range is as follows

0x0380-0x0396 Pre-Computation Result Register 0-11 PREDAT00-0B Z-coordinate register 192bit

(*4) For the case of ECC, below becomes the register for parameter b of ECC and its range is as follows

0x0400-0x0416 Input Data Register 0-11 IDATA00-0B parameter b register 192bit

(*5) For the case of ECC, the Output Data Register of ECC and its range are as follow

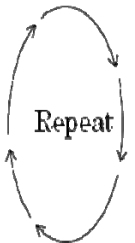
0x0500-0x0516 Output Data Register 0-11 ODATA00-0B Output Data Register 192bit

2.5. Operating Procedure

According to the interface register, the procedure of executing the process in cryptographic algorithm core is shown as follows.

(1) Cryptographic algorithm cores other than AES5 (CTR mode support pipeline implementation)

1. Operating IP selection: set the corresponding bit on the IP Selection Register (IPSEL0, 1).
2. Selected IP Reset: write 1 then 0 to CONT[IPRST]
3. Output IP selection: set the corresponding bit on the output Select Register (OUTSEL0, 1).
4. Mode setting: Set desired operating modes on the mode register (MODE). (*1)
5. Key setting:
 - 5-1 Set KEY0-7 for symmetric key ciphers, EXP00-1F and MOD00-1F for RSA, or IDATA00-03 for ECC.
 - 5-2 Set CONT[KSET], then wait until this bit is cleared.
6. Initial value (IV) setting: set IV0-7. (*2)
7. Random number (SEED) setting: set RAND0-7. (*3)
8. Cryptographic operation:
 - 8-1 Set ITEXT0-7(*4) for symmetric key ciphers, IDATA00-1F for RSA, and IDATA08-13 for ECC.
 - 8-2 Set CONT[RUN], then wait until this bit is cleared.
 - 8-3 Read OTEXT0-7(*5) for symmetric key ciphers, ODATA00-1F for RSA, and ODATA00-03 for ECC.



(*1) When selecting AES6, also set the round selection registers (KRSEL, DRSEL) as necessary.

(*2) Only for AES12 and AES13 that require initial values.

(*3) Only for AES8, AES9, and AES10 that use random numbers.

(*4) Set ITEXT0-3 for 64-bit block ciphers.

(*5) Read OTEXT0-3 for 64-bit block ciphers.

Also, when AES6 is selected, RDATA0-7/RKEY0-7 can be read (the intermediate value in the round selection register, or at fault)

Settings can be changed as follows:

- To change the cryptographic core, perform 1-8 again.
- To change the operation modes of the already selected cryptographic core, perform 4-8 again.
- To change the key of already selected cryptographic core, perform 5-8 again.
- To change the initial value of the already selected cryptographic core, perform 6-8 again.
- To change the random number of the already selected cryptographic core, perform 7-8 again.

(2) AES5 (CTR mode, pipeline implementation) core

※same with (1) until the key setting.

1. Initial value (IV) setting:

- 1-1 Set IV0-7.
- 1-2 Set 1 to control register CONT[RUN], then wait until this bit is cleared.
- 2. Random number (SEED) setting: No need to set.
- 3. Cryptographic operation:
 - 3-1 Set ITEXT0-31.
 - 3-2 Set 1 to the control register CONT[RUN], then wait until this bit is cleared.
 - 3-3 Read OTEXT0-31.



To change the initial value, perform the sequence 6-8 again.

3. Detailed Specification

Below, we explain the detailed internal specification of the cryptographic LSI. The cryptographic algorithms implemented inside the cryptographic algorithm cores in side cryptographic LSI are the ones released by AIST. In this section, we will mainly explain the overview of external parts and the interface to each cryptographic algorithm of the LSI.

3.1. Overall Block Diagram

In this section, we explain the overall structure of the cryptographic LSI. Figure 3-1 shows the overall block of the cryptographic LSI.

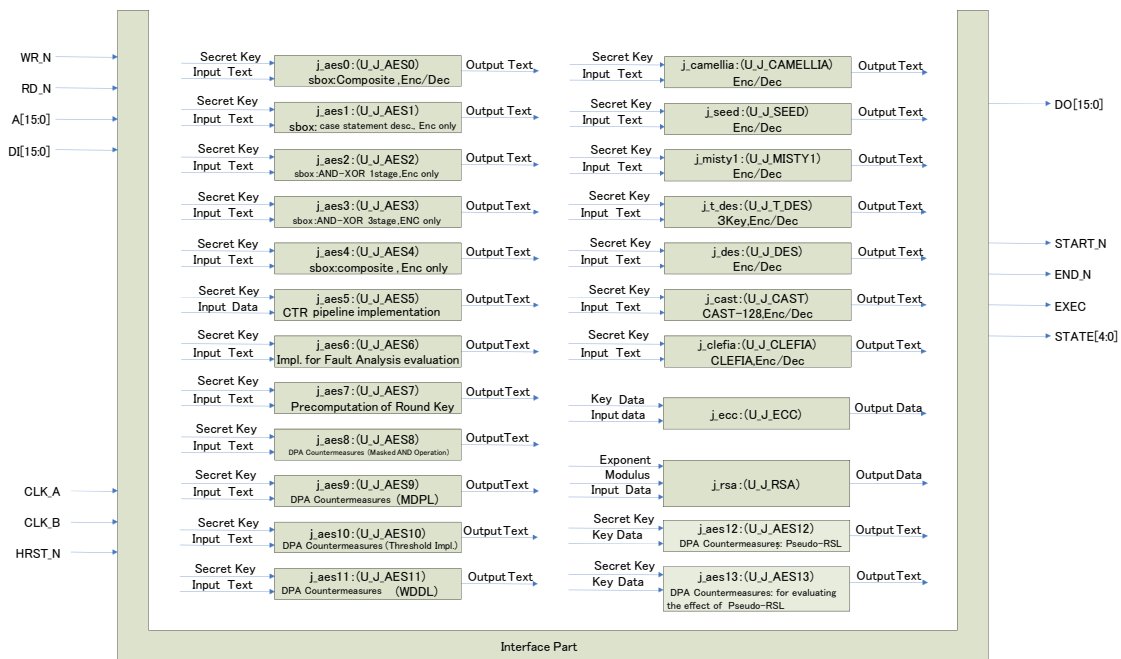


Figure 3-1 Overall Block Diagram

3.2. Hierarchical Structure

The hierarchical structure of the cryptographic LSI is shown in Figure 3-2.

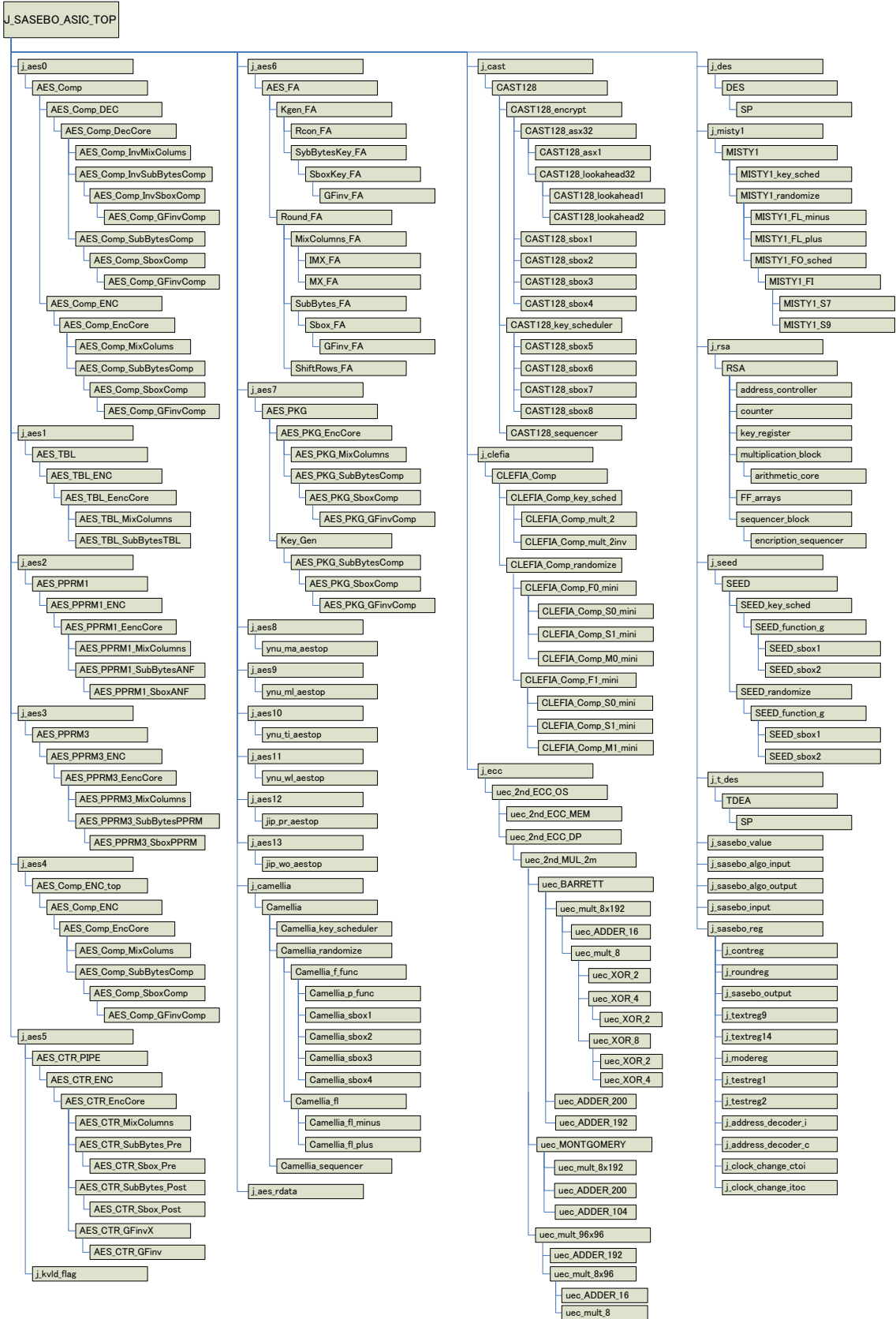
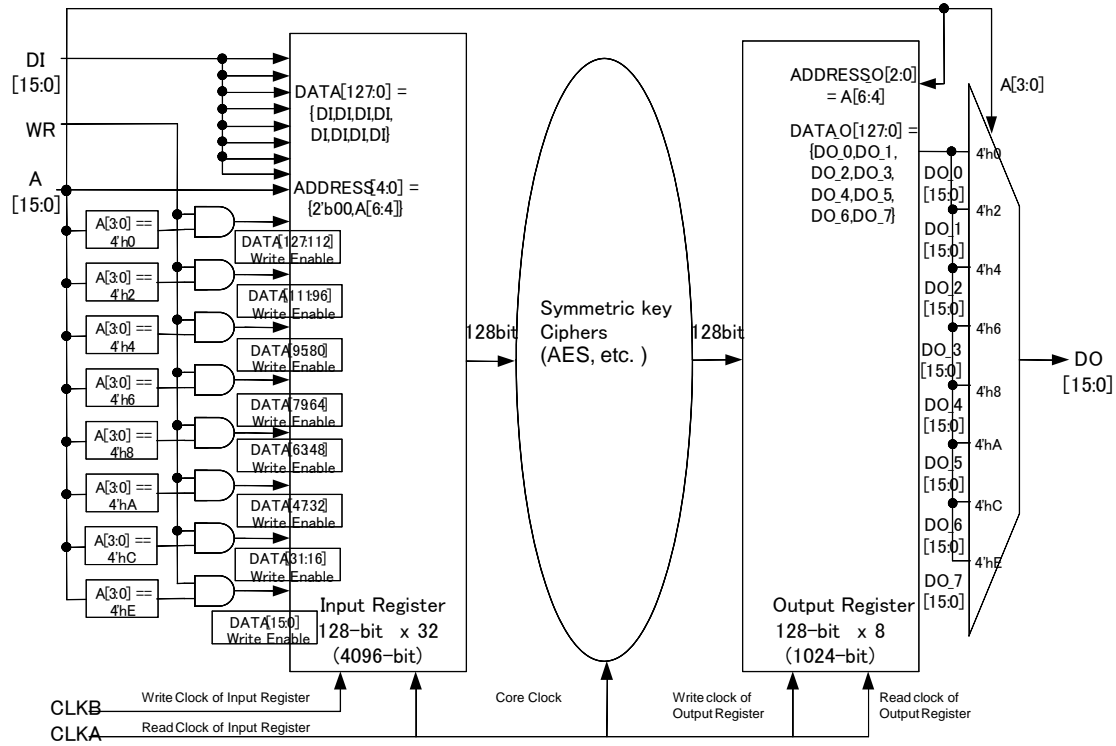


Figure 3-2 The Hierarchical Structure

3.3. External Interface Circuit

We explain the external interface circuit for symmetric key cryptographic algorithms (AES, DES, MISTY1, Camellia, SEED, CAST128, and CLEFIA) and public key cryptographic algorithms (RSA, ECC) separately. The external interface circuit for symmetric key cryptographic algorithms is shown in Figure 3-3 and that for public key cryptographic algorithms is shown in Figure 3-4.

< On Registers for Symmetric Key Cryptographic Algorithms >



[Memory Map]

① Input Register

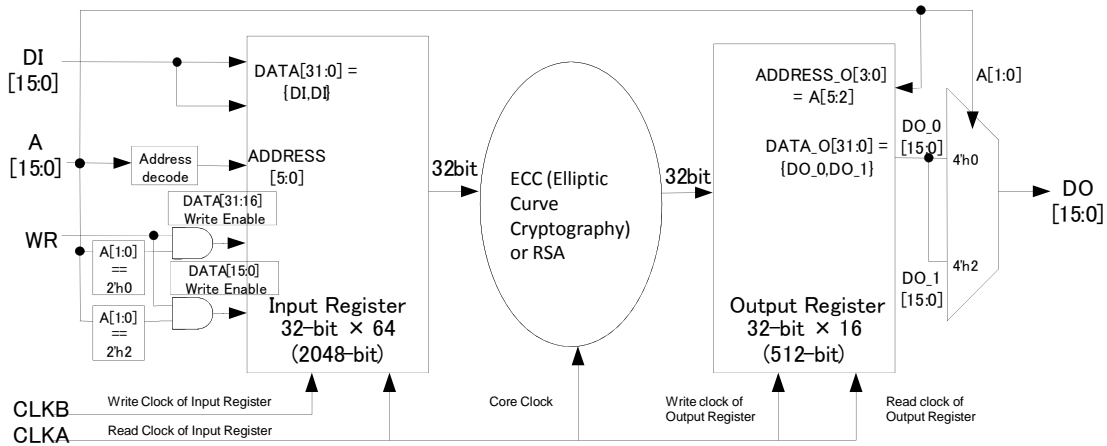
127	Secret Key	128bit	0
127	IV	128bit	0
127	Input Text		0
127	128bit x4		0
127			0
127	Random data	128bit	0
Unused 3200 bit			

② Output Register

127			0
127	Input Text		0
127	128bit x4		0
127			0
127 Intermediate Value Data 128 bit 0			
127 Intermediate Key Data 128 bit 0			
Unused 128 bit			

Figure 3-3 The interface circuit of the symmetric key cipher algorithms.

<On Registers for RSA and ECC>



[Memory Map]

① Input Register

For RSA

511	Key 512bit	0
511	Modulus 512 bit	0
255	Pre Computation's Result Input 256bit	0
511	Input Data 512bit	0
	Unused 256 bit	

For ECC

63	Key 64bit	0
63	Random Data 64 bit	0
	Unused 384bit	
191	X Coordinate Input 192 bit	0
	Unused 320bit	
191	Z Coordinate Input 192 bit	0
	Unused 64bit	
191	Parameter b 192bit	0
	Unused 576 bit	

② Output Register

For RSA

511	Output Data 512bit	0
-----	--------------------	---

For ECC

191	Output Data 192bit	0
	Unused 320bit	

Figure 3-4 The interface circuit of the asymmetric key cipher algorithms

3.4. Interfaces for Cryptographic Algorithm Cores

In this section, we briefly explain the interface to each cryptographic algorithm core, such as the key schedule and the necessary number of cycles for cryptographic process.

(0) AES0

- S-Box structure: composite
- Number of cycles for key schedule: encryption: 1[cycle]/decryption: 11[cycle]
- Number of cycles for encryption/decryption: 10[cycle/block]
- Remark: both encryption and decryption are supported.

(1) AES1

- S-Box structure: implementation using table according to the case statement
- Number of cycles for key schedule: 1[cycle]
- Number of cycles for encryption: 10[cycle/block]
- Remark: only encryption is supported

(2) AES2

- S-Box structure: AND-XOR structure with 1 stage description

- Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption: 10[cycle/block]
 - Remark: only encryption is supported.
- (3) AES3
- S-Box structure: AND-XOR structure with 3 stages description
 - Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption: 10[cycle/block]
 - Remark: only encryption is supported.
- (4) AES4
- S-Box structure: composite
 - Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption: 10[cycle/block]
 - Remark: only encryption is supported.
- (5) AES5
- S-Box structure: composite
 - Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption: 46[cycle/4block]
 - Remark: 1 round/4 stages inner pipeline with CTR mode circuit.
- (6) AES6
- S-Box structure: composite
 - Number of cycles for key schedule: 20[cycle]/21[cycle](encryption/decryption)
 - Number of cycles for encryption: 21[cycle/block]
 - Remark: AES implementation with countermeasures against fault injection attack. Both encryption and decryption are supported.
- (7) AES7
- S-BOX structure: composite
 - Number of cycles for key schedule: 11[cycle]
 - Number of cycles for encryption/decryption: 10[cycle/block]
 - Remark: AES implementation with precomputation of round keys. Only encryption is supported.
- (8) AES8
- Implementation for evaluating countermeasures against DPA (Masked AND Operation)
- (9) AES9
- Implementation for evaluating countermeasures against DPA (MDPL)
- (10) AES10
- Implementation for evaluating countermeasures against DPA (WDDL)
- (11) AES11
- Implementation for evaluating countermeasures against DPA (Masked AND Operation)
- (12) AES12
- Implementation for evaluating countermeasures against DPA (Pseudo RSL)
- (13) AES13
- Implementation for evaluating countermeasures against DPA (For evaluating the effect of Pseudo RSL)
- (14) Camellia
- Number of cycles for key schedule: 6[cycle]
 - Number of cycles for encryption/decryption: 23[cycle/block]
 - Remark: Both encryption and decryption are supported.

- (15) SEED
- Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption/decryption: 16[cycle/block]
 - Remark: Both encryption and decryption are supported.
- (16) MISTY1
- Number of cycles for key schedule: 8[cycle]
 - Number of cycles for encryption/decryption: 9[cycle/block]
 - Remark: Both encryption and decryption are supported.
- (17) Triple-DES
- Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption/decryption: 48[cycle/block]
 - Remark: Both encryption and decryption are supported.
- (18) DES
- Number of cycles for key schedule: 1[cycle]
 - Number of cycles for encryption/decryption: 16[cycle/block]
 - Remark: Both encryption and decryption are supported.
- (19) CAST128
- Number of cycles for key schedule: 128[cycle]
 - Number of cycles for encryption/decryption: 16[cycle/block]
 - Remark: Both encryption and decryption are supported.
- (20) RSA
- The modulo exponentiation process of RSA can be run.
 - 6 types of module exponentiation algorithms (left binary method, left binary method with countermeasures, right binary method, right binary method with countermeasure, Montgomery Powering Ladder, right binary method of M. Joye) and CRT computation are supported.
 - As countermeasures against side channel attacks; "square-and-multiply always method" (countermeasure method using dummy computation), Montgomery Powering Ladder and right binary method proposed by M. Joye are supported.
 - For square modulo computation, Montgomery High Radix Square Algorithm is used.
- (21) ECC
- Scalar point multiplication for field with characteristic 2 and key up to 64 bit are supported.
 - For scalar point multiplication, Lopez-Dahab algorithm is used.
- (22) CLEFIA
- Number of cycles for key schedule: 13[cycle]
 - Number of cycles for encryption: 18[cycle/block]
 - Number of cycles for decryption: 19[cycle/block]
 - Remark: Both encryption and decryption are supported.

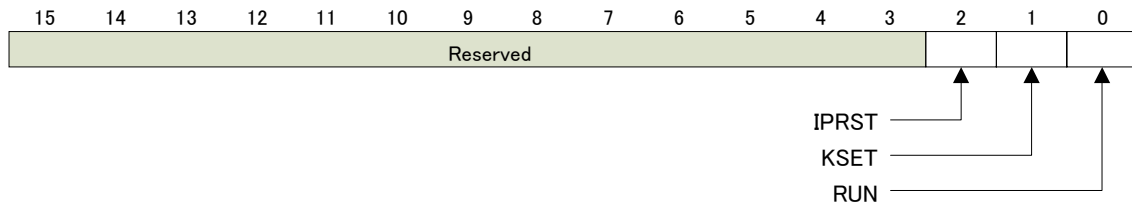
3.5. Detail of Interface Registers

In this section, we will explain the detail of interface registers.

3.5.1. Set of Registers for System Control

1) Control Register: CONT

This register is relevant to initiation and termination of cryptographic operation.



Bit 0: RUN

Write '1' to this bit, and the cryptographic IP designated by the IP Select Register (IPSEL) starts to operate. For the internal process, the information on the RUN bit initially captured by the interface clock CLKB will be synchronized with the internal clock CLKA. 16 CLKA cycles after that, the IP operation actually begins. When the cryptographic IP designated by the Output Select Register (OUTSEL) completes its operation and the Output Text/Data Registers (OTEXT/ODATA) become ready to read, this bit will be automatically cleared to '0'. When this bit is '1', writing to any registers is prohibited, and the read value on an Output Text/Data Register is not valid.

Bit 1: KSET

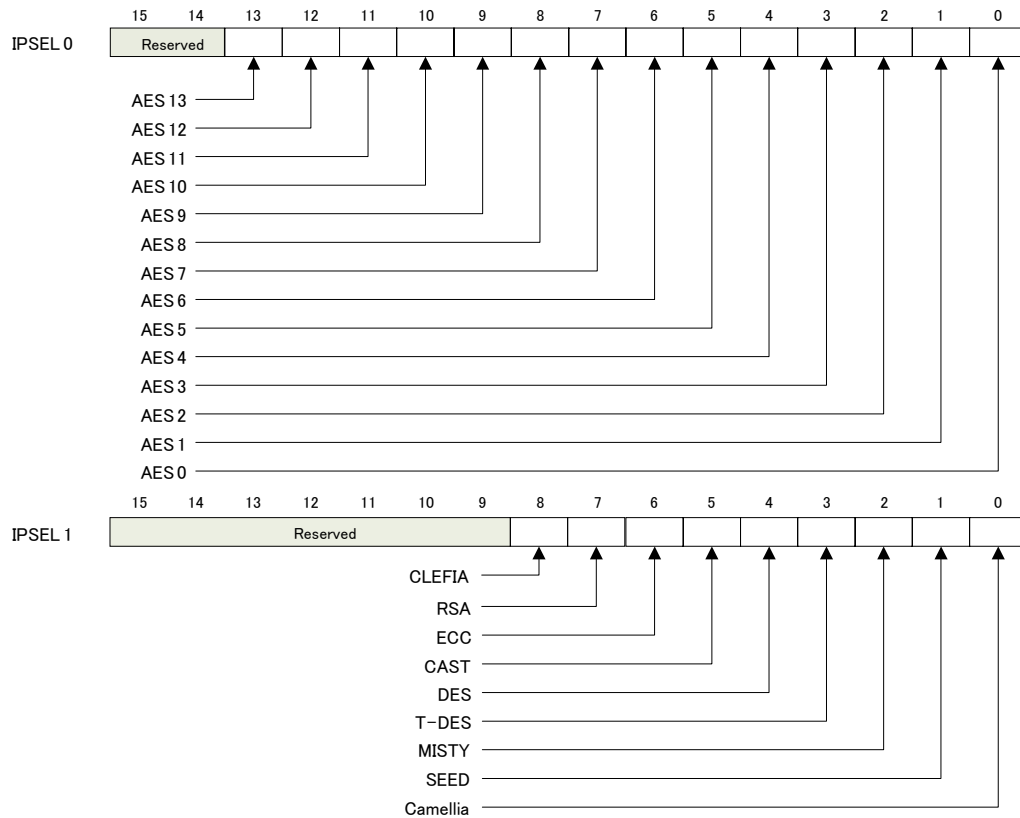
Write '1' to this bit, and a key is generated, in accordance with the mode register (MODE), within the cryptographic IP designated by the IP Selection Register (IPSEL). When the key generation in the cryptographic IP designated by the Output Select Register (OUTSEL) is completed and a cryptographic operation with the generated key becomes ready to start, this bit will be automatically cleared to '0'. When this bit is '1', writing to any registers is prohibited. If the KSET bit is '1' and the RUN bit is set, there is no guarantee of proper operation.

Bit 2: IPRST

Write '1' to this bit, and the cryptographic IP designated by the IP Selection Register (IPSEL) is reset. Write '0' to this bit, and the reset state of the IP is released. The initial value of the bit is '1'

2) IP Select Register : IPSEL0, 1

Among cryptographic IPs in ASIC, the IPs whose corresponding bits are set to '1' enter the active state. The other unselected IPs are not provided with the clock signal.

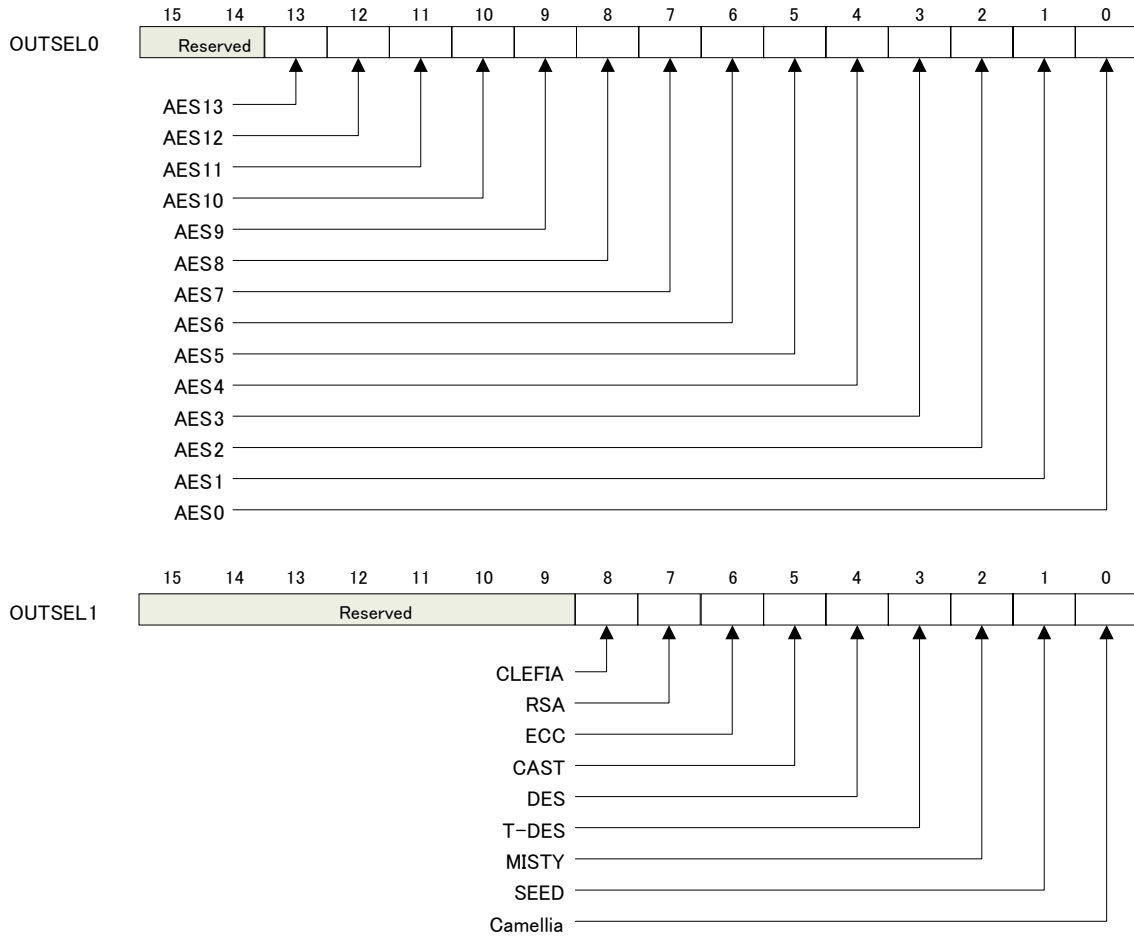


Bit	Cryptographic IP	Remark
AES0	S-Box Implementation→Composite, Encryption/Decryption supported	AES_Comp
AES1	S-Box Implementation→case statement, only encryption is supported	AES_TBL
AES2	S-Box Implementation → AND-XOR implementation (1-Stage), only encryption is supported	AES_PPRM1
AES3	S-Box Implementation → AND-XOR implementation (3-Stage), only encryption is supported	AES_PPRM3
AES4	S-Box Implementation→Composite, only encryption is supported	AES_Comp_ENC_top
AES5	CTR mode supported, pipelined implementation	AES_CTR_PIPE
AES6	Implementation for evaluation of countermeasures against fault injection attacks	AES_FA
AES7	Implementation for precomputation of round keys	AES_PKG
AES8	Implementation for evaluation of countermeasures against DPA (Masked AND Operation)	
AES9	Implementation for evaluation of countermeasures against DPA (MDPL)	
AES10	Implementation for evaluation of countermeasures against DPA (Threshold Implementation)	
AES11	Implementation for evaluation of countermeasures against DPA (WDDL)	
AES12	Implementation for evaluation of countermeasures against DPA (Pseudo RSL)	
AES13	Implementation for evaluation of countermeasures against DPA (For evaluation of the effect of pseudo RSL)	
Camellia	Key size: 128 bit, both encryption and decryption are supported	
SEED	SEED: both encryption and decryption are supported	
MISTY	MISTY1: both encryption and decryption are supported	
T DES	Triple-DES: 3Key, both encryption and decryption are supported	
DES	Both encryption and decryption are supported	
CAST	CAST128: both encryption and decryption are supported	
ECC	Key size: 64bit. Point scalar multiplication on field with characteristic 2 is	

	supported	
RSA	RSA: 1024bit modulo exponentiation	
CLEFIA	CLEFIA: both encryption and decryption are supported	

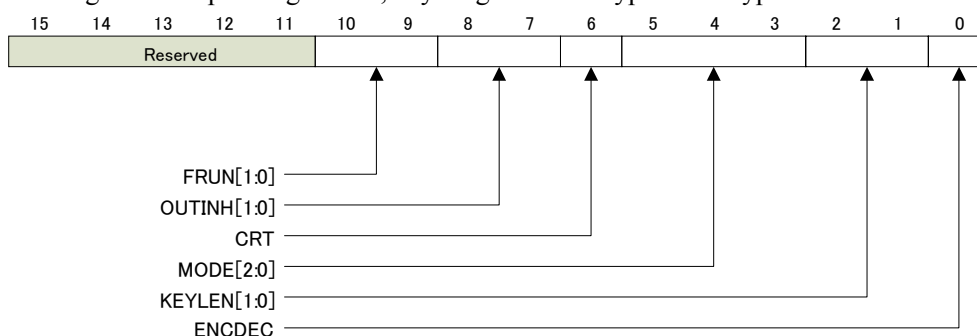
3) Output Select Register : OUTSEL0, 1

Write '1' to one of the bits corresponding to the IP Select Register (IPSEL)'s bits with '1' representing active IPs, and the corresponding cryptographic IP is designated to export the operation result. The operation result of the designated IP will be stored in the Output Text/Data Registers (OTEXT/ODATA). The output value will not be defined if two or more bits of the Output Select Register are set to '1'.



4) Mode Register: MODE

Designate the operating modes, key length and encryption/decryption.



Bit 10-9: FRUN

Controls the free-run mode where the operation run repeats every 0.3 seconds, supported by AES0 only.

- | | | |
|-----------|---|--|
| FRUN[1] : | 0 | FRUN mode OFF |
| | 1 | FRUN mode ON |
| FRUN[0] : | 0 | Adopts ITEXT as the initial input and increments it at every run. |
| | 1 | Adopts ITEXT as the initial input and assigns the operation result to the next input at every run. |

Bit 8-7: OUTINH

Controls the output enable for the control signals.

- | | | |
|-------------|---|--|
| OUTINH[1] : | 0 | Control signal output inhibition OFF (The control signals will be enabled to be exported.) |
| | 1 | Control signal output inhibition ON (The function is further specified in OUTINH[0]) |
| OUTINH[0] : | 0 | Output of all the control signals is disabled. |
| | 1 | Output of all the control signals except START is disabled. |

Bit 6: CRT

This bit directly connects with the CRT port of the RSA core. This bit has no effect on the other cores. Current LSIs have a bug in CRT mode.

- 0: CRT mode OFF
- 1: CRT mode ON

Bit 5-3: MODE[2:0]

For RSA, these bits directly connect with the MODE input of the RSA core to specify the following operating modes:

- 000: Left binary method
- 001: Right binary method
- 010: Left binary method with a countermeasure
- 011: Right binary method with a countermeasure
- 100: Montgomery powering ladder
- 101: M. Joye's right binary method

For ECC, the 3-bit operating mode control port of the ECC core is not connected to these bits, but is fixed to 3'b000 in the interface circuit.

For IPs other than AES12 and RSA, the modes of operation (e.g. ECB and CTR) or operating modes are fixed with the specific values depending on the IP. For AES12, the Test Register TEST2 controls the operating mode.

Bit 2-1: KEYLEN[1:0]

The value is fixed to 00. The actual key length is already specified for each IP.

Bit 0: ENCDEC

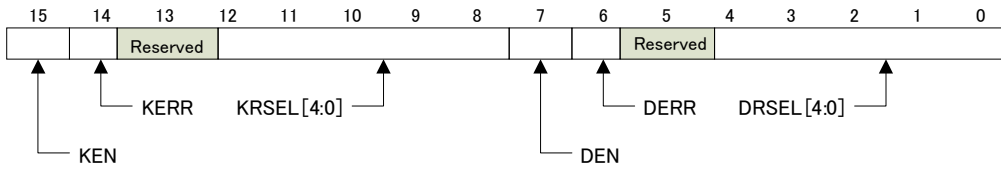
Specifies encryption by 0 or decryption by 1.

(For the IPs that only support encryption, this bit is not valid.)

5) Round Select Register : RSEL

Designate the rounds whose value and key will be stored in Intermediate Value Register (RDATA0-RDATA7) and Intermediate Key Register (RKEY0-RKEY7).

KRSEL/RDATA0-RDATA7 and DRSEL/RKEY0-RKEY7 are valid only when AES6 is selected.



Bit 15: KEN

- 0: Deactivates the intermediate key register by not supplying the clock signal.
- 1: Activates the intermediate key register by supplying the clock signal.

Bit 14: KERR

Represents the key error status. (Directly connects with Err[0] of AES_FA.)

- 0: normal operation
- 1: error occurs

Bit 12-8: KRSEL[4:0]

Designates the round number where the intermediate key register (RKEY0-RKEY7) takes the intermediate key.

Bit 7: DEN

- 0: Deactivates the intermediate data register by not supplying the clock signal. The register continues to store the intermediate data, but cannot latch new intermediate data.
- 1: Activates the intermediate data register by supplying the clock signal.

Bit 6: DERR

Represents the data error status. (Directly connects with Err[1] of AES_FA.)

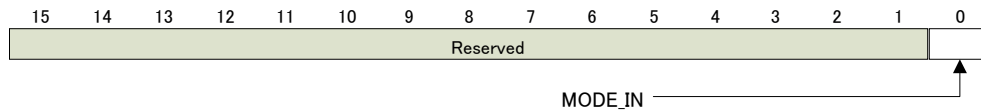
- 0: normal operation
- 1: error occurs

Bit 4-0: DRSEL[4:0]

Designates the round number where the intermediate data register (RDATA0-RDATA7) takes the intermediate data.

6) Test Register1: TEST1

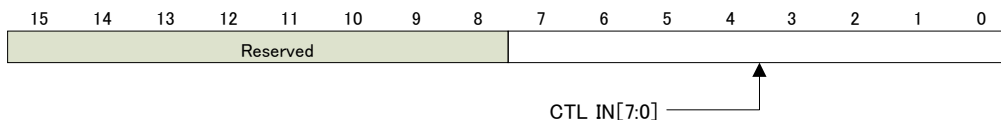
Write '0x0001' to the Test Register TEST1, and encryption will use an internal key instead of externally applied keys. Once TEST1 is set, either cycling power or asserting the hardware reset HRST_N is required to resume the normal cryptographic operation (which uses externally applied keys). This mode is supported in all AES cryptographic IP cores (AES0-13).



Bit 0: MODE_IN

7) Test Register2: TEST2

The Test Register TEST2 is the debugging register for controlling AES12. The test register functionality is not available. CTL_IN[7: 0] is directly connected to ctl_in[7:0] of AES12. For the detailed specification on ctl_in signal, refer to "The External Specification of Cryptographic IP for Evaluation of Tamper Resistance for AIST".



Bit 7-0: CTL_IN[7: 0]
 Controls ctl_in[7:0] of AES12 core

3.5.2. Set of Registers for Symmetric Key Cryptography

1) Key Register : KEY0-7

The Key Register (KEY0-7) has a capacity of 128 bits, made up of 8 16-bit subregisters. However, due to the export control regulation, only the 56 bits consisting of the lower 8 bits of KEY4 and whole KEY5~7 are usable in practice. For each cryptographic core, the key will be used as follows:

DES: Provide the lower 8 bits of KEY4 and whole KEY5~7 with the original 56-bit key excluding parity bits. The DES cryptographic core adds parity bits to the 56-bit key making 64 bits in the circuit.

TDES: Provide the lower 8 bits of KEY4 and whole KEY5~7 with the original 56-bit key excluding parity bits. The T-DES cryptographic core takes the full-length key as follows:

[191:64]: 0x000102030405060708090a0b0c0d0e0f (Fixed value)

[63:0]: The user-provided 56-bit key with added parity bits.

Others: The 128 bits of the key are fixed as shown below.

[127:56]: 0x000102030405060708

[55:0]: The user-provided 56-bit key.

(The rest 56 bits are set in the lower 8 bits of KEY4 and whole KEY5-7 by the user.)

127 (MSB)								(LSB)0
	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6	KEY7

2) IV Data Register: IV0-7

- Used as IV for AES5(AES_CTR_PIPE), AES12(JIP_PR_AESTOP), and AES13(JIP_WO_AESTOP).

- If some data are inputted from outside of ASIC circuit to IV Data Register (even only once) in the next encryption process, IV will be updated.

127 (MSB)								(LSB)0
	IV0	IV1	IV2	IV3	IV4	IV5	IV6	IV7

3) Input Text Register: ITEXT0-31

The Input Text Register holds the Input Text Data used by the IP that the IP Select Register IPSEL designates. Note that each cryptographic core has different sizes of Input Data and different positions of Input Text Register as follows:

AES5 (AES_CTR_PIPE): Takes 128 bits x 4 blocks.

ITEXT0-7	First block of 128 bits
ITEXT8-15	Second block of 128 bits
ITEXT16-23	Third block of 128 bits
ITEXT24-31	Forth block of 128 bits

64-bit block ciphers (MISTY1, TDES, DES, CAST128)

ITEXT0-3	64-bit input
ITEXT4-31	Unused

128-bit block ciphers

ITEXT0-7	128-bit input
ITEXT8-31	Unused

4) Random Number Data Register

- Used as SEED in AES8 (U_YNU_MA_AESTOP), AES9(U_YNU_ML_AESTOP), AES10 (U_YNU_TI_AESTOP).
- Once a seed is set to the Random Number Data Register, the random number will be updated in the next encryption process.
- The random number for AES9(U_YNU_ML_AESTOP) is 32bit. It is inputted at the most significant bits of Random Number Data Register (RAND0-1).

127(MSB)				(LSB)0			
RAND 0	RAND 1	RAND 2	RAND 3	RAND 4	RAND 5	RAND 6	RAND 7

5) Output Text Register: OTEXT0-31

The Output Text Register holds the Output text data exported by the IP that the Output Select Register OUTSEL designates. Note that each cryptographic core has different sizes of Output Data and different positions of Output Text Register as follows:

AES5 (AES_CTR_PIPE) :	Exports 128 bits x 4 blocks.
	OTEXT0-7 First block of 128 bits
	OTEXT8-15 Second block of 128 bits
	OTEXT16-23 Third block of 128 bits
	OTEXT24-31 Forth block of 128 bits
64-bit block ciphers (MISTY1, TDES, DES, CAST128)	
	OTEXT0-3 64-bit output
	OTEXT4-7 0x0000000000000000
	OTEXT8-31 Don't care
128-bit block ciphers	
	OTEXT0-7 128-bit output
	OTEXT8-31 Don't care

(Attention)

- Different from other cryptographic cores, consecutive data input (128bit x 4 blocks) and consecutive data output (128bit x 4 blocks) are performed.
- Once some data are inputted to IV Data Register from the outside of ASIC circuit, in the next encryption process, the IV setting operation is executed, only the counter mode of random number generating process is performed. Therefore, there is no input on Output Text Register (OTEXT). Again, in the next encryption process, the encryption process using input from Input Text Register (ITEXT) is executed, and after that, in the next process, the operation on the part of random number generation being used is performed.

127(MSB)				(LSB)0			
OTEXT 0	OTEXT 1	OTEXT 2	OTEXT 3	OTEXT 4	OTEXT 5	OTEXT 6	OTEXT 7
OTEXT 8	OTEXT 9	OTEXT 10	OTEXT 11	OTEXT 12	OTEXT 13	OTEXT 14	OTEXT 15
OTEXT 16	OTEXT 17	OTEXT 18	OTEXT 19	OTEXT 20	OTEXT 21	OTEXT 22	OTEXT 23
OTEXT 24	OTEXT 25	OTEXT 26	OTEXT 27	OTEXT 28	OTEXT 29	OTEXT 30	OTEXT 31

6) Intermediate Value Register: RDATA0-7

A set of these register fractions reads an Intermediate value data at the designated round for AES6. The register is valid when the IP Select Registers IPSEL/OUTSEL designate AES6 and the DEN bit of the Round Select Register RSEL is '1'. The intermediate data register holds the data value in the following two cases:

1. Designating the round at which the register holds the intermediate data.
The Round Select Register RSEL[DRSEL] specifies the round. The data aligns in RDATA0 toward RDATA7 starting with the upper 16 bits.
2. When a fault error occurs.
A fault error asserts the Err[0] signal in the AES_FA module of AES6. It also raises the DERR bit of the Round Select Register RSEL and captures the intermediate data.

(Regardless of whether the intermediate data value exporting function is valid or not, the encryption or decryption operation continues, and the output text register OTEXT holds the final result.)

127(MSB)								(LSB)0
RTEXT 0	RTEXT 1	RTEXT 2	RTEXT 3	RTEXT 4	RTEXT 5	RTEXT 6	RTEXT 7	

7) Intermediate Key Register: RKEY0-7

A set of these register fractions reads the Intermediate key data at the designated round for AES6. The register is valid when the IP Select Registers IPSEL/OUTSEL designate AES6 and the KEN bit of the Round Select Register RSEL is '1'. Regardless of whether the intermediate key value exporting function is valid or not, the encryption or decryption operation continues, and the Output Text Register OTEXT holds the final result. The intermediate key register holds the key value in the following two cases:

1. Designating the round at which the register holds the intermediate key.
The Round Select Register RSEL[KRSEL] specifies the round. The data aligns in RKEY0 toward RKEY7 starting with the upper 16 bits.
2. When a fault error occurs.
A fault error asserts the Err[1] signal in the AES_FA module of AES6. It also raises the KERR bit of the Round Select Register RSEL and captures the intermediate key.

127(MSB)								(LSB)0
RKEY 0	RKEY 1	RKEY 2	RKEY 3	RKEY 4	RKEY 5	RKEY 6	RKEY 7	

3.5.3. Set of Registers for Public Key Cryptography

1) Exponent Register : EXP00-EXP1F

RSA: Input of 512 bit exponent data. EXP00 is associated with the upper 16 bits, EXP1 has the next 16 bits, and the rest follows accordingly.

ECC: The data are inputted as follows.

EXP00-EXP03: 64 bit secret key

EXP04-EXP07: 64bit random number

EXP08-EXP1F: unused

Putting upper bits of data on register with smaller index.

2) Modulus Register : MOD00-MOD1F

RSA: Input of 512 bit modulus data. MOD00 is associated with the upper 16 bits, MOD1 has the next 16 bits, and the rest follows accordingly.

ECC: The data are inputted as follows.

MOD00-MOD0B: x-coordinate data of the inputted point in Affine coordinate (192 bits)

MOD0C-MOD1F: Unused

Putting upper bits of data on register with smaller index.

- 3) Pre-Computation Result Register : PREDAT00-PREDAT0F
 RSA: Input the 25bit of Pre-computation result in CRT mode. In PREDAT0, the Most significant 16 bits of Pre-computation result is stored, and the rest follows accordingly, e.g., PREDAT01, PREDAT02, ...
 ECC: The following data are inputted.
 PREDAT00-PREDAT0B: Z coordinate of the inputted point in projective coordinate (192 bits)
 PREDAT0C-PREDAT0F: unused
 Putting upper bits of data on register with smaller index.
- 4) Input Data Register : IDATA00-1F
 RSA: 512 bit Input Data Register. In IDATA0, Most significant 16 bits of Input Data is stored, following by IDATA01, IDATA02, ...
 ECC: The data below are inputted.
 IDATA00-IDATA0B: elliptic curve's parameter b(192 bit)
 IDATA0C-IDATA1F: unused
 Putting upper bits of data on register with smaller index.

3.5.4. Set of Registers for Chip Information

- 1) Version Register: VER
 A special register which reads the chip version. 0xE27C will be read out.

3.6. Clock Tree

In cryptographic LSI, the clock signal is only provided to the target core according to the setting of interface registers. To ease the fault analysis, the core clock is isolated from the interface circuit clock. This allows noise to be added on the core clock without affecting the interface circuit clock. The clock tree of the LSI is shown in Figure 3-5.

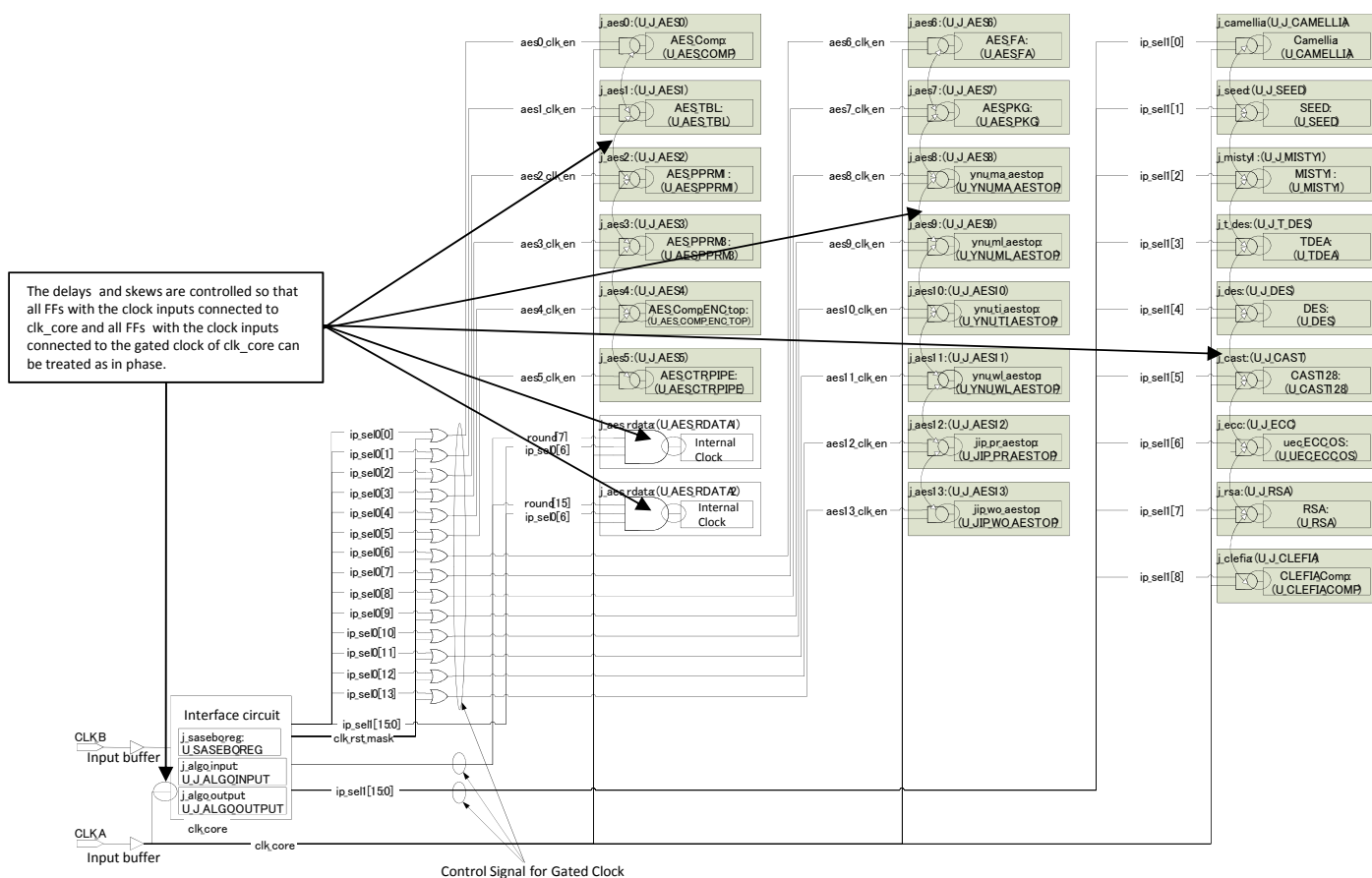


Figure 3-5 The clock tree of the LSI.

3.7. Reset

Figure 3-6 provides an overview of the reset system of the LSI. The reset sequence is explained below.

1. HRST_N assertion/deassertion
Assert the HRST_N signal to reset the interface circuit. This assertion also turns the IPRST bit of the control registers CONT of the interface circuit to '1', and activates every IP's reset signal. Then deassert the HRST_N to bring the cryptographic LSI to the initial state.
2. Feeding CLK_A and CLK_B (activation)
Activate these clock signals to make the interface circuit operable. At this point, still no active clock signals connect to any of the cryptographic cores and the reset signal is kept asserted.
3. IP core selection
Select the IP to operate by setting '1' to the corresponding bit of the IP selection register IPSEL of the interface circuit. It initiates supplying the selected core with the clock. At this point, the reset signals connected to every IP including the selected IP are kept asserted.
4. Release of the selected core's reset
Deassert and release the reset signal of the IP selected in the sequence 3, by writing '0' to the IPRST bit of the control register CONT of the interface circuit. Although not a part of the reset sequence, as the operation procedure of cryptographic LSI the Output Select Registers: OUTSEL0, 1 subsequently must be properly set.

Note that the IP cores that are not activated by the IP selection register do not have an active clock

signal. The reset signal is kept asserted to those IPs

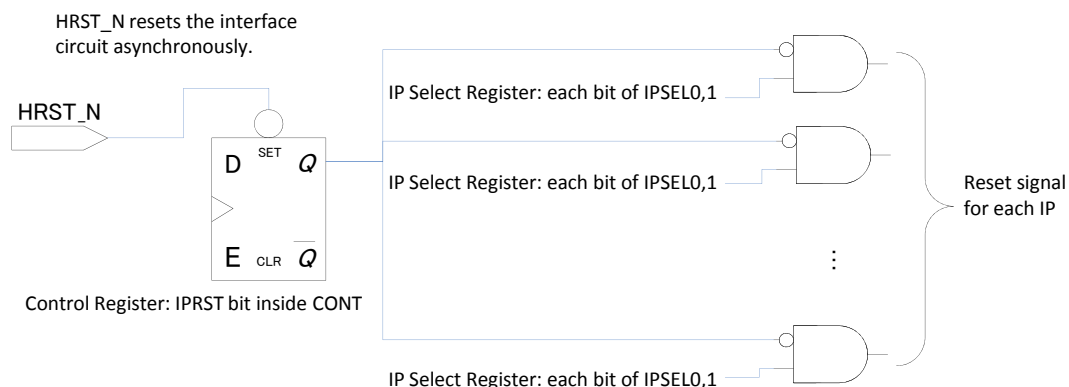


Figure 3-6 The reset system of the LSI.

3.8. Supplementary Functions and Considerations

In this section, we explain several supplementary functions which have not been described up to this point and also some considerations.

3.8.1. Core Clock and Interface Clock

The LSI has two separated clock signals as the core clock CLK_A and the interface clock CLK_B to ease fault analysis by injecting a clock based fault into only the core. For simplification, as the assumption for the design, the frequency of interface clock is set to be lower than the frequency of core clock. In short, the clock provided to cryptographic LSI needs the following to hold.

$$\text{Frequency of CLK}_A > \text{frequency of CLK}_B$$

(If CLK_A is 24MHz, even CLK_B on the level of 23MHz is allowed. If it becomes complicated on the board, simply setting CLK_A to be twice of CLK_B, i.e., CLK_B=12MHz, is allowed.)

3.8.2. Key Length Limit

In the design data (RTL) of cryptographic LSI, to meet the export control regulations the description on the limitation of key length of algorithm cores is added. For this, the key length for symmetric key algorithm is limited to 56 bit. (Length for RSA and ECC are limited to 512 bits and 64 bits, respectively.) The treatment for each symmetric key algorithm is as follows.

- (1) DES: set the 56 bit key without parity to the lower 8 bits of KEY4 and KEY5-7. The key data bit assignment of DES in cryptographic LSI is shown in Figure 3-7 and Figure 3-8.

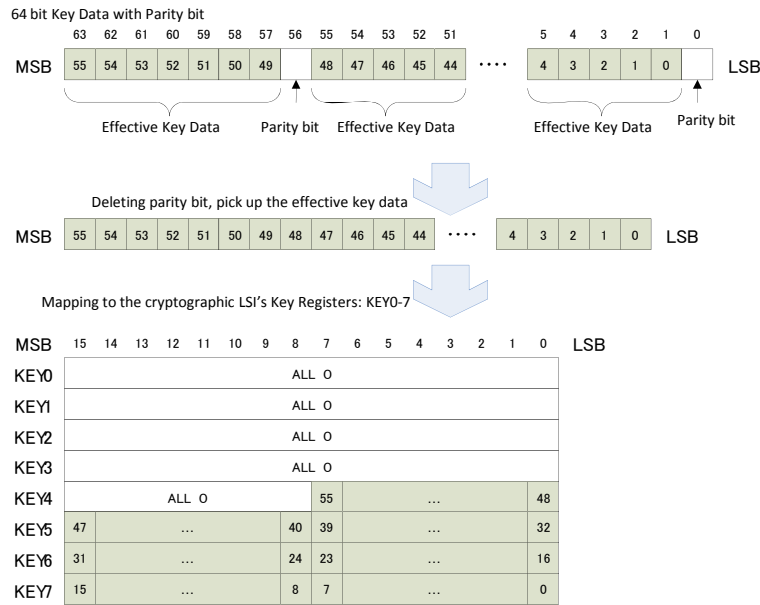


Figure 3-7 Key Data Bit Assignment of the DES

(2) T-DES: the fixed value shown below goes to the upper [191:64] of the key. The lower [63: 0] is the same as for DES.

[191:64]: 0x000102030405060708090a0b0c0d0e0f (fixed value)

(3) For other symmetric key ciphers: for cores with 128 bits key length, the setting is as shown below.

[127: 56]: 0x000102030405060708 (fixed value)

[55:0]: 56 bits key inputted from outside: lower 8 bits of KEY4 and KEY5-7

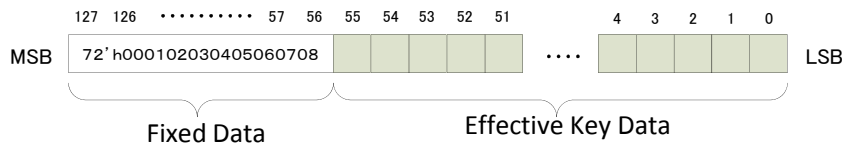


Figure 3-8 Key Data Bit

3.8.3. Delayed Operation

To obtain power traces precisely during cryptographic operations in the cryptographic LSI, the timings for the power changes caused by the key settings and data input/output is separated from the timings for the power of real cryptographic process.

Concretely, if the RUN bit of Control Register: CONT is turned on to instruct the process initiation, after 8 CLK, the evaluation signals (START_N, EXEC) are asserted, and after another 8 CLK the signal indicating the start of operation on cryptographic algorithm core is asserted. (CLK is equivalent to CLK_A). At the termination of the algorithm core operation, after the signal indicating the termination of computation, after 8 CLK, the evaluation signals END_N is asserted and EXEC is deasserted. And after another 8 CLK, the Control Register CONT[RUN] is set to 0. See the detail flow at Figure 3-9.

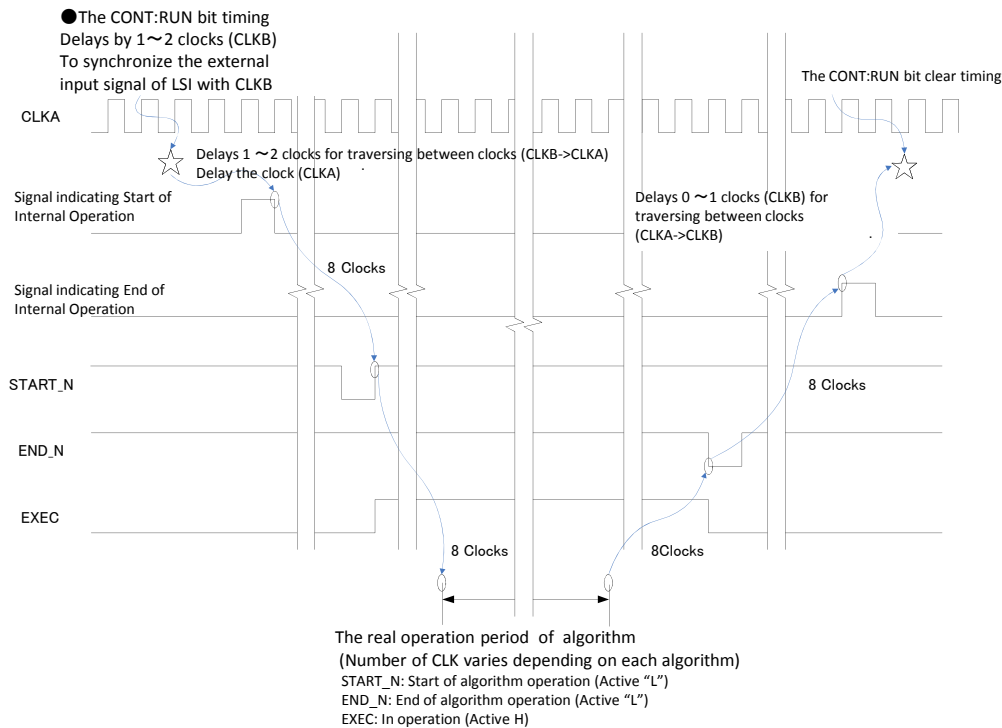


Figure 3-9 Timing Chart for Delayed Operation

3.8.4. Noise Source

The experimenter can exploit cryptographic IPs other than the targeted one as noise sources to evaluate their effects on power analysis or electromagnetic analysis in the future. To take advantage of this feature, select two or more cryptographic IPs on the IP Select Register IPSEL, while simultaneously selecting the target IP on the Output Select Register OUTSEL.

3.8.5. Free-Run Mode

AES0 (AES_Comp) core has the free-run mode in which encryption or decryption continues every 0.3 seconds. Turning the control register CONT[RUN] to '1' activates the free-run mode.

- This support is only for AES0 (AES_Comp).
- When the first run finishes, the control register CONT[RUN] returns to '0', and the mode keeps repeating the second run and after.
- Unless the export suppression function is enabled, the STARTS_N, EXEC, and END_N signals will be controlled accordingly as described in the delayed operation section.
- In the free-run mode, other operations cannot function. To cancel the free-run mode, impose the power reset or assert the HRST_N signal.
- The input text during this mode can be configured according to the following two options.
 1. The plaintext or ciphertext set in the Input Text Register ITEXT becomes the initial input value, every time each run finishes the value automatically increments by 1. (Set the mode register MODE[FRUN] to 2'b10.)
 2. The plaintext or ciphertext set in the Input Text Register ITEXT becomes the initial input value, the ciphertext or plaintext output from encryption or decryption becomes the next run's input. (Set the mode register MODE[FRUN] to 2'b11.)

3.8.6. Suppression of Exporting Evaluation Signals

To reduce noise emission from the control circuit while capturing power traces or electromagnetic waveforms, the evaluation signals START_N, END_N, EXEC, and STATE may be disabled by the following two ways.

1. To suppress all evaluation signals: set the mode register MODE[OUTINH] to 2'b10.
2. To suppress all evaluation signals except START_N: set the mode register MODE[OUTINH] to 2'b11.

3.8.7. Handling the Input Text Register (ITEXT)

For the symmetric-key cryptographic algorithm cores, note that for each core, as shown below, the size of Input Data and the mapping of Input Text Register varies.

1. AES5(CTR mode supported, pipelined implementation)
128bit x 4 block input
ITEXT0-7 128bit 1st block input
ITEXT8-15 128bit 2nd block input
ITEXT16-23 128bit 3rd block input
ITEXT24-31 128bit 4th block input
2. MISTY1, T-DES, DES, CAST-128(64bit block ciphers)
ITEXT0-3 64bit input
ITEXT4-31 unused
3. Other ciphers(128bit block ciphers)
ITEXT0-7 128bit input
ITEXT8-31 unused

3.8.8. Handling the Output Text Register (OTEXT)

For the symmetric-key cryptographic algorithm cores, note that for each core, as shown below, the size of Output Data and the mapping of Output Text Register varies

1. AES5(CTR mode supported, pipelined implementation)
128bit x 4 block output
OTEXT0-7 128bit 1st block output
OTEXT8-15 128bit 2nd block output
OTEXT16-23 128bit 3rd block output
OTEXT24-31 128bit 4th block output
2. MISTY1, T-DES, DES, CAST-128(64bit block ciphers)
OTEXT0-3 64bit output
OTEXT4-7 0x0000000000000000
OTEXT8-1F don't care
3. Other ciphers(128bit block ciphers)
OTEXT0-7 128bit output
OTEXT8-1F don't care

3.8.9. Handling the Random Number for DPA Countermeasure (SEED) Register (RAND)

- Used in AES8 (Masked AND Operation), AES9 (MDPL), AES10 (Threshold Implementation).
- If SEED data is provided to random number Data Register from the external input of cryptographic LSI, in the next encryption process, SEED is used.
- The random number used in AES9 (MDPL) is 32bit. The upper bits of random number Data Register become effective. (Input to RAND0-1.)

3.8.10. CTR Operation on AES5 (CTR mode supported, pipelined)

- Unlike the other cores, the AES5 core performs data input and output of 128bit x 4 blocks consecutively.
- If one sets the initial value of the counter to the initial value register (IV), since the core generates the random numbers for the 4 blocks for the CTR mode, the Output Text Register (OTEXT) will not immediately export the output text. In the next operation of cryptographic process, the Output Text Register (OTEXT) exports the encryption result of Input Text Register (ITEXT) and the next 4-block random number is generated.

3.8.11. Exporting Intermediate Value Data

The function to export the intermediate value data of cryptographic algorithm process.

- Support for AES6 (Fault injection attack countermeasure implemented) only. To enable this function, select AES6 on both the IP Select Register (IPSEL0) and the Output Select Register (OUTSEL0), and set the Round Selection Register DRSEL[DEN] to '1', and then the Intermediate Data Register (RDATA0-7) will export the intermediate data value.
- Use the Round Select Register DRSEL[DRSEL] to specify the exporting round.
- The intermediate value data can be obtained from the output Dout of AES6.

3.8.12. Exporting Intermediate Key Data

The function to export the intermediate key data of cryptographic algorithm process.

- Support for AES6 (Fault injection attack countermeasure implemented) only. To enable this function, select AES6 on both the IP Select Register (IPSEL0) and the Output Select Register (OUTSEL0), and set the Round Selection Register KRSEL[KEN] to '1', and then the Intermediate Key Register (RKEY0-7) will export the intermediate data value.
- Use the Round Select Register KRSEL[KRSEL] to specify the exporting round.
- The intermediate key data can be obtained from the output Kout of AES6.

3.8.13. Support for FA(Fault Attack) Countermeasure Function

This is the process when Fault Error occurs during the operation of cryptographic algorithm.

Only the fault Attack countermeasure implemented AES6 (implementation for evaluation of the tamper-resistance against fault attack). Inside AES6, when Fault Error occurs, Round Select Register KRSEL[KERR] or DRSEL[DERR] turns to "1", the intermediate value data and the intermediate key data are exported to Intermediate Value Register (RDATA0-7) and Intermediate Key Register (RKEY0-7) respectively.

3.8.14. The Behaviour of Interface Circuit during the Operation of Cryptographic Algorithm

In order to reduce the noise and to allow high precision measurement, as shown in Figure 3-10, we improve the design such that the interface circuit does not run during the operation of cryptographic algorithm.

Also, for cryptographic LSI developed in FY 2008, by executing the procedure below before the start of the encryption, we can obtain the same effect.

1. Write the key data on Key Register.
2. Write "1" on KSET of CONT register.
-> The key data is sent to cryptographic IP, key generation process is executed.
3. After that, write ALL "0" on Key Register.

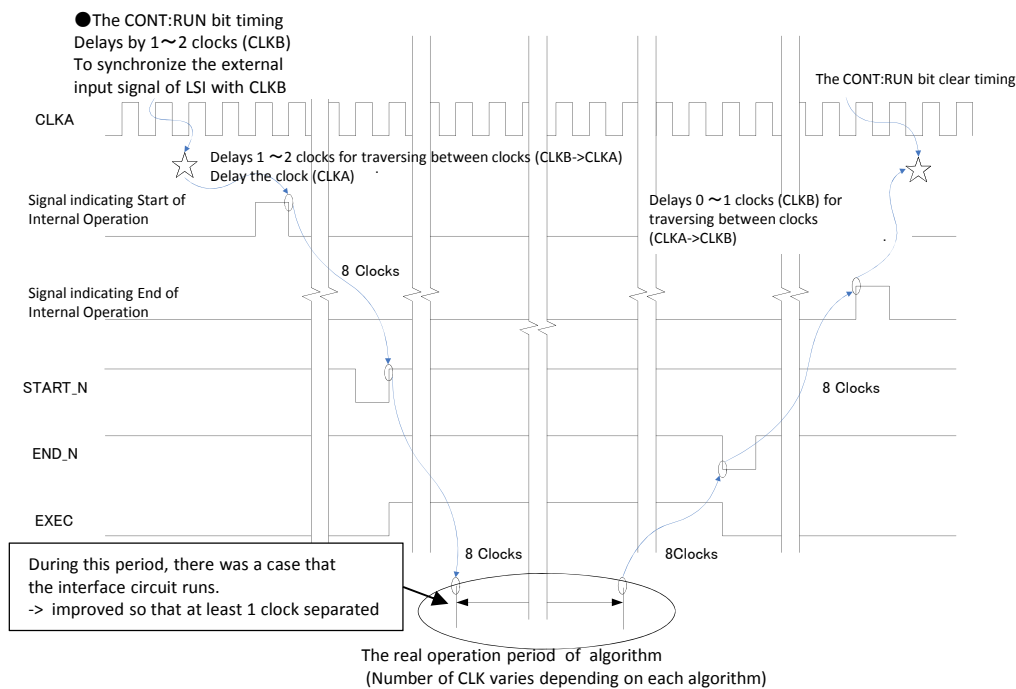


Figure 3-10 Improvement of the run period of interface

4. The Environment for Run Verification by Logic Simulation

4.1. The Outline of Run Verification Module

The run verification module is a module which verifies the behavior of the entire evaluated LSI by comparing the input signals to run the RTL of the entire LSI on the logic simulator and the output signals with the prepared expected output value. The run verification module, as same as the RTL of the entire evaluated LSI, is written in Hardware Description Language, and together with the RTL of evaluated LSI, is simulated on the top of logic simulator.

4.2. The Function of Run Verification Module

The input signals to evaluated LSI provided by the run verification module are based on test vector used in the testbench of each cryptographic algorithm core (the run verification module of each cryptographic algorithm core).

From the testbench of each cryptographic algorithm core, the key data and the input text data are drawn, and by providing the inputs according to the external specification shown in chapter 2, the RTL of evaluated LSI is run.

On the other hand, similarly, from the testbench of each cryptographic algorithm core, the expected output values are drawn, and by comparing them to the output of RTL, the verification of the logic process of the evaluated LSI is performed.

4.3. The Verification and Verification Result of Run Verification Module

The items for verification and their result are shown in Table 4-1 and Table 4-2.

Table 4-1 Verification Item and Verification Result(1/2)

Cryptographic Algorithm Core	Verification Item	Verification Data (Number)	Verification Result
AES0	Encryption	1 block	Passed
	Decryption	1block	Passed
AES1	Encryption	2 blocks with same data	Passed
AES2	Encryption	2 blocks with same data	Passed
AES3	Encryption	2 blocks with same data	Passed
AES4	Encryption	2 blocks with same data	Passed
AES5		8 blocks	Passed
AES6	Encryption	1block	Passed
	Intermediate value data during encryption	Output each intermediate value from 0th round(only KEY and EXOR) - 20th round	Passed
	Intermediate key data during encryption	Output each intermediate value from 0th round(only KEY and EXOR) - 20th round	Passed
	Fault attack during encryption	Output to Intermediate Value Register, Intermediate Key Register	Passed
	Decryption	1 block	Passed
	Intermediate value data during decryption	Output each intermediate value from 0th round(only KEY and EXOR) - 20th round	Passed
	Intermediate key data during decryption	Output each intermediate value from 0th round(only KEY and EXOR) - 20th round	Passed
	Fault attack during decryption	Output to Intermediate Value Register, Intermediate Key Register	Passed
AES7	Encryption	2 blocks with same data	Passed
AES8	Encryption	2 blocks with same data	Passed
AES9	Encryption	2 blocks with same data	Passed
AES10	Encryption	2 blocks with same data	Passed
AES11	Encryption	2 blocks with same data	Passed
AES12	Encryption	2 blocks with same data	Passed
AES13	Encryption	2 blocks with same data	Passed

Table 4-2 Verification Item and Verification Result (2/2)

Cryptographic Algorithm Core	Verification Item	Verification Data (Number)	Verification Result
Camellia	Encryption	1block	Passed
	Decryption	1block	Passed
SEED	Encryption	1block	Passed
	Decryption	1block	Passed
MISTY1	Encryption	1block	Passed
	Decryption	1block	Passed
T-DES (Triple-DES)	Encryption	1block	Passed
	Decryption	1block	Passed
DES	Encryption	3 blocks	Passed
	Decryption	3 blocks	Passed
CAST-128	Encryption	1block	Passed
	Decryption	1block	Passed
CLEFIA	Encryption	1block	Passed
	Decryption	1block	Passed
ECC	Computation	100 blocks	Passed
RSA	Modulo Exponentiation (Normal Mode)	For all 6 types of calculation method, 512 bit once.	Passed
	Modulo Exponentiation (CRT Mode)	For all 6 types of calculation method, 512bit (256 bit x 2) once.	Passed

5. The Restriction of Logic Synthesis

It is necessary to pay attention on the following point when proceed to the logic synthesis of RTL of the evaluated LSI.

“CTS(Clock Tree Synthesis) and clock skew control which considers the gated clock.”

Inside the evaluated LSI, all clocks which use CLK_A (core clock), i.e., the clock which runs the cryptographic algorithm core, are designed to be one phase synchronous clocks. Therefore, it is necessary that in the process of CTS and the clock skew control during the logic synthesis/wiring, all clock systems below are handled as the same clock line:

- 23 clock trees for cryptographic algorithm cores controlled by IP Select Register,
- 1 clock tree for intermediate data value output control circuit of AES 6,
- 1 clock tree for intermediate key value output control circuit of AES 6, and
- 1 clock tree for traversing with interface clock in the normal time CLK_B.

For the detailed clock systems, refer to Figure 3-5.

6. Physical Layout of the LSI

6.1. Design Environment

The developed cryptographic LSI utilizes 77.72% gates of 2.1 x 2.1 mm² die size. The target operating frequency was 24MHz, while the logic synthesis was performed after adding 30% margin of the final target operating frequency to the clock frequency in order to ease the timing adjustment at layout phase. The summary of the LSI is shown in Table 6-1. The libraries used are supplied by e-Shuttle Inc. The detailed information of the used libraries is given in Table 6-2 and the conditions of logic synthesis in Table 6-3.

Table 6-1 Summary of the cryptographic LSI.

Library	Fujitsu e-Shuttle CS202 (LVt)
Process	65nm CMOS
Wiring Layer	Metal 12 layers
Die size	2.1mm x 2.1mm
Package	Ceramic QFP 160 pin
Cell Area	1,404,242 um ²
Number of Gates	731,376 2-NAND gates
Cell Utilization	77.72%
Operating frequency	35.1MHz

Table 6-2 The cell library used.

Classification	Library	Process
Standard Cell	CS202L CS202MZ(12 Tracks LVt)	CS202L
Digital I/O	CS202L Common	CS202L
RAM	ChaRAM	CS202L

Table 6-3 Synthesis conditions

Conditions	Value	
Target frequency	CLKA	31MHz (32000 ps) including 30% margin
	CLKB	31MHz (32000 ps) including 30% margin
I/O Delay	Input Delay	2000 ps
	Output Delay	2000 ps
Dummy Load	cs202mx_3600area	

6.2. Result of Synthesis

The logic after synthesis is shown in Table 6-4 and the power consumption estimation in Table 6-5.

Table 6-4 Area report

Level Name	Cell Area [um ²]	Cell Area [2-input NAND Conversion※]	Percentage[%]	Number of Instances
J_SASEBO_ASIC_TOP	1,387,994	722,914	100	269,562
U_AES_RDATA1	1,928	1,004	0.1	161
U_AES_RDATA2	1,928	1,004	0.1	161
U_J_AES0	42,104	21,929	3	10,349
U_J_AES1	34,969	18,213	2.5	11,618
U_J_AES2	95,268	49,619	6.9	28,435
U_J_AES3	25,500	13,281	1.8	6,565
U_J_AES4	20,007	10,421	1.4	4,975
U_J_AES5	39,274	20,455	2.8	7,178
U_J_AES6	32,041	16,688	2.3	7,601
U_J_AES7	37,295	19,424	2.7	4,827
U_J_AES8	72,646	37,836	5.2	13,220
U_J_AES9	127,345	66,326	9.2	27,810
U_J_AES10	207,247	107,941	14.9	30,477
U_J_AES11	56,056	29,196	4	11,610
U_J_AES12	61,210	31,880	4.4	10,842
U_J_AES13	38,101	19,845	2.7	5,717
U_J_CAMELLIA	23,230	12,099	1.7	6,627
U_J_SEED	33,951	17,683	2.4	10,893
U_J_MISTY1	27,338	14,239	2	8,774
U_J_T_DES	8,763	4,564	0.6	2,066
U_J_DES	5,501	2,865	0.4	1,497
U_J_CAST	46,095	24,008	3.3	13,118
U_J_ECC	138,838	72,311	10	16,984
U_J_RSA	106,884	55,669	7.7	16,692
U_J_CLEFIA	15,576	8,113	1.1	3,521

※2-input NAND Conversion= SC43BUFXC1 (1.92[um²]) unit

Table 6-5 Report on Power Consumption Estimation

Level Name	Switching Power[W]	Internal Power[W]	Leak Power[pW]	Total Power[W]	Percentage[%]
J_SASEBO_ASIC_TOP	2.69E-04	1.04E-03	2.94E+10	3.07E-02	100
U_AES_RDATA2	3.69E-09	7.19E-09	4.49E+07	4.49E-05	0.1
U_AES_RDATA1	3.69E-09	7.19E-09	4.49E+07	4.49E-05	0.1
U_J_AES0	2.32E-08	1.00E-07	7.47E+08	7.47E-04	2.4
U_J_AES1	1.37E-08	5.10E-08	6.24E+08	6.25E-04	2
U_J_AES2	1.49E-08	5.28E-08	1.42E+09	1.42E-03	4.6
U_J_AES3	1.30E-08	5.12E-08	4.17E+08	4.17E-04	1.4
U_J_AES4	1.30E-08	5.23E-08	3.45E+08	3.45E-04	1.1
U_J_AES5	5.28E-06	1.70E-07	6.66E+08	6.72E-04	2.2
U_J_AES6	9.23E-09	3.73E-08	5.57E+08	5.57E-04	1.8
U_J_AES7	4.40E-05	3.02E-05	4.20E+08	4.94E-04	1.6
U_J_AES8	1.24E-08	3.92E-08	2.59E+09	2.59E-03	8.5
U_J_AES9	5.34E-06	8.15E-08	2.36E+09	2.37E-03	7.7
U_J_AES10	5.29E-06	9.74E-08	7.39E+09	7.40E-03	24.1
U_J_AES11	5.26E-06	6.62E-08	1.23E+09	1.23E-03	4
U_J_AES12	1.31E-08	4.69E-08	2.00E+09	2.00E-03	6.5
U_J_AES13	6.80E-09	3.35E-08	1.21E+09	1.21E-03	3.9
U_J_CAMELLIA	0	0	3.71E+08	3.71E-04	1.2
U_J_SEED	0	0	6.18E+08	6.18E-04	2
U_J_MISTY1	0	0	4.75E+08	4.75E-04	1.5
U_J_T_DES	0	0	1.41E+08	1.41E-04	0.5
U_J_DES	0	0	8.86E+07	8.86E-05	0.3
U_J_CAST	0	0	1.06E+09	1.06E-03	3.4
U_J_ECC	1.40E-05	1.71E-05	1.87E+09	1.90E-03	6.2
U_J_RSA	0	0	1.76E+09	1.76E-03	5.7
U_J_CLEFIA	0	0	2.22E+08	2.22E-04	0.7

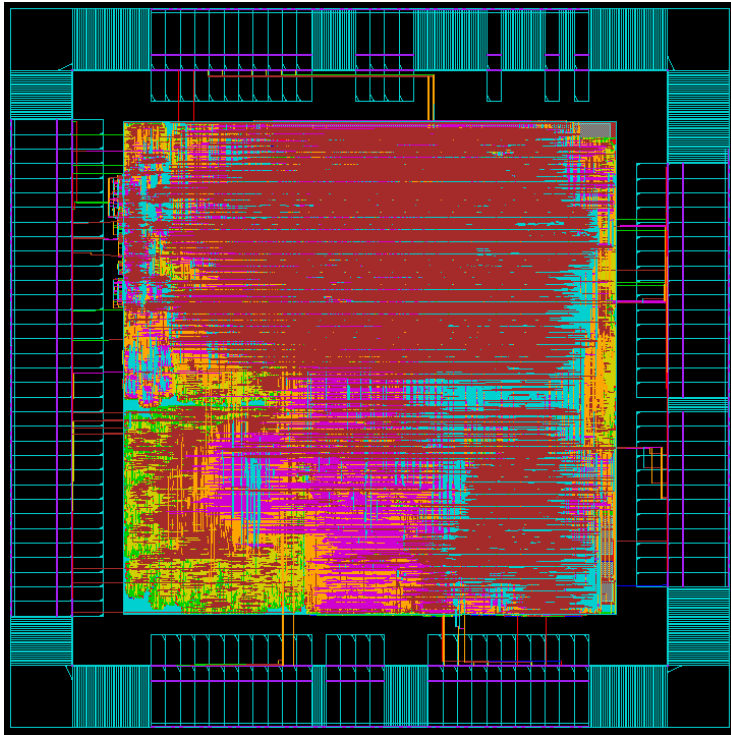
Switching Power: Power consumption from the output load capacity of drive cell.

Internal Power: Power consumption generated by the change on the input of drive cell (power consumption due to pass-through current included).

Leak Power: Power consumption at the time of run and stand-by.

6.3. Power Source Plan

Figure 6-1 shows the signal wiring without the power wiring in Metal10 - Metal12, and Figure 6-2 shows the power line. The power mesh is constructed by stretching the stripe on the internal power ring. The detailed explanation on power ring region and power mesh region will be presented in the next page. The power mesh is constructed using Metal4, Metal7, Metal9, and Metal10 as Figure 6-3 shows. The power to the standard cell is supplied from 4Metal stripes through Stack Via as Figure 6-4 shows. Each mesh is directly connected to the power ring and Metal11 and Metal12 are used as the incoming power lines for IO-BUF as Figure 6-5 shows.



- Metal1: ■
- Metal2: ■
- Metal3: ■
- Metal4: ■
- Metal5: ■
- Metal6: ■
- Metal7: ■
- Metal8: ■
- Metal9: ■

Figure 6-1 Top View of the Cryptographic LSI

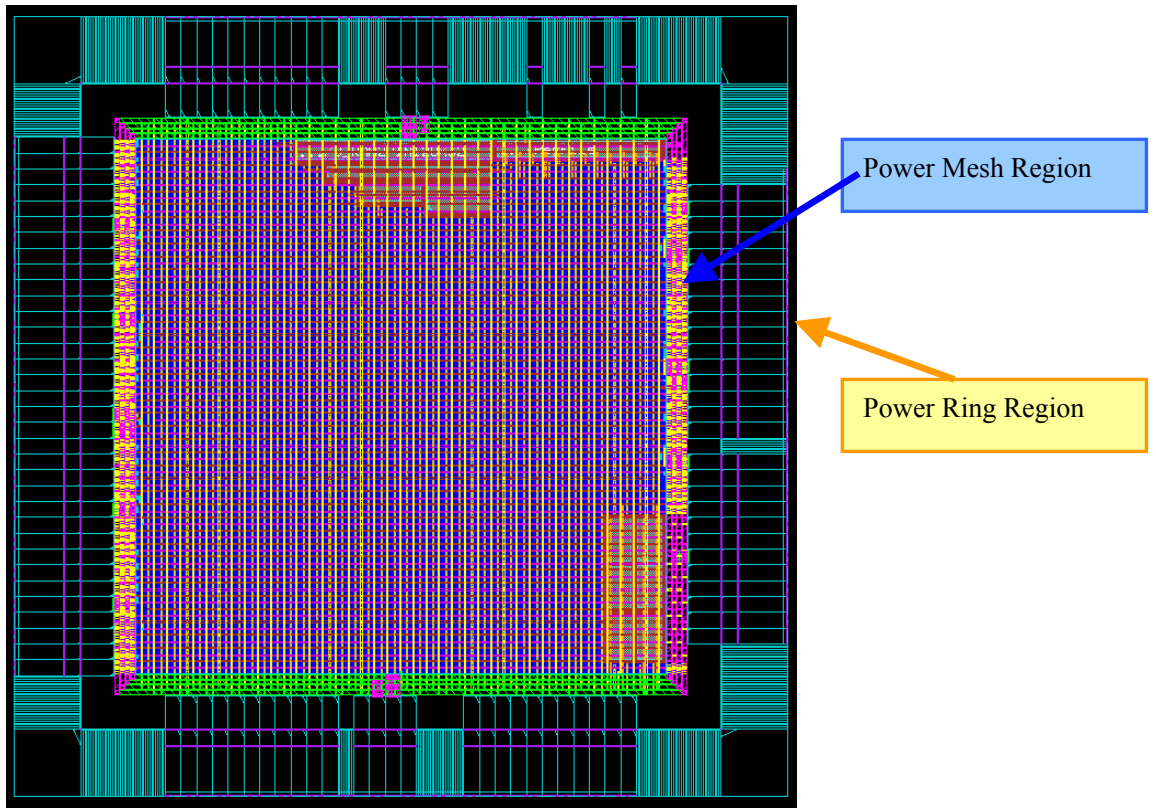


Figure 6-2 Power Line

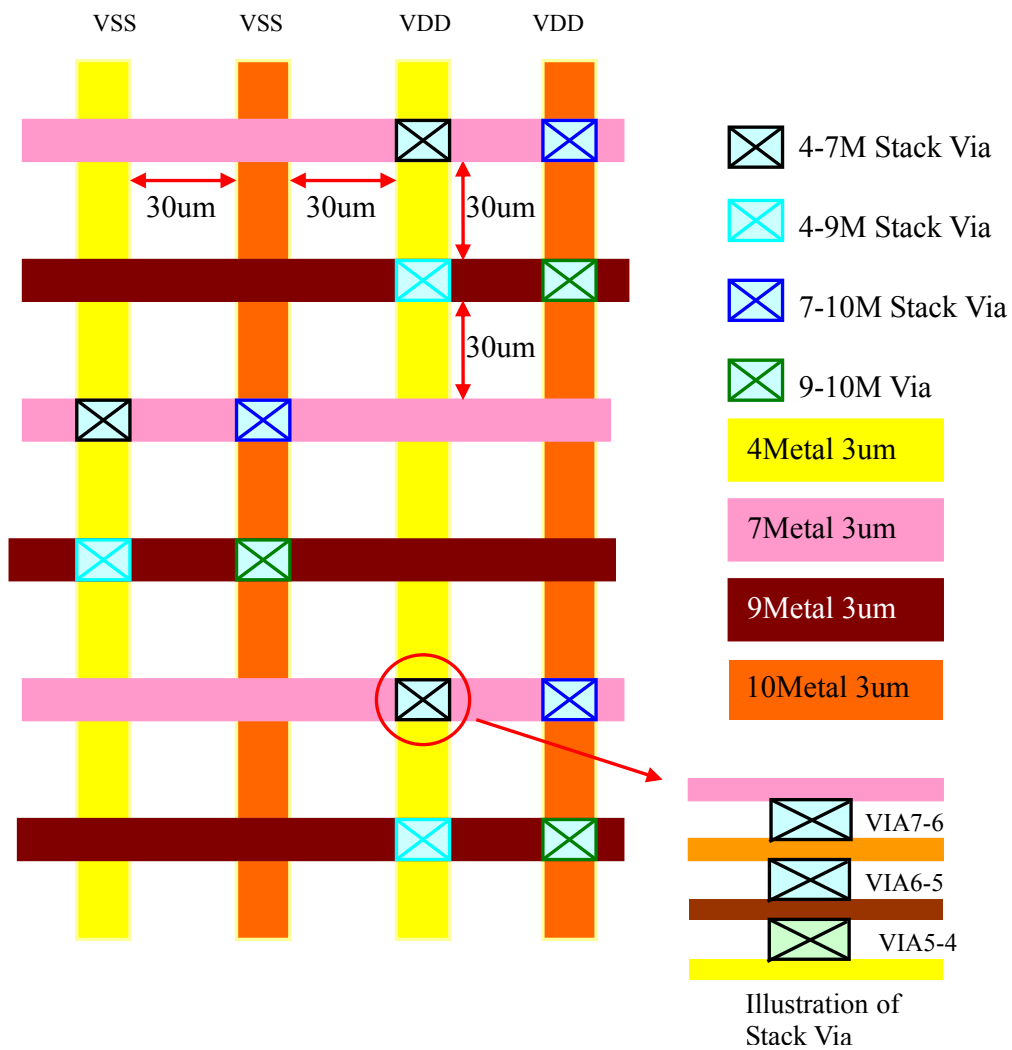


Figure 6-3 Power Mesh

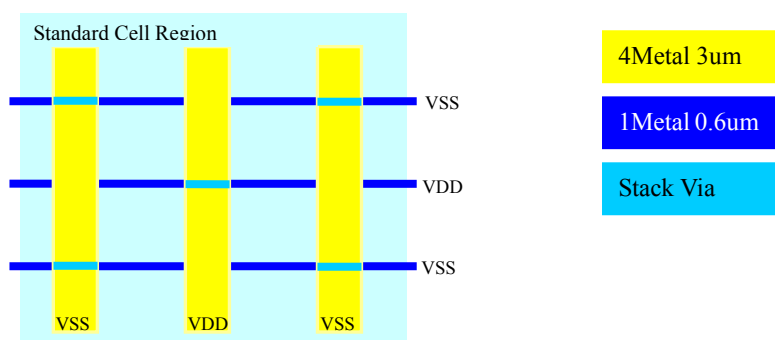


Figure 6-4 Power Supply to Standard Cell

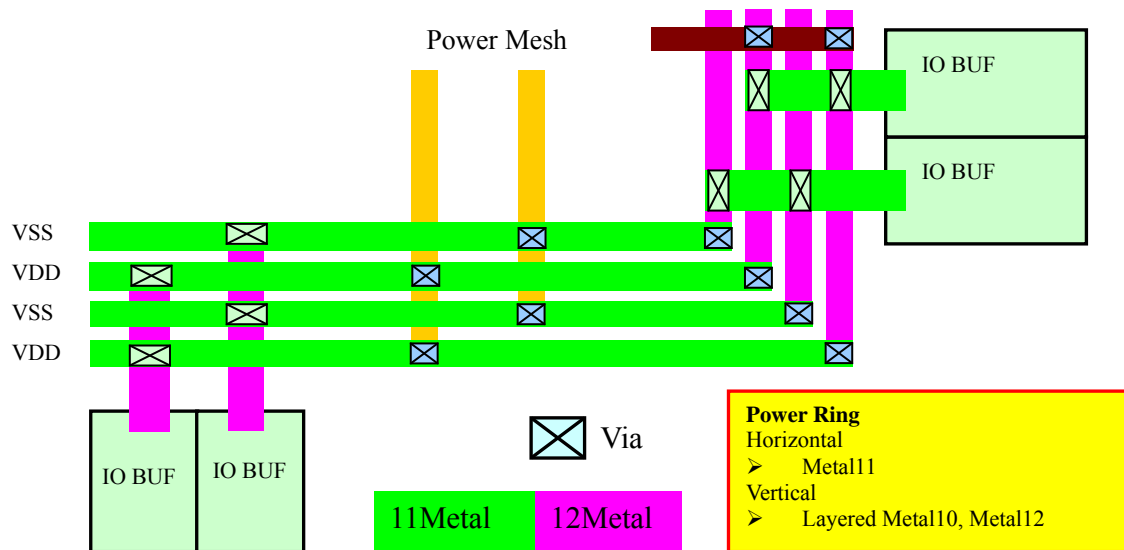


Figure 6-5 Power Ring

6.4. Macro Layout

Figure 6-6 illustrates the layout of the load macros.

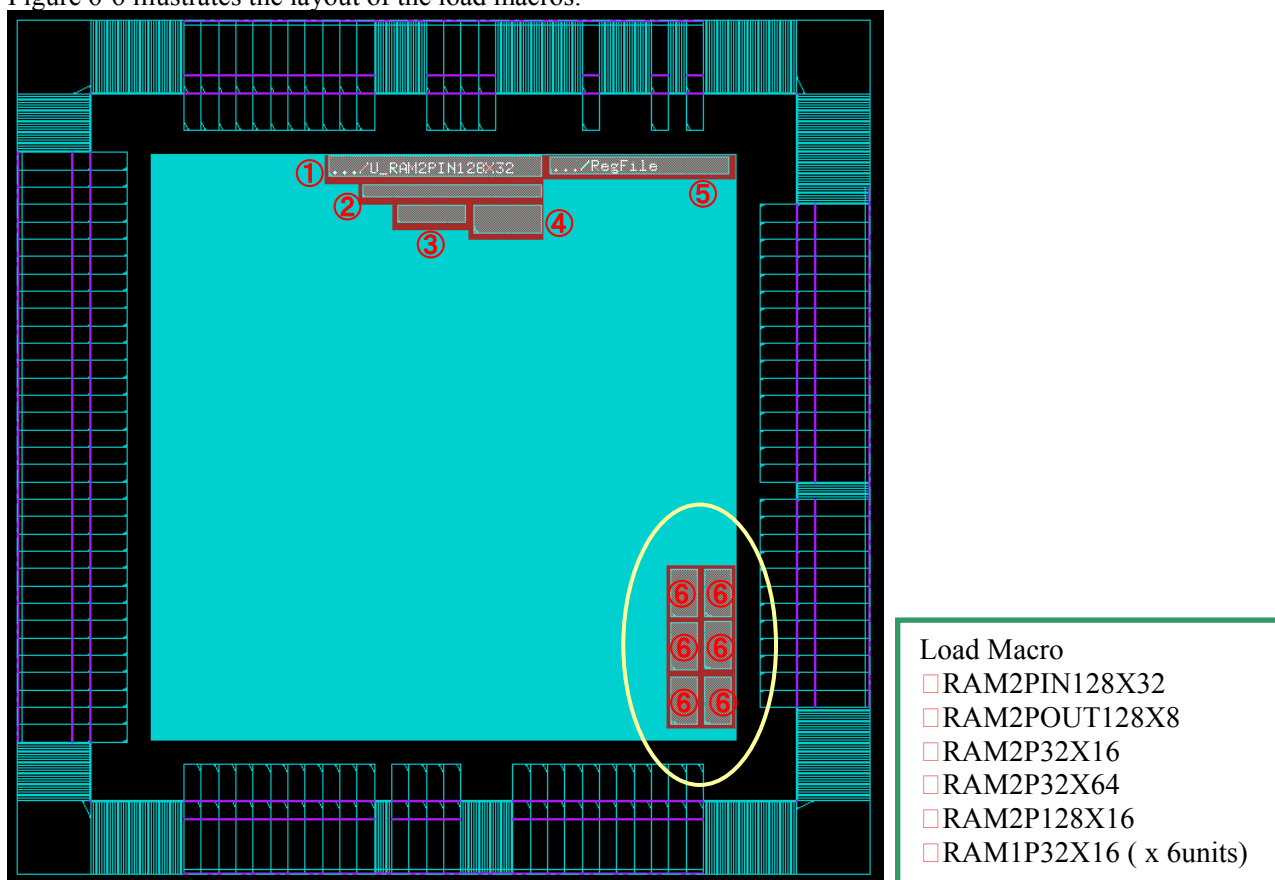


Figure 6-6 The layout of the cryptographic macros.

6.5. Main Module Layout

Figure 6-6 illustrates the layout of the main cryptographic modules. The placement of main modules on the netlist is highlighted. For these modules, at the initial placement step, designation of placement region is performed so that they are placed based on the layout with divisions according to the functions.

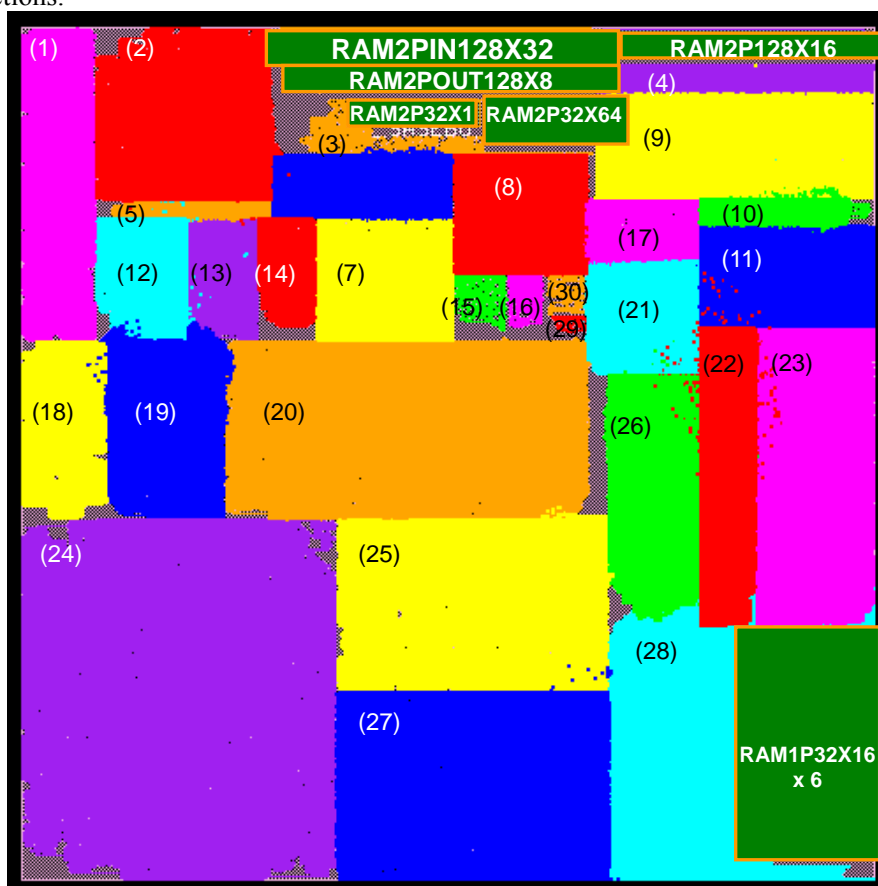


Figure 6-7 The layout of the main modules.

Table 6-6 The main modules.

Module Name	Number in Chart	Cell's area [um ²]	Number of instances	Module Name	Number in Chart	Cell's area [um ²]	Number of instances
U_J_AES0	26	45,344	11,098	U_J_CAMELLIA	6	24,927	7,257
U_J_AES1	22	36,470	11,940	U_J_SEED	7	35,223	11,179
U_J_AES2	25	97,306	28,814	U_J_MISTY1	18	27,799	8,876
U_J_AES3	21	27,110	6,895	U_J_T_DES	14	9,612	2,349
U_J_AES4	12	21,852	5,324	U_J_DES	5	5,825	1,576
U_J_AES5	19	46,213	8,732	U_J_CAST	1	46,993	13,239
U_J_AES6	8	33,339	7,993	U_J_ECC	28	158,845	21,600
U_J_AES7	4	38,663	5,047	U_J_RSA	27	114,815	18,517
U_J_AES8	23	62,055	13,602	U_J_CLEFIA	13	17,364	4,097
U_J_AES9	20	131,948	28,769	U_AES_RDATA1	15	3,573	345
U_J_AES10	24	171,577	30,930	U_AES_RDATA2	16	3,320	356
U_J_AES11	2	60,260	12,404	U_SASEBO_VALUE	29	269	68
U_J_AES12	9	53,180	11,129	U_SASEBO_ALGO_OUTPUT	10	13,319	2,679
U_J_AES13	11	33,927	6,083	U_SASEBO_INPUT	30	1,361	106
U_SASEBO_REG	3	75,072	926	U_SASEBO_ALGO_INPUT	17	20,565	5,638

6.6. Report on Cell Area

Using the final netlist, the layout is performed with Design Compiler. The area report from Design Compiler is shown in Table 6-7, and area comparison between before and after layout is given in Table 6-8.

Table 6-7 Report on Area after Layout

Level Name	Cell Area[um ²]	Cell Area [2-input NAND Conversion※]	Percentage[%]	Number of instances
J_SASEBO_ASIC_TOP	1,404,242	731,376	100	289,344
U_AES_RDATA1	2,680	1,396	0.2	345
U_AES_RDATA2	2,491	1,297	0.2	356
U_J_AES0	45,344	23,617	3.2	11,098
U_J_AES1	36,470	18,995	2.6	11,940
U_J_AES2	97,306	50,680	6.9	28,814
U_J_AES3	27,110	14,120	1.9	6,895
U_J_AES4	21,852	11,381	1.6	5,324
U_J_AES5	46,213	24,069	3.3	8,732
U_J_AES6	33,339	17,364	2.4	7,993
U_J_AES7	38,663	20,137	2.8	5,047
U_J_AES8	62,055	32,321	4.4	13,602
U_J_AES9	131,948	68,723	9.4	28,769
U_J_AES10	171,577	89,363	12.2	30,930
U_J_AES11	60,260	31,385	4.3	12,404
U_J_AES12	53,180	27,698	3.8	11,129
U_J_AES13	33,927	17,671	2.4	6,083
U_J_CAMELLIA	24,927	12,983	1.8	7,257
U_J_SEED	35,223	18,345	2.5	11,179
U_J_MISTY1	27,799	14,479	2	8,876
U_J_T_DES	9,612	5,007	0.7	2,349
U_J_DES	5,825	3,034	0.4	1,576
U_J_CAST	46,993	24,476	3.3	13,239
U_J_ECC	158,845	82,732	11.3	21,600
U_J_RSA	114,815	59,799	8.2	18,517
U_J_CLEFIA	17,364	9,044	1.2	4,097

※2-input NAND Conversion = SC43BUFXC1 (1.92[um²]) Units

Table 6-8 Area Comparison Before and After Layout

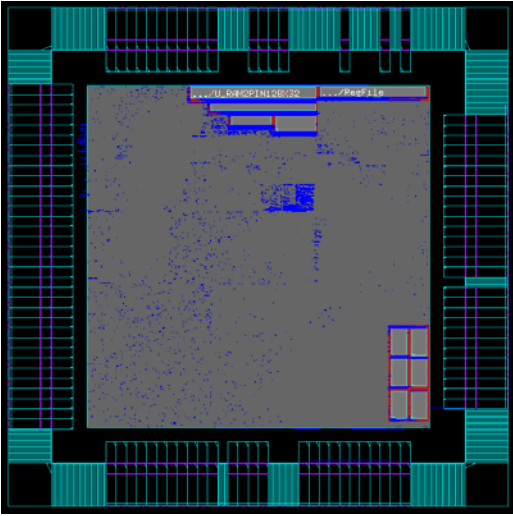
Level Name	After Layout			Immediately After Logic Synthesis		
	Cell Area[um ²]	Cell Area 2-input NAND Conversion [*]	Percentage [%]	Cell Area [um ²]	Cell Area 2-input NAND Conversion [*]	Percentage [%]
J_SASEBO_ASIC_TOP	1,404,242	731,376	100	1,387,994	722,914	100
U_AES_RDATA1	2,680	1,396	0.2	1,928	1,004	0.1
U_AES_RDATA2	2,491	1,297	0.2	1,928	1,004	0.1
U_J_AES0	45,344	23,617	3.2	42,104	21,929	3
U_J_AES1	36,470	18,995	2.6	34,969	18,213	2.5
U_J_AES2	97,306	50,680	6.9	95,268	49,619	6.9
U_J_AES3	27,110	14,120	1.9	25,500	13,281	1.8
U_J_AES4	21,852	11,381	1.6	20,007	10,421	1.4
U_J_AES5	46,213	24,069	3.3	39,274	20,455	2.8
U_J_AES6	33,339	17,364	2.4	32,041	16,688	2.3
U_J_AES7	38,663	20,137	2.8	37,295	19,424	2.7
U_J_AES8	62,055	32,321	4.4	72,646	37,836	5.2
U_J_AES9	131,948	68,723	9.4	127,345	66,326	9.2
U_J_AES10	171,577	89,363	12.2	207,247	107,941	14.9
U_J_AES11	60,260	31,385	4.3	56,056	29,196	4
U_J_AES12	53,180	27,698	3.8	61,210	31,880	4.4
U_J_AES13	33,927	17,671	2.4	38,101	19,845	2.7
U_J_CAMELLIA	24,927	12,983	1.8	23,230	12,099	1.7
U_J_SEED	35,223	18,345	2.5	33,951	17,683	2.4
U_J_MISTY1	27,799	14,479	2	27,338	14,239	2
U_J_T_DES	9,612	5,007	0.7	8,763	4,564	0.6
U_J_DES	5,825	3,034	0.4	5,501	2,865	0.4
U_J_CAST	46,993	24,476	3.3	46,095	24,008	3.3
U_J_ECC	158,845	82,732	11.3	138,838	72,311	10
U_J_RSA	114,815	59,799	8.2	106,884	55,669	7.7
U_J_CLEFIA	17,364	9,044	1.2	15,576	8,113	1.1

^{*}2-input NAND Conversion = SC43BUFXC1 (1.92[um²]) Units

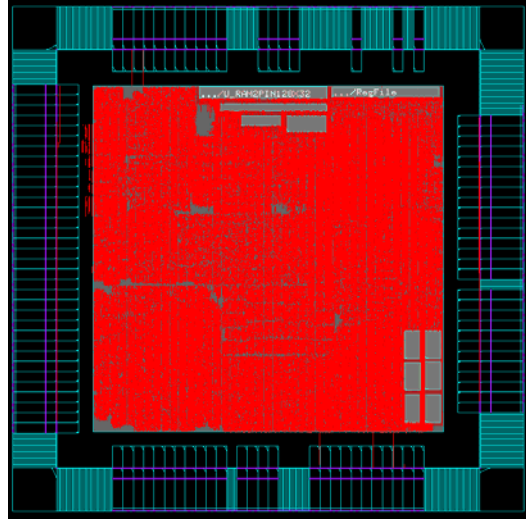
6.7. Signal Wiring

Metal 1-9 are used for signal wiring. In figures below, power wirings are made invisible, while the states of signal wirings are shown.

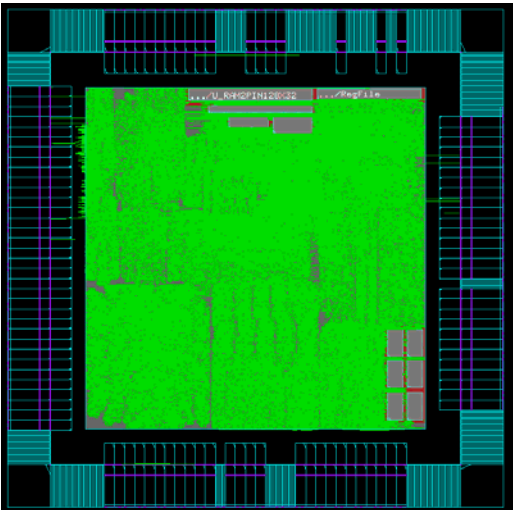
◆Metal 1



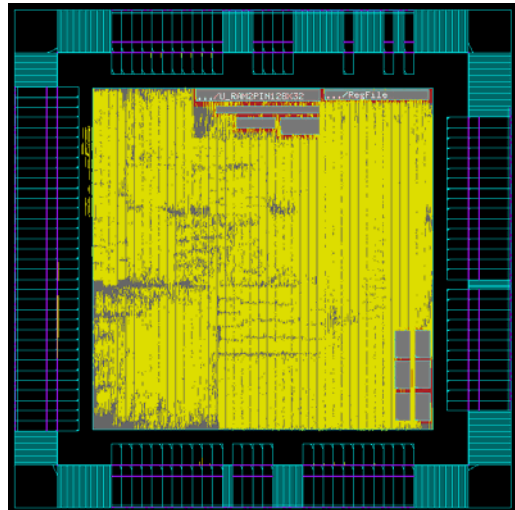
◆Metal 2



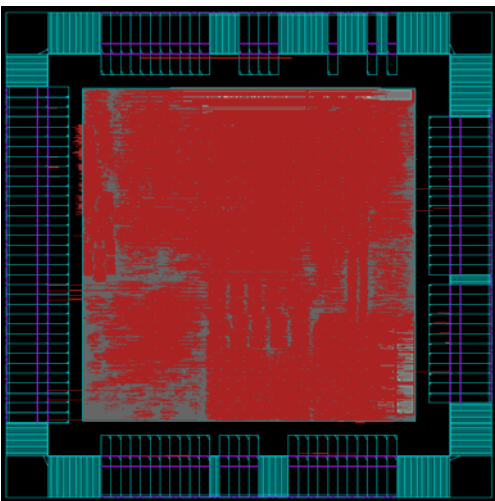
◆Metal 3



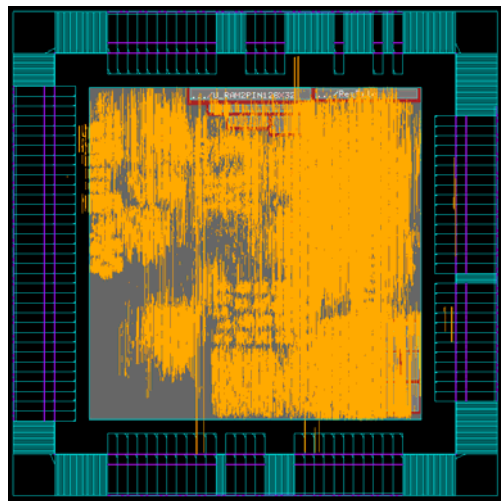
◆Metal 4



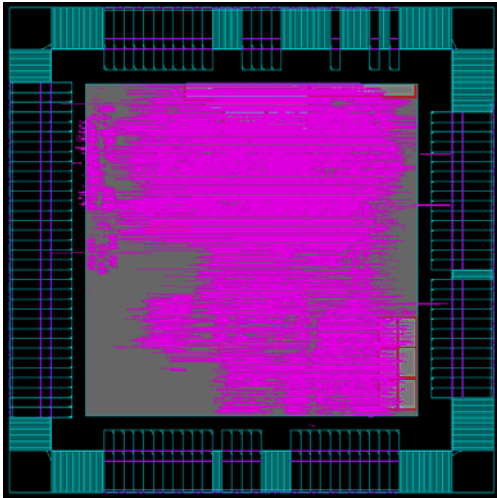
◆Metal 5



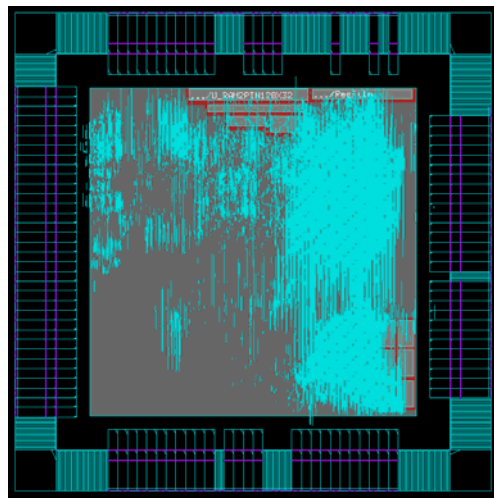
◆Metal 6



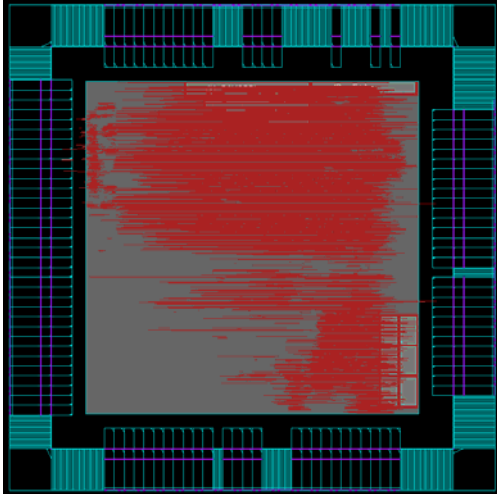
◆Metal 7



◆Metal 8



◆Metal9



6.8. About Insertion of Dummy Metal

In the metal layer of the chip from Fujitsu CS202L process, other than wirings for signals and powers on the layout, in order to satisfy the metal density condition, the dummy metal pattern was installed.

In Figure 6-8, it is illustrated the dummy metal pattern which is installed between the power meshes. The space between the power meshes is covered with square metal pattern. The size of the squares are shown below, it differs depending on the metal layer.

- Metal 1 – Metal 9 ... $0.7 \times 0.7 \text{ um}^2$
- Metal 10 – Metal 11 ... $1.0 \times 0.7 \text{ um}^2$
- Metal 12 ... $2.2 \times 0.7 \text{ um}^2$

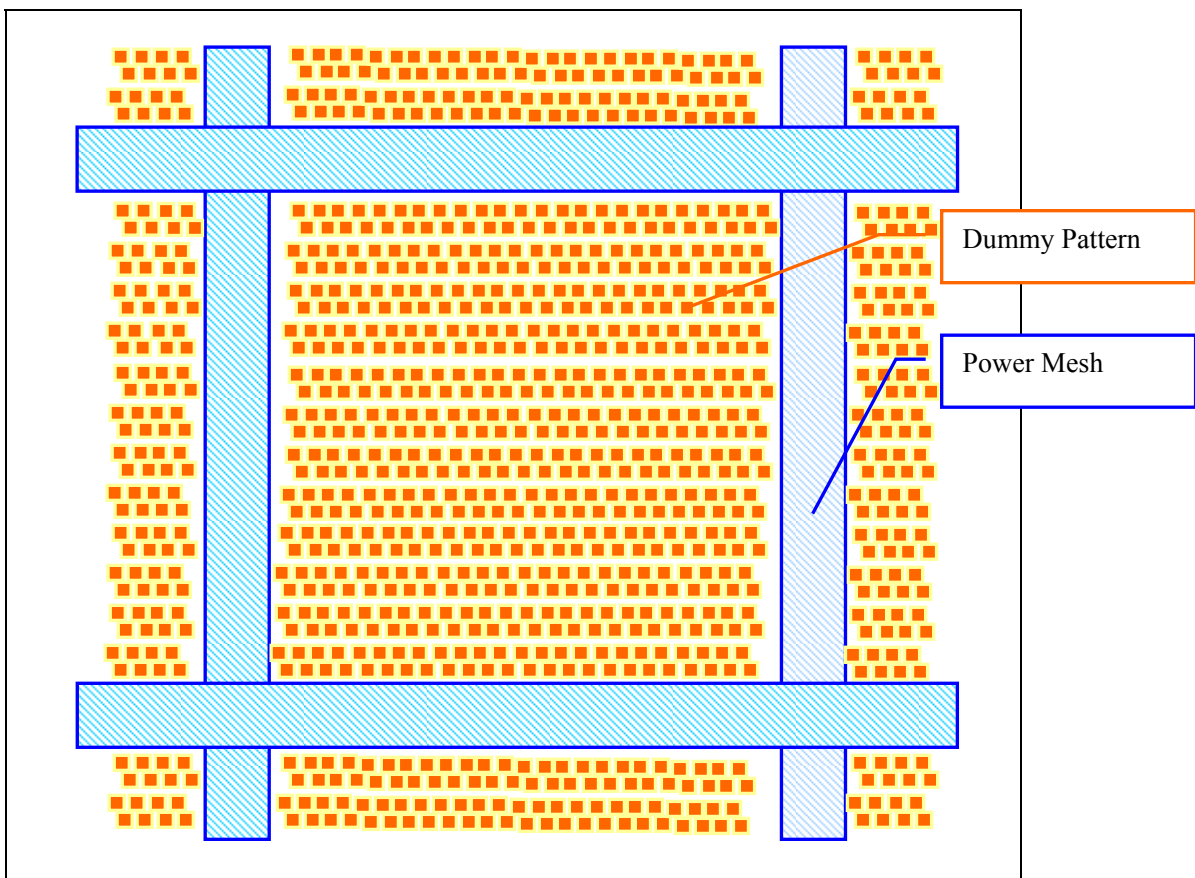


Figure 6-8 Dummy Metal

6.9. On the Chip's Orientation Matching

To show the chip's orientation, on the top right corner, "F" mark is put. Pad placement on the chip is as follows: the top right side is as the Number 1, and then from that, the placement is anti clockwise direction. The signal name of each pad and the center coordinate are listed in the next page.

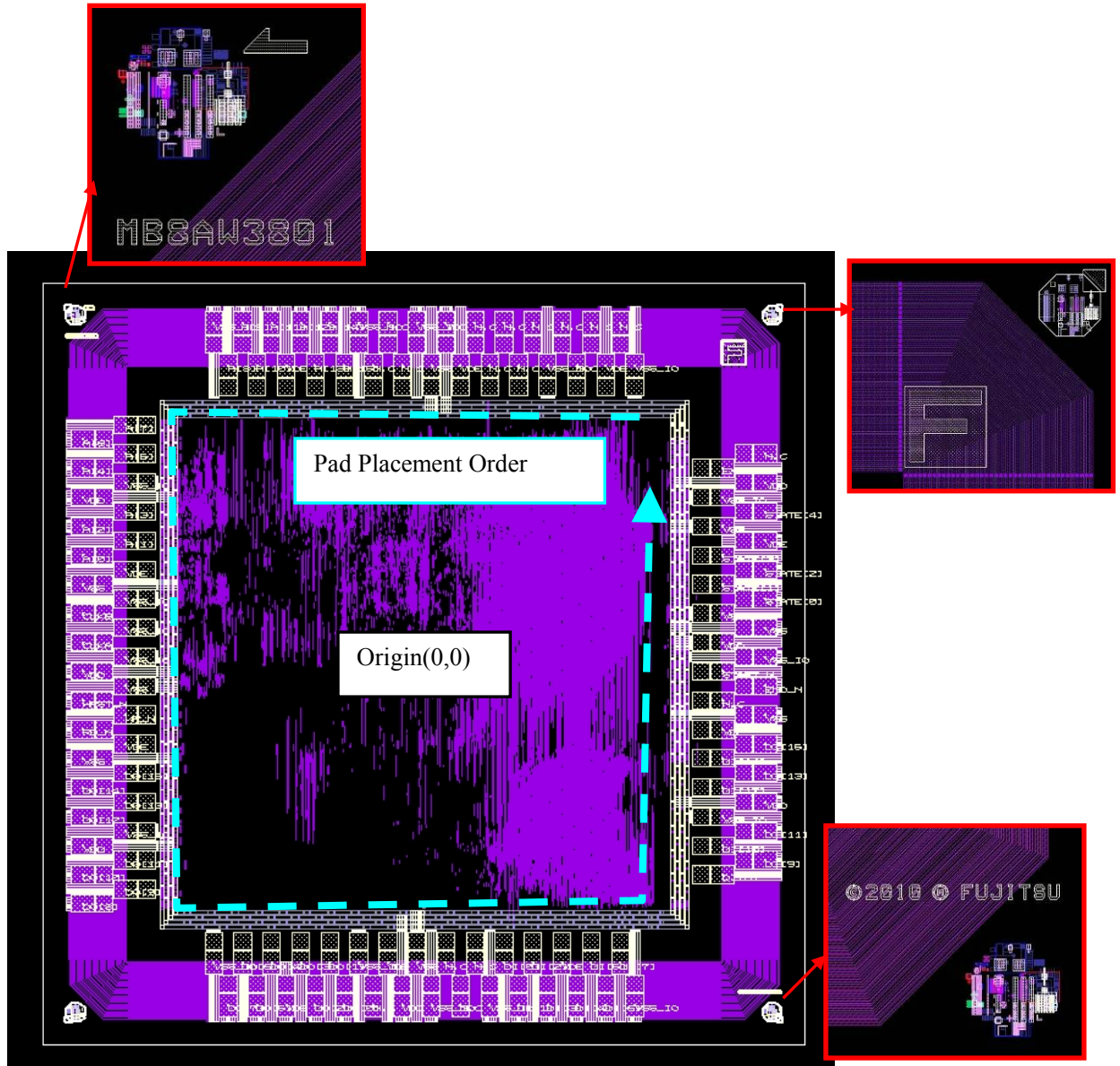


Figure 6-9 Orientation Matching of the Chip

Table 6-9 Signal - pad center coordinate (Top side /Left side)

Number	Signal Name	Chip Pad Coordinate		Number	Signal Name	Chip Pad Coordinate	
		x [um]	y [um]			x [um]	y [um]
1	VSS_IO	580	822.5	31	A[7]	-822.5	660
2	N.C	540	939.5	32	A[6]	-939.5	620
3	VDE	500	822.5	33	A[5]	-822.5	580
4	N.C	460	939.5	34	A[4]	-939.5	540
5	N.C	420	822.5	35	VSS_IO	-822.5	500
6	N.C	380	939.5	36	VDD	-939.5	460
7	VSS_IO	340	822.5	37	A[3]	-822.5	420
8	N.C	300	939.5	38	A[2]	-939.5	380
9	N.C	260	822.5	39	A[1]	-822.5	340
10	N.C	220	939.5	40	A[0]	-939.5	300
11	N.C	180	822.5	41	VDE	-822.5	260
12	N.C	140	939.5	42	VSS	-939.5	220
13	VDE	100	822.5	43	VSS_IO	-822.5	180
14	VDD	60	939.5	44	CLKB	-939.5	140
15	VSS	20	822.5	45	VSS_IO	-822.5	100
16	VSS_IO	-20	939.5	46	CLKA	-939.5	60
17	N.C	-60	822.5	47	VSS_IO	-822.5	20
18	N.C	-100	939.5	48	VDD	-939.5	-20
19	N.C	-140	822.5	49	VSS	-822.5	-60
20	VSS_IO	-180	939.5	50	HRST_N	-939.5	-100
21	A[15]	-220	822.5	51	WR_N	-822.5	-140
22	A[14]	-260	939.5	52	RD_N	-939.5	-180
23	A[13]	-300	822.5	53	VDE	-822.5	-220
24	A[12]	-340	939.5	54	VSS	-939.5	-260
25	VDE	-380	822.5	55	DO[15]	-822.5	-300
26	A[11]	-420	939.5	56	DO[14]	-939.5	-340
27	A[10]	-460	822.5	57	DO[13]	-822.5	-380
28	A[9]	-500	939.5	58	DO[12]	-939.5	-420
29	A[8]	-540	822.5	59	VSS_IO	-822.5	-460
30	VSS_IO	-580	939.5	60	VDD	-939.5	-500
				61	DO[11]	-822.5	-540
				62	DO[10]	-939.5	-580
				63	DO[9]	-822.5	-620
				64	DO[8]	-939.5	-660

VDD: Power for Core, VDE: Power for IO, VSS: GND for Core, VSS_IO: GND for IO

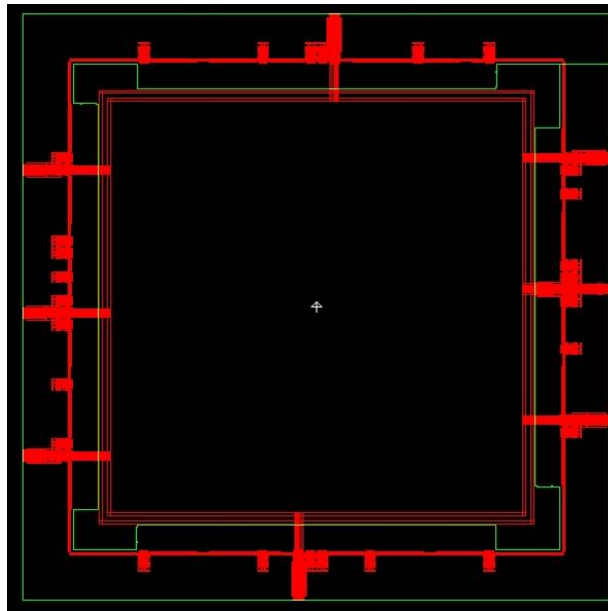
Table 6-10 Signal - pad center coordinate (Bottom side, Right side)

Number	Signal Name	Chip Pad Coordinate		Number	Signal Name	Chip Pad Coordinate	
		x [um]	y [um]			x [um]	y [um]
65	VSS_IO	-580	-822.5	95	DI[8]	822.5	-580
66	DO[7]	-540	-939.5	96	DI[9]	939.5	-540
67	DO[6]	-500	-822.5	97	DI[10]	822.5	-500
68	DO[5]	-460	-939.5	98	DI[11]	939.5	-460
69	DO[4]	-420	-822.5	99	VSS_IO	822.5	-420
70	VDE	-380	-939.5	100	VDD	939.5	-380
71	DO[3]	-340	-822.5	101	DI[12]	822.5	-340
72	DO[2]	-300	-939.5	102	DI[13]	939.5	-300
73	DO[1]	-260	-822.5	103	DI[14]	822.5	-260
74	DO[0]	-220	-939.5	104	DI[15]	939.5	-220
75	VSS_IO	-180	-822.5	105	VDE	822.5	-180
76	N.C	-140	-939.5	106	VSS	939.5	-140
77	VDE	-100	-822.5	107	N.C	822.5	-100
78	VDD	-60	-939.5	108	END_N	939.5	-60
79	VSS	-20	-822.5	109	START_N	822.5	-20
80	VSS_IO	20	-939.5	110	VSS_IO	939.5	20
81	N.C	60	-822.5	111	VDD	822.5	60
82	N.C	100	-939.5	112	VSS	939.5	100
83	N.C	140	-822.5	113	VSS_IO	822.5	140
84	VSS_IO	180	-939.5	114	STATE[0]	939.5	180
85	DI[0]	220	-822.5	115	STATE[1]	822.5	220
86	DI[1]	260	-939.5	116	STATE[2]	939.5	260
87	DI[2]	300	-822.5	117	STATE[3]	822.5	300
88	DI[3]	340	-939.5	118	VDE	939.5	340
89	VDE	380	-822.5	119	VSS	822.5	380
90	DI[4]	420	-939.5	120	STATE[4]	939.5	420
91	DI[5]	460	-822.5	121	VSS_IO	822.5	460
92	DI[6]	500	-939.5	122	VDD	939.5	500
93	DI[7]	540	-822.5	123	EXEC	822.5	540
94	VSS_IO	580	-939.5	124	N.C	939.5	580

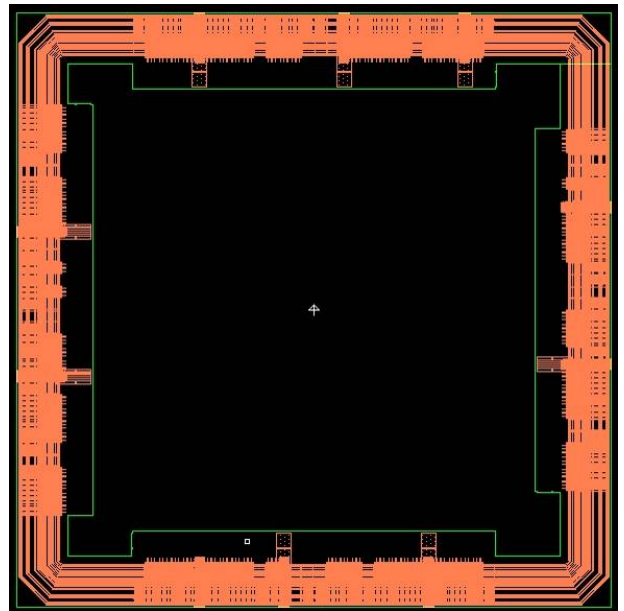
VDD: Power for Core, VDE: Power for IO, VSS: GND for Core, VSS_IO: GND for IO

6.10. The Result of Verifying Power Separation

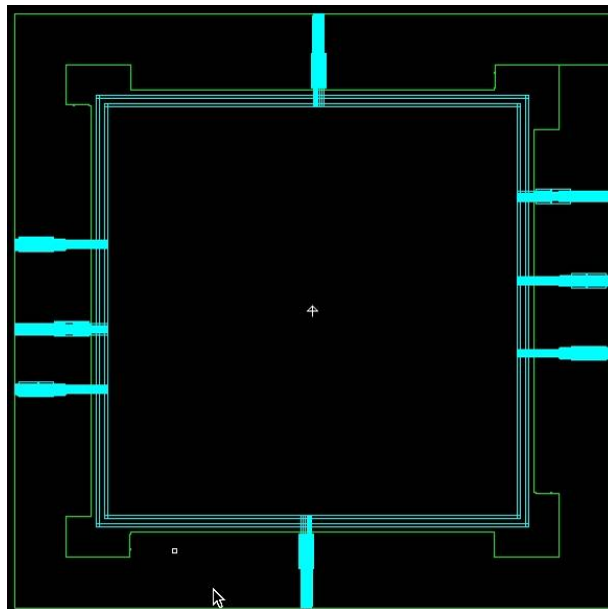
Scanning from power source PAD along the metal wiring, we have verified that there is no short with other power sources or signal wirings. Figure 6-10 shows the wiring pattern of the power supply layer.



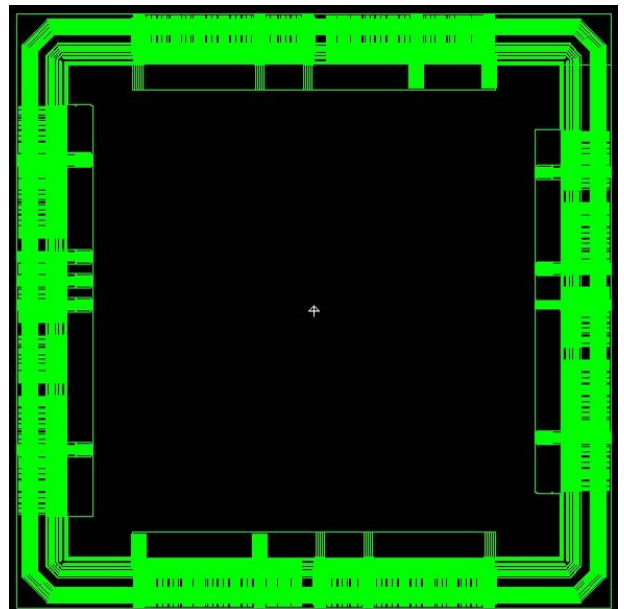
Core VDD (1.2V)



I/O VDD (3.3V)



Core VSS



IO VSS

Figure 6-10 Power Separation Test

7. IR-Drop Verification

Combining VDD/VSS, we get drop ratio 0.8535%. In the time of STA, we perform the verification with running margin larger than this value. Figure 7-1 shows the IR-drop of the VDD and VSS planes under the assumption that 30% of the entire cells are active. Three results of the IR-drop verification are given in Table 7-1. The IR-drop of VDD is 0.3985% and VSS is 0.4455%. Considering that only one module is activated at the same time in the actual LSI, the result values are small enough for the chip to correctly work. STA is performed with a sufficient margin larger than these values.

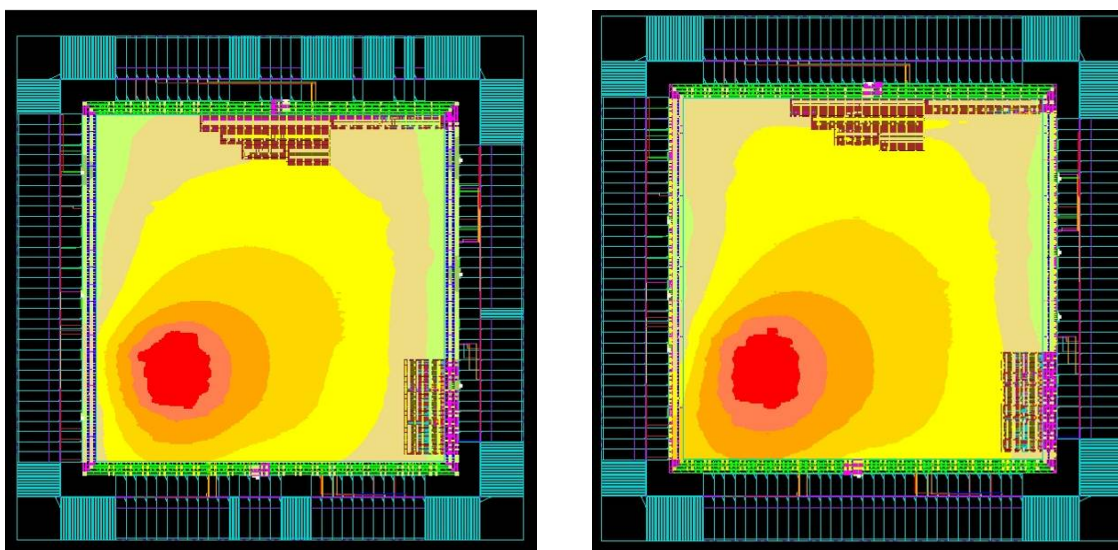


Figure 7-1 Drop of VDD (left) and VSS (right)

Table 7-1 Drop Verification of VDD and VSS

	VDD	VSS
Operating Frequency	24MHz	
Operating Condition	best (1.3V, -40 deg C)	
Primary Input Activity	30%	
Sequential Element Activity	10%	
Clock Gates Enable Activity	10%	
Power Consumption	106 mW	106 mW
Value of Worst Drop	5.18 mV	5.791 mV
Drop Ratio	0.3985%	0.4455%

8. X-Talk Noise Test

8.1. About X-Talk Noise Test

Between two long parallel signal wirings, there is possibility that the net in running state (aggressor net) affects the neighboring net in static state (victim net) and it can cause noise error in the net in static state. In X-Talk Noise test, the possibility of occurrence of running errors caused by noises from such parallel signal wirings is checked.

In the test, 4 variances from manufacture (tc, tcw, capb, capw) and 7 running conditions (worst, worstLT, nominal, nonirworst, nonirworstLT, best, bestHT) are considered, i.e., total 28 test patterns. The threshold values are taken from the library values.

8.2. Test Result

As an illustration, the test result with manufacture condition capb is shown below.

◆ The Noise test result with worst condition (1.05V, 125 deg C)

```
*****
# Run settings
# Run mode           = coupling analysis with RCs
# Process            = 65nm
# Failure Thresholds
# Functional (sequential) = 0.26 V
# Functional (combinatorial) = 0.84 V
# Delay absolute     = 1000.00 ps
# Delay relative     = 0.50
# Slope              = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped = 0
# For complete details, view the html or text ECO report
# *****
```

◆ The Noise test result with worstLT condition (1.05V, -40 deg C)

```
*****
# Run settings
# Run mode           = coupling analysis with RCs
# Process            = 65nm
# Failure Thresholds
# Functional (sequential) = 0.26 V
# Functional (combinatorial) = 0.84 V
# Delay absolute     = 1000.00 ps
# Delay relative     = 0.50
# Slope              = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped = 0
# For complete details, view the html or text ECO report
# *****
```

◆ The Noise test result with nonirworst condition (1.1V, 125 deg C)

```
*****
# Run settings
# Run mode                = coupling analysis with RCs
# Process                  = 65nm
# Failure Thresholds
# Functional (sequential)  = 0.28 V
# Functional (combinatorial) = 0.88 V
# Delay absolute           = 1000.00 ps
# Delay relative           = 0.50
# Slope                    = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped    = 0
# For complete details, view the html or text ECO report
# *****
```

◆ The Noise test result with nonirworstLT condition (1.1V, -40 deg C)

```
*****
# Run settings
# Run mode                = coupling analysis with RCs
# Process                  = 65nm
# Failure Thresholds
# Functional (sequential)  = 0.28 V
# Functional (combinatorial) = 0.88 V
# Delay absolute           = 1000.00 ps
# Delay relative           = 0.50
# Slope                    = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped    = 0
# For complete details, view the html or text ECO report
# *****
```

◆ The Noise test result with nominal condition (1.2V, 25 deg C)

```
*****
# Run settings
# Run mode                = coupling analysis with RCs
# Process                  = 65nm
# Failure Thresholds
# Functional (sequential)  = 0.30 V
# Functional (combinatorial) = 0.96 V
# Delay absolute           = 1000.00 ps
# Delay relative           = 0.50
```

```

# Slope = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped = 0
# For complete details, view the html or text ECO report
# *****

```

◆ The Noise test result with best condition (1.3V, -40 deg C)

```

*****
# Run settings
# Run mode = coupling analysis with RCs
# Process = 65nm
# Failure Thresholds
# Functional (sequential) = 0.33 V
# Functional (combinatorial) = 1.04 V
# Delay absolute = 1000.00 ps
# Delay relative = 0.50
# Slope = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped = 0
# For complete details, view the html or text ECO report
*****

```

◆ The Noise test result with bestHT condition (1.3V, 125 deg C)

```

*****
# # Run settings
# Run mode = coupling analysis with RCs
# Process = 65nm
# Failure Thresholds
# Functional (sequential) = 0.33 V
# Functional (combinatorial) = 1.04 V
# Delay absolute = 1000.00 ps
# Delay relative = 0.50
# Slope = 0.00
# Number of glitch ECO failures = 0
# Number of delay ECO failures = 0
# Number of slope ECO failures = 0
# Number of ECOs skipped = 0
# For complete details, view the html or text ECO report
#
*****

```

9. STA Test

9.1. STA Condition

In STA analysis with delay information from real wirings, there was no setup timing errors, while violations were detected at Clock Gating. These errors are treated as pseudo errors which do not affect the correct operation of the LSI. For the details of the pseudo errors, please refer to Section 9.2. STA is performed under the conditions shown in Table 9-1 and Table 9-2. The results show that no error is detected except the pseudo errors in clock gating. For the detailed clock spec, please refer to Section 9.3.

Table 9-1 STA Conditions

Tool		PrimetimeSI	
Netlist used		J_SASEBO_ASIC_TOP.v	
Process condition (PROCESS)		tc, tcw, capw, capb	
Running condition (COND)		best (1.3V / -40 deg C)	
		bestHT (1.3V / 125 deg C)	
		worst (1.05V / 125 deg C)	
		worstLT (1.05V / -40 deg C)	
		nonirworst (1.1V / 125 deg C)	
		nonirworstLT (1.1V / -40 deg C)	
STA Conditions		Setup Frequency	24MHz (41666 ps)
		Input Delay	2000 ps
		Output Delay	2000 ps
		Derate Factor	表 18 参照
		False Path	CLKA domain <--> CLKB domain

Table 9-2 Derate Factor

COND	-net_delay		-cell_delay		set_clock_uncertainty[ps]
	early	late	early	late	
best / bestHT	0.93	1	0.95	1.07	10
worst / worstLT nonirworst / nonirworstLT	0.93	1	0.9	1.04	10
nominal	1	1	1	1	10

9.2. Summary on Clock Gating Timing Error

9.2.1. Timing Error Spots

Hold violation errors of clock gating occurred in CTS ROOT Cells shown in Table 9-3.

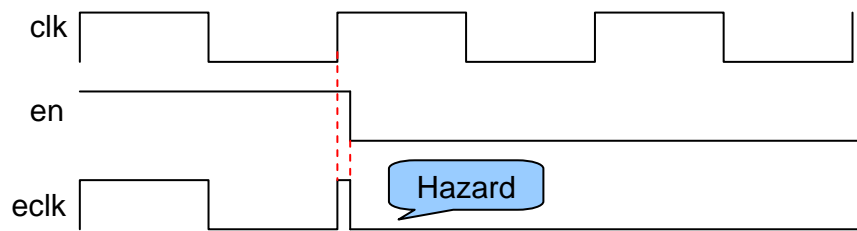
Table 9-3 The list of the gates where Hold Violations on Clock Gating occur.

U_AES0_CLK_GATE	U_AES9_CLK_GATE	U_MISTY1_CLK_GATE
U_AES1_CLK_GATE	U_AES10_CLK_GATE	U_T_DES_CLK_GATE
U_AES2_CLK_GATE	U_AES11_CLK_GATE	U_DES_CLK_GATE
U_AES3_CLK_GATE	U_AES12_CLK_GATE	U_CAST_CLK_GATE
U_AES4_CLK_GATE	U_AES13_CLK_GATE	U_ECC_CLK_GATE
U_AES5_CLK_GATE	U_AES_RDATA1_CLK_GATE	U_RSA_CLK_GATE
U_AES6_CLK_GATE	U_AES_RDATA2_CLK_GATE	U_CLEFIA_CLK_GATE
U_AES7_CLK_GATE	U_CAMELLIA_CLK_GATE	
U_AES8_CLK_GATE	U_SEED_CLK_GATE	

9.2.2. Problems of Clock Gating Timing Error

When the Hold Violation occurs at Clock Gating, hazards (“mustaches”) or narrow clocks might be generated and could affect the running circuit (Refer to the following figure).

at Stop



at Release

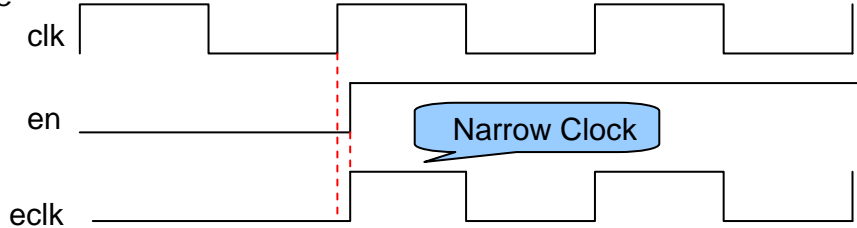


Figure 9-1 Hazard and Narrow Clock

In this product, en signals are only changing during few μ SECs of reset state. Since it will enter reset state after hazards or narrow clocks, there is no actual effect on the circuit. (Refer to the following figure)

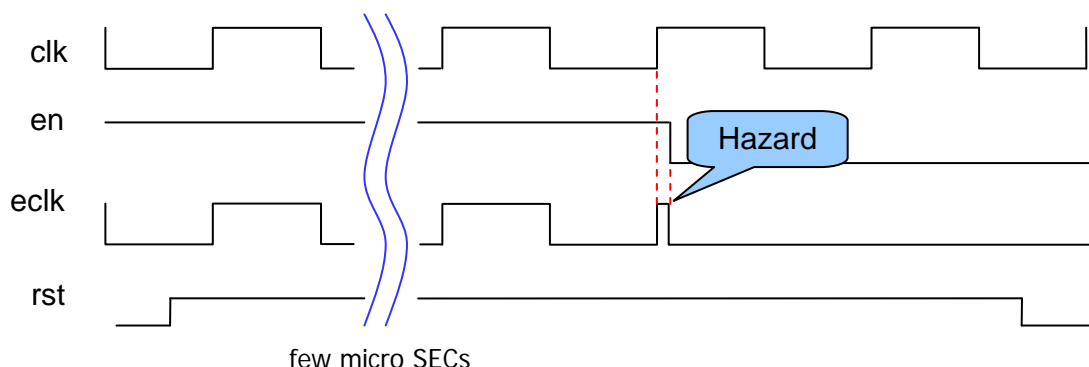


Figure 9-2 Hazard during the reset status

9.3. Maximum Operating Speed

The maximum operating speeds reported in STA are shown in the following tables. **The red color is the worst value among all conditions.** Static Timing Analysis (STA) with X-Talk on final data considered was done. It was confirmed that no timing error (Setup/Hold) at all conditions.

Table 9-4 PROCESS = capb

COND	CLKA			CLKB		
	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]
worst	13,771.5	35.8	36.5	39,011.0	376.6	131.6
worstLT	14,547.1	36.9	4.3	39,354.9	432.7	102.4
nonirworst	14,353.0	36.6	36.8	39,223.4	409.4	125.2
nonirworstLT	15,182.3	37.8	1.7	39,580.8	479.6	101.4
nominal	16,693.6	40.0	20.0	40,081.3	631.0	112.5
best	17,951.2	42.2	27.1	40,495.1	854.0	86.0
bestHT	17,407.8	41.2	37.9	40,224.4	693.7	81.4

Table 9-5 PROCESS = capw

COND	CLKA			CLKB		
	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]
worst	13,201.4	35.1	46.7	38,674.4	334.3	153.1
worstLT	14,044.0	36.2	13.2	39,072.7	385.6	146.6
nonirworst	13,833.4	35.9	47.9	38,918.2	363.9	156.1
nonirworstLT	14,737.0	37.1	8.6	39,327.5	427.6	141.6
nominal	16,382.8	39.6	17.4	39,881.6	560.4	143.5
best	17,730.6	41.8	27.5	40,351.7	760.9	93.4
bestHT	17,145.6	40.8	42.7	40,031.5	611.8	106.3

Table 9-6 PROCESS = tc

COND	CLKA			CLKB		
	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]
worst	13,507.1	35.5	42.3	38,857.3	356.0	152.3
worstLT	14,313.5	36.6	8.4	39,222.7	409.3	124.1
nonirworst	14,112.6	36.3	42.2	39,082.3	387.0	154.0
nonirworstLT	14,977.8	37.5	4.7	39,466.5	454.7	118.9
nominal	16,551.2	39.8	26.1	39,987.0	595.6	126.6
best	17,850.5	42.0	29.7	40,430.0	809.1	89.6
bestHT	17,288.8	41.0	40.7	40,137.1	654.1	93.5

Table 9-7 PROCESS = tcw

COND	CLKA			CLKB		
	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]	SETUP[ps]	Maximum Running Frequency[MHz]	HOLD[ps]
worst	13,698.6	35.8	38.3	38,944.2	367.4	144.5
worstLT	14,477.3	36.8	5.6	39,294.4	421.7	110.8
nonirworst	14,286.7	36.5	38.5	39,160.3	399.1	136.8
nonirworstLT	15,109.3	37.7	15.0	39,530.2	468.2	108.0
nominal	16,638.9	40.0	21.3	40,043.0	616.2	116.8
best	17,912.5	42.1	26.9	40,464.9	832.5	85.8
bestHT	17,363.8	41.1	38.4	40,186.9	676.1	82.8

9.4. Not Annotated Analysis

Not Annotated analysis will report if there exists a Net such that its real load capacitance information is unconsidered. If the normal wiring becomes “Not Annotated”, since the parasitic capacitance is not considered in STA, the correct timing test will not be performed. But, for the nets connected to VDD, VSS (TIE Hi / TIE Lo) or the actual unused ports (UNCONNECT / FLOAT_PIN), there is no problem as they have no Timing Arc on STA.

Net Type	Total	Lumped	RC pi	network	Annotated
Internal nets					
- Pin to pin nets	290937	0	0	290918	19
- Driverless nets	5297	0	0	0	5297
- Loadless nets	111	6	0	0	105
Boundary/port nets					
- Pin to pin nets	61	0	0	61	0
- Driverless nets	0	0	0	0	0
- Loadless nets	0	0	0	0	0
	296406	6	0	290979	5421

Breakdown of Not Annotated Net:

Pin to pin nets + Loadless nets = 124 = TIEHi Net(28) + TIELo Net(96)

Driverless nets = 5297 = UNCONNECT(2116) + FLOAT_PIN(3171)

10. Formal Verification

The formal verification is performed on RTL submitted by National Institute of Advanced Industrial Science and Technology (AIST) and the final netlist after layout and it is checked whether they are functionally equivalence.

The test whether the RTL released by AIST and the netlist after layout are functionally equivalence was performed. In equivalence test running log, **SUCCEEDED** is written, thus we can conclude the above two are functionally equivalence.

```

***** Verification Results *****
Verification SUCCEEDED

ATTENTION: synopsys_auto_setup mode was enabled.
           See Synopsys Auto Setup Summary for details.

ATTENTION: RTL interpretation messages were produced during link
           of reference design.

           Verification results may disagree with a logic simulator.

-----

Reference design: r:/WORK/J_SASEBO_ASIC_TOP
Implementation design: i:/WORK/J_SASEBO_ASIC_TOP
25079 Passing compare points

-----

Matched Compare Points   BBPin   Loop   BBNet   Cut   Port   DFF   LAT   TOTAL
-----
Passing (equivalent)    1109     0     0     0    24  23946   0  25079
Failing (not equivalent)  0       0     0     0     0     0     0   0
*****

```

Figure 10-1 The extracted result of the formal verification.

Net Type	Total	Lumped	RC pi	RC network	Not Annotated
Internal nets					
- Pin to pin nets	290937	0	0	290918	19
- Driverless nets	5297	0	0	0	5297
- Loadless nets	111	6	0	0	105

Boundary/port nets						
- Pin to pin nets	61	0	0	61	0	
- Driverless nets	0	0	0	0	0	
- Loadless nets	0	0	0	0	0	
	296406	6	0	290979	5421	

Breakdown of Not Annotated Net:

Pin to pin nets + Loadless nets = 124 = TIEHi Net(28) + TIELo Net(96)

Driverless nets = 5297 = UNCONNECT(2116) + FLOAT_PIN(3171)

11. Layout Test

11.1. DRC

Using Calibre v2008.3_16.12, it is tested whether the design rule on Fujitsu CS202 Process is satisfied. The test environment is shown in Table 11-1. Since the result shows error=0, we can conclude that all conditions in the design rule are satisfied in our design. Figure 11-1 is the extracted result of DRC.

Table 11-1 DRC test environment

Tool (Version)	Calibre (v2008.3_16.12)
GDS File Name	MB8AW3801_10031701.gds
Top Cell Name	MB8AW3801
Rule File Name	MB8AW3801_drc.rul
Rule Version	r2.91

===== CALIBRE::DRC-H	
SUMMARY REPORT	
==	
Execution Date/Time:	Wed Mar 17 13:46:42 2010
Calibre Version:	v2008.3_16.12 Tue Aug 19 13:58:10 PDT 2008
Rule File Pathname:	MB8AW3801_drc.rul
Rule File Title:	
Layout System:	GDS
Layout Path(s):	MB8AW3801_10031701.gds
Layout Primary Cell:	MB8AW3801
Excluded Cells:	

--- RULECHECK RESULTS STATISTICS	

--- RULECHECK RESULTS STATISTICS (BY CELL)	

--- SUMMARY	

TOTAL CPU Time:	10434
TOTAL REAL Time:	10523
TOTAL Original Layer Geometries:	3772023 (111217971)
TOTAL DRC RuleChecks Executed:	2756
TOTAL DRC Results Generated:	0 (0)

Figure 11-1 The extracted result of DRC.

11.2. ANT

Here it is checked whether there exists gate destruction caused by wiring due to the antenna effect during manufacture process. The test environment is shown in Table 11-2. The result shows that no wiring causes gate destruction. Figure 11-2 is the extracted result of ANT test.

Table 11-2 ANT test environment

Tool (Version)	Calibre (v2008.3_16.12)
GDS File Name	MB8AW3801_10031701.gds
Top Cell Name	MB8AW3801
Rule File Name	MB8AW3801_ant.rul
Rule Version	r2.91

```

===== CALIBRE::DRC-H
SUMMARY REPORT
===
Execution Date/Time:      Wed Mar 17 13:46:42 2010
Calibre Version:         v2008.3_16.12   Tue Aug 19 13:58:10 PDT 2008
Rule File Pathname:      MB8AW3801_drc.rul
Rule File Title:
Layout System:           GDS
Layout Path(s):          MB8AW3801_10031701.gds
Layout Primary Cell:     MB8AW3801
Excluded Cells:

--- RULECHECK RESULTS STATISTICS

--- RULECHECK RESULTS STATISTICS (BY CELL)

--- SUMMARY
---
TOTAL CPU Time:          10434
TOTAL REAL Time:         10523
TOTAL Original Layer Geometries: 3772023 (111217971)
TOTAL DRC RuleChecks Executed: 2756
TOTAL DRC Results Generated: 0 (0)

```

Figure 11-2 The extracted result of the ANT test.

11.3. DFM

This is a layout test to improve the yield of chip manufacturing. The test environment is shown in Table 11-3. With the default setting submitted by Fujitsu Corp., there was no error. Figure 11-3 is the extracted results of DFM test.

Table 11-3 DFM test environment

Tool (Version)	Calibre (v2008.3_16.12)
GDS File Name	MB8AW3801_10031701.gds
Top Cell Name	MB8AW3801
Rule File Name	MB8AW3801_dfm.rul
Rule Version	r2.91

```
===== CALIBRE::DRC-H
SUMMARY REPORT
===
Execution Date/Time:      Wed Mar 17 13:46:42 2010
Calibre Version:         v2008.3_16.12   Tue Aug 19 13:58:10 PDT 2008
Rule File Pathname:      MB8AW3801_drc.rul
Rule File Title:
Layout System:           GDS
Layout Path(s):          MB8AW3801_10031701.gds
Layout Primary Cell:     MB8AW3801
Excluded Cells:

--- RULECHECK RESULTS STATISTICS

--- RULECHECK RESULTS STATISTICS (BY CELL)

--- SUMMARY
---
TOTAL CPU Time:           10434
TOTAL REAL Time:          10523
TOTAL Original Layer Geometries: 3772023 (111217971)
TOTAL DRC RuleChecks Executed: 2756
TOTAL DRC Results Generated: 0 (0)
```

Figure 11-3 The extracted result of the DFM test.

11.4. FL

FL test is a provisional DRC's item on Fujitsu 65nm CS202L process. The test environment is shown in Table 11-4. It is checked whether a special case occurs on the density of FL layer (diffusion layer). Since the error is 0, the layout is not a subject of above problems. Figure 11-4 is the extracted result of FL test.

Table 11-4 FL test environment

Tool (Version)	Calibre (v2008.3_16.12)
GDS File Name	MB8AW3801_10031701.gds
Top Cell Name	MB8AW3801
Rule File Name	J_SASEBO_ASIC_TOP.rul
Rule Version	r2.91

```
=====  
=== CALIBRE::DRC-H SUMMARY REPORT  
=====  
Execution Date/Time:      Wed Mar 17 17:22:06 2010  
Calibre Version:         v2008.3_16.12   Tue Aug 19 13:58:10 PDT 2008  
Rule File Pathname:      J_SASEBO_ASIC_TOP.rul  
Rule File Title:  
Layout System:           GDS  
Layout Path(s):          MB8AW3801_10031701.gds  
Layout Primary Cell:     MB8AW3801  
Excluded Cells:  
  
--- RULECHECK RESULTS STATISTICS  
--- RULECHECK RESULTS STATISTICS (BY CELL)  
--- SUMMARY  
---  
TOTAL CPU Time:          94  
TOTAL REAL Time:         340  
TOTAL Original Layer Geometries: 21084 (20374350)  
TOTAL DRC RuleChecks Executed: 1  
TOTAL DRC Results Generated: 0 (0)
```

Figure 11-4 The extracted result of the FL test.

11.5. LVS

From layout data (.GDS), the information on connections in transistor level is sampled, and it is checked whether it is consistent to the information on connections in netlist (.v). The test environment is shown in Table 11-5. From the result, we conclude that the layout data and the netlist are equivalent. Figure 11-5 is the extracted result of LVS test.

Table 11-5 LVS test environment

Tool (Version)	Calibre (v2008.3_16.12)
GDS File Name	MB8AW3801_10031701.gds
Netlist File Under Test	J_SASEBO_ASIC_TOP.lsv.v
Source CDL Name	J_SASEBO_ASIC_TOP.cdl
Top Cell Name of Layout Data	MB8AW3801
Top Cell Name of Source Data	J_SASEBO_ASIC_TOP
Rule File Name	MB8AW3801_lvs.rul
Rule Version	r2.91

```

REPORT FILE NAME:      MB8AW3801_lvs.sum
LAYOUT NAME:          MB8AW3801.layout_net.gz ('MB8AW3801')
SOURCE NAME:          J_SASEBO_ASIC_TOP.cdl ('J_SASEBO_ASIC_TOP')
RULE FILE:            MB8AW3801_lvs.rul
CREATION TIME:        Wed Mar 17 17:31:33 2010
CURRENT DIRECTORY:    backend/AIST/j_sasebo3_FUJITSU_65nm/Calibre/LVS
CALIBRE VERSION:      v2008.3_16.12 Tue Aug 19 13:58:10 PDT 2008

                                OVERALL COMPARISON RESULTS

                                #####
                                #                                     #
                                #          CORRECT          #
                                #                                     #
                                #####
                                #                                     #
                                #          *          *          #
                                #          |          #
                                #          ¥          #
                                #####
    
```

Figure 11-5 The extracted result of the LVS test.

12. The Summary of the Result of Each Test

The results of the tests are summarized as follows:

STA Test:	setup/hold	No Error
X-Talk (Noise) Test:	Noise Error	No Error
Power Test:	IR-Drop	0.8535% (STA is performed with running margin)
	VDD	0.3985%
	VSS	0.4456%
	Power Source Separation	No short circuit in each power source
Layout Test:	DRC	No Error
	ANT	No Error
	DFM	No Error
	FL	No Error
	LVS	The consistency between Netlist and layout is checked
Netlist Equivalence Test:	Equivalence Test	The equivalence between the final RTL and the final netlist is verified

Verification Result: No problem at all test.

13. CRYPTOGRAPHIC HARDWARE IPs

13.1. AES0 (Composite Field S-box)

Table 13-1 and Table 13-2 show the overview specifications of the AES cryptographic macro AES0 and the I/O ports of the macro, respectively. AES is one of the symmetric key block ciphers standardized by NIST, United States, and is also standardized as ISO/IEC 18033-3¹⁾. Refer to “FIPS PUB 197: ADVANCED ENCRYPTION STANDARD (AES)”²⁾ for further information on the algorithm. While the length of the user-definable part of the key is limited to 56 bits in the cryptographic LSI, the macro itself supports encryption and decryption with a 128-bit key.

Table 13-1 AES0 Overview Specifications

Algorithm	AES
Data block length	128 bits
Key length	128 bits
Functions	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	AES_Comp.v
Description language	Verilog-HDL
Top module	AES_Comp_ENC_top
S-box	Composite field $GF((2^2)^2)^2$ base
Throughput	128 bits / 10 clocks
Round key generation	On-the-fly

Table 13-2 AES0 I/O Ports

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 128-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge, and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the AES0 macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0.

AES0 is comprised of two circuit blocks, the encryption circuit and decryption circuit shown in Figure 13-1 and Figure 13-2, respectively. These circuits do not share a register or datapath. The S-box is implemented using a multiplication inversion circuit defined over the composite field $GF(((2^2)^2)^2)$.

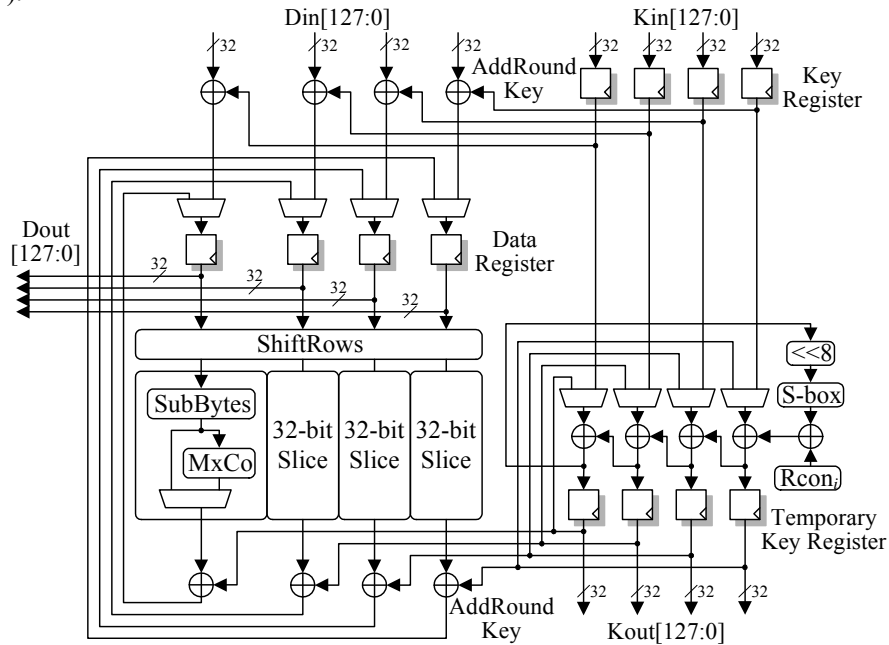


Figure 13-1 Encryption Datapath of AES0

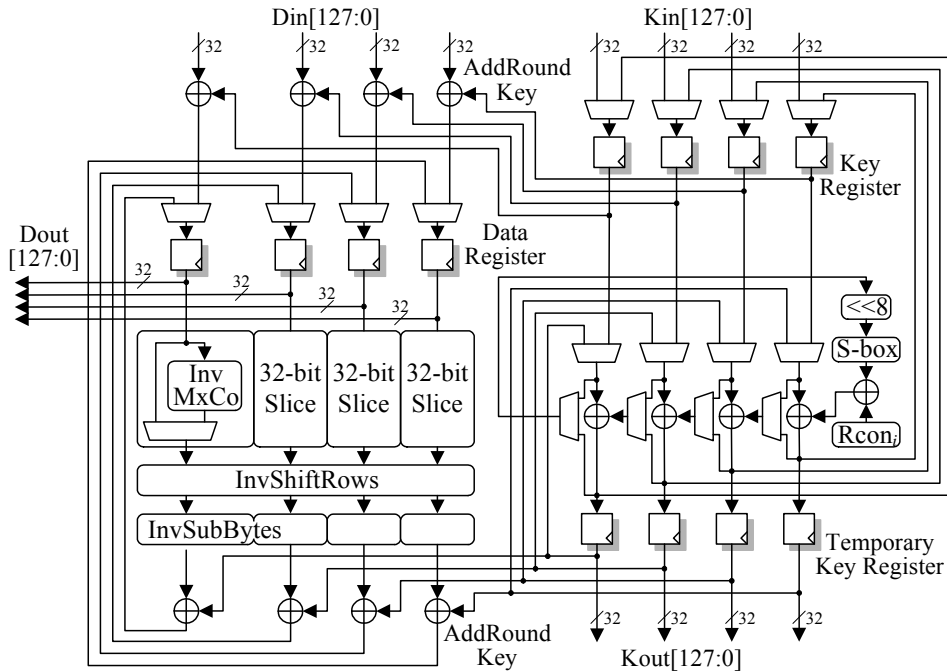


Figure 13-2 Decryption Datapath of AES0

Figure 13-3 presents the timing for encryption with the minimum possible cycles. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.

- CLK2:** Krdy=1 transfers the 128-bit secret key presented on Kin to the internal register.
- CLK3:** Although EncDec=0 indicates that the operation to run is encryption, initialization of the first round key for decryption (the last round key for encryption) starts in the decryption block, turning the busy signal BSY to 1. Note that the round key being initialized does not come out at Kout during round-key initialization because Kout is connected with the output of the encryption circuit.
- CLK14:** Round-key initialization completes. Kvld goes to 1 for one clock cycle, while BSY turns to 0. The internal register latches the 128-bit plaintext presented on Din.
- CLK15:** Encryption begins since EncDec=0, turning the busy signal BSY to 1. From this cycle on, Kout will be exporting the round keys forwarded from the Temporary Key Register every cycle.
- CLK16~25:** Encryption takes 10 clocks and completes at CLK24. Dout presents the 128-bit ciphertext, BSY falls to 0, and the data output signal Dvld turns to 1 for a single clock cycle at CLK25.

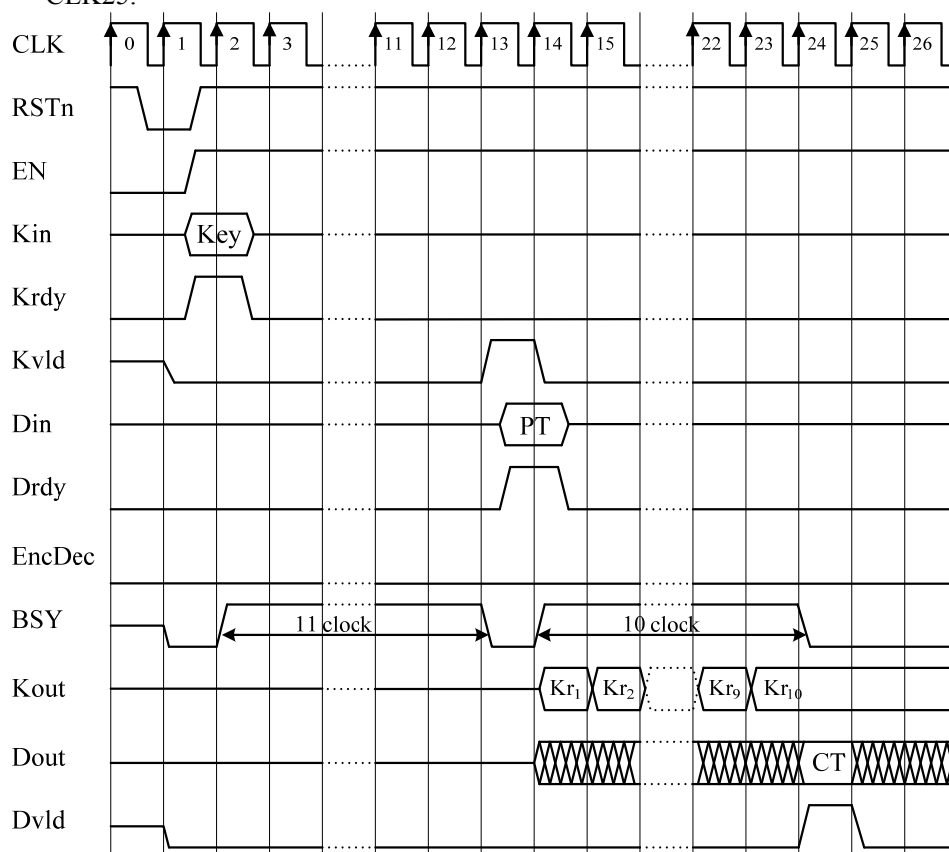


Figure 13-3 Timing Chart for Encryption on AES0

Figure 13-4 illustrates the timing for decryption with the minimum possible cycles. The operation(s) within each clock cycle follow:

- CLK1:** RSTn=0 resets the control circuit.
- CLK2:** Krdy=1 transfers the 128-bit secret key presented on Kin to the internal register.
- CLK3:** Initialization of the first round key for decryption (the last round key for encryption) starts, turning the busy signal BSY to 1.
- CLK14:** Round-key initialization completes. Kvld goes to 1 for one clock cycle, while BSY turns to 0. The internal register latches the 128-bit ciphertext presented on Din.
- CLK15:** Decryption begins since EncDec=1, turning the busy signal BSY to 1. From this cycle on, Kout will be presenting the round keys transferred from the Temporary Key Register every cycle.
- CLK16~25:** Decryption takes 10 clocks like encryption and completes at CLK24. Dout exports the 128-bit plaintext, BSY turns to 0, and the data output signal Dvld goes to 1 for a single clock.

cycle at CLK25.

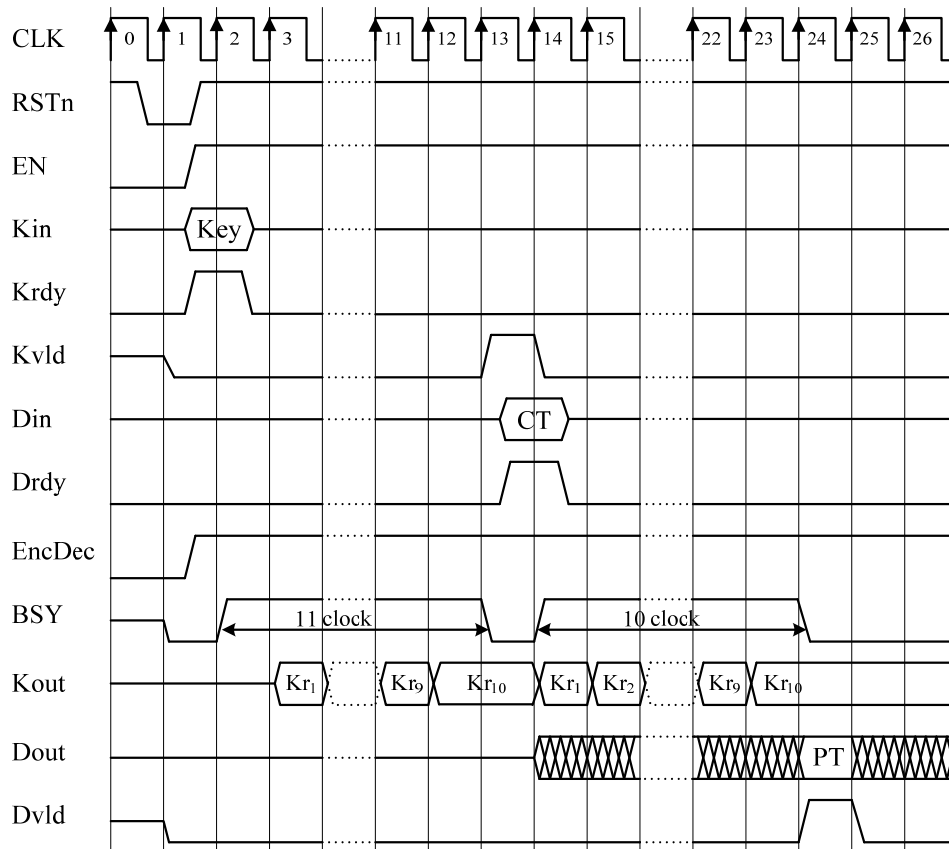


Figure 13-4 Timing Chart for Decryption on AES0

13.2. AES1/AES2/AES3/AES4 (Variety of S-boxes)

For the purpose of comparison evaluation of the dependency of side-channel attack resistance on the S-box, the AES cryptographic macros AES1, AES2, AES3, and AES4 are identical to one another except for their S-box structures. The AES1's S-box uses a look-up table. AES2 and AES3 employ the PPRM (Positive Polarity Reed-Muler) logic⁴. AES4 implements the multiplicative inverse circuit with a composite field³. These macros do not support decryption but only perform encryption. Accordingly, they have the same interface as that of AES0's with the exception of the encryption/decryption selector signal EncDec excluded. The overview specifications and I/O ports of these macros are shown in Table 13-4 and Table 13-5, respectively. Although they have the same datapath architecture as that of the AES0 encryption circuit shown in Figure 13-1, their timing shown in Figure 13-5 differs from AES0 since they do not need the round key initialization at the time of secret key entry on the decryption circuit.

Table 13-3 Overview Specifications of AES1, AES2, AES3, and AES4

Algorithm	AES
Data block length	128 bits
Key length	128 bits
Function	Encryption
Mode of operation	Electronic Code Book (ECB)

Source file	AES1: AES_TBL.v AES2: ASE_PPRM1.v AES3: AES_PPRM3.v AES4: AES_Comp.v
Description language	Verilog-HDL
Top module	AES1: AES_TBL AES2: ASE_PPRM1 AES3: AES_PPRM3 AES4: AES_Comp
S-box	AES1: Look-up Table AES2: PPRM1 AES3: PPRM3 AES4: Composite field $GF(((2^2)^2)^2)$
Throughput	128 bits / 10 clocks
Round key generation	On-the-fly

Table 13-4 I/O Ports of AES1, AES2, AES3, and AES4

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 128-bit plaintext given to Din is latched into the internal register on the rising clock edge, and encryption begins.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge at the clock signal CLK.
BSY	Out	1	During an active encryption or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption can be activated.
Dvld	Out	1	When encryption completes and the ciphertext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0.

Figure 13-5 shows the timing for encryption with the minimum possible cycles. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.

CLK2: Krdy=1 transfers the 128-bit secret key presented on Kin to the internal register.

CLK3: Round-key initialization completes. Kvld goes to 1 for one clock cycle, while BSY turns to 0. The internal register latches the 128-bit plaintext presented on Din.

CLK4: Encryption begins and the busy signal BSY turns to 1. From this cycle on, Kout will be exporting the round keys forwarded from the Temporary Key Register every cycle.

CLK5~14: Encryption takes 10 clocks and completes at CLK13. Dout presents the 128-bit ciphertext, BSY falls to 0, and the data output signal Dvld turns to 1 for a single clock cycle at CLK14.

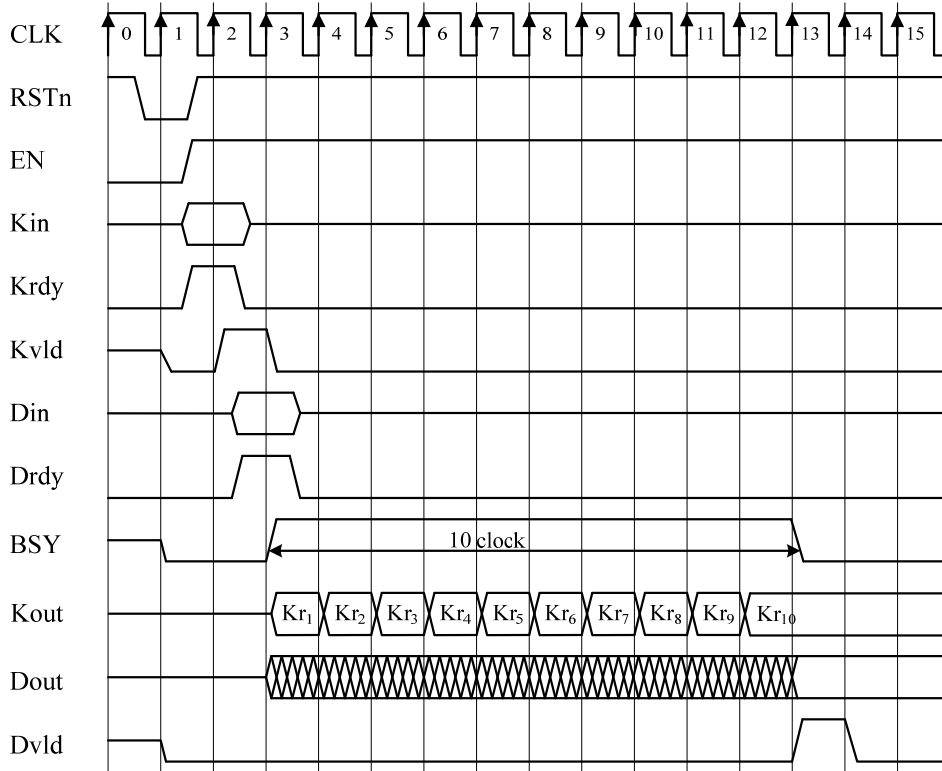


Figure 13-5 Timing Chart for AES1, AES2, AES3, and AES4

13.3. AES5 (CTR Mode)

The AES cryptographic macro AES5 supports the CTR mode of operation⁵⁾. It has a 4-stage pipeline to achieve fast operation. Table 13-5 and Table 13-6 show the overview specifications and I/O ports of AES5, respectively. Encryption and decryption are the same XOR operation with the same random number generated by the AES core, taking a plaintext or ciphertext as the input. Thus, the AES5 macro does not have the EncDec signal that switches between encryption and decryption as AES0 does.

Table 13-5 Overview Specifications of AES5

Algorithm	AES
Data block length	128 bits
Key length	128 bits
Functions	Encryption/Decryption
Mode of operation	Counter (CTR)
Source file	AES_CTR_Pipe_Comp.v
Description language	Verilog-HDL
Top module	AES
S-box	Composite field $GF(((2^2)^2)^2)$
Throughput	128 bits * 4 blocks / 46 clocks
Round key generation	On-the-fly

Table 13-6 I/O Ports of AES5

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1 and Drcv=1, a 128-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge for encryption or decryption process. As long as Drcv=1, data blocks can be fed continuously through Din at every clock cycle even if BSY=1.
CTRrdy	In	1	While BSY=0, turning CTRrdy to 1 initiates random number generation immediately regardless of the logic level at the encryption/decryption start signal START. The generated random number will be XORed with the plaintext or ciphertext in the internal register to output the resulting ciphertext or plaintext when START becomes 1.
START	In	1	After a sequential input of 4 data blocks of plaintext or ciphertext and a subsequent rise of START to 1, 4 random numbers are generated and XORed with the input data blocks resulting in ciphertext or plaintext in sequence. The next random number generation will immediately be ready to run so that the output data block will be available soon after the next input data block arrives. It is recommended to keep START=1 for the maximum throughput.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the AES5 macro.
CLK	In	1	Every internal register latches input data synchronously on the rising edge at the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0.
Drcv	Out	1	Only when the data input enable signal Drcv=1, an input data block of ciphertext or plaintext can be passed through Din.

Figure 13-6 shows the AES5 datapath that supports the CTR mode of operation with a pipeline. The S-box is implemented using a pipelined multiplication inversion circuit over the composite field $GF(((2^2)^2)^2)$. Both the randomization part shown on the left of the figure and the key scheduling part on the right have a 4-stage pipeline. The AES encryption part is used as a pseudo random number

generator. The generated random number is XORed with an input data block of plaintext or ciphertext, resulting in an output data block of ciphertext or plaintext. Accordingly, encryption and decryption compute each XOR operation using the same random number. The secret key for random number generation and the initial value of the counter go to the 128-bit key register Kreg and the 128-bit counter register CTRreg, respectively; 4 random numbers will be generated based on auto-incremented counter values (initially +0/+1/+2/+3). Even during random number generation, 4 blocks of plaintext or ciphertext are transferable through Din, buffered by the 4 128-bit data input registers RegDI0~RegDI3. Taking 4-block input after random number generation causes a slower throughput than the maximum throughput of $128 * 4 \text{ bits} / 46 \text{ clocks}$. Immediately after encryption or decryption of 4-block data finishes, the counter value will be incremented 4 times automatically for subsequent random number generation processes. To achieve the maximum throughput, a sequential 4-block feeding must be performed in 46 clock cycles on average.

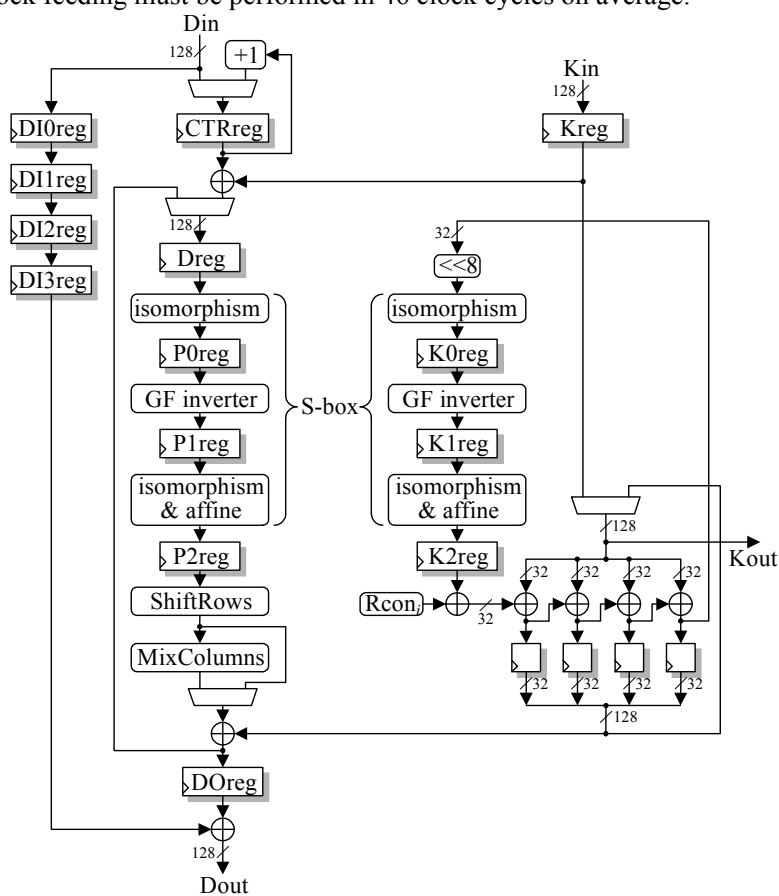


Figure 13-6 Datapath of AES5

Figure 13-7 shows the timing for encryption and decryption each with the minimum possible cycles. During these operations, the START signal is kept 1.

CLK1: RSTn=0 resets the control circuit.

CLK2: Krdy=1 transfers the 128-bit secret key Key presented on Kin to the key register Kreg.

CLK3: CTRrdy=1 transfers the 128-bit counter value Ctr presented on Din to the counter register CTRreg. The first round key Kr0 is identical to the secret key Key.

CLK4: Random number generation starts, raising the busy signal BSY to 1. Even though BSY=1, the data input enable signal Drcv=1 indicates that there is an empty slot in the data input registers and the empty register can latch a plaintext or ciphertext input. Drdy=1 transfers the plaintext block Pt0 presented on Din to the data input register. This plaintext will be encrypted and exported after random number generation completes.

CLK5~7: At the following 3 clocks, the 3 plaintexts Pt1, Pt2 and Pt3 are stored.

CLK8: Because all the 4 128-bit data input registers have captured plaintext blocks, the data input enable signal Drcv falls to 0 to indicate that there is no vacancy for further inputs. The key output port Kout exports the second round key Kr1. The round keys Kr2~Kr10 will follow every 4 clocks.

CLK46: Random number generation completes, turning BSY to 0. The data output port Dout shows the first 128-bit ciphertext data block Ct0. The data valid signal Dvld turns to 1.

CLK47~49: The next 3 ciphertext blocks Ct1, Ct2, and Ct3 come out in sequence. As can be seen, an output sequence is comprised of a set of 4 ciphertext blocks. Therefore, if only three plaintext blocks have entered in the data input registers, no ciphertext blocks will go out until the last plaintext block comes in. If the total number of input data blocks is not a multiple of 4, dummy input blocks are necessary to fill all the data input registers and to push out the last ciphertext blocks.

CLK50: After all the 4 plaintext blocks on the data input registers have been XORed with pseudo random numbers and the resulting ciphertext blocks have gone out, Dvld becomes 0. Immediately the next random number generation begins, and BSY turns to 1. With the data input enable signal Drcv=1, Drdy=1 stores the next plaintext block Pt4 provided on Din into the data input register.

CLK51: Although the plaintext block Pt4 is still kept on Din, it is not taken as the second data block at this cycle since Drdy=0.

CLK52, 53: With the state Drdy=1 the two plaintext blocks Pt5 and Pt6 are taken into the data input registers.

CLK54: Since Drdy=0, the plaintext block is not stored.

CLK55: Since Drdy=1, the plaintext block Pt7 is stored.

CLK56: Since the subsequent 4 plaintext blocks have been stored into the data input register, Drcv falls to 0.

CLK92~95: After the last ciphertext export during CLK46~CLK49, 46 clock cycles later (the earliest possible cycle) the 4 blocks of ciphertext Ct4~Ct7 begin coming out continuously.

CLK96: Even though Drcv=1, data import does not take place during this cycle.

CLK137: Random number generation completes and BSY turns to 0. Because no plaintext or ciphertext blocks have been taken, no corresponding ciphertext or plaintext blocks exist. Note that a new key and counter value can be set only when BSY=0. In order to set a new key and counter value without waiting for the operation to complete and BSY=0, the whole macro has to be reset by RSTn=0. At this CLK137, to decrypt the ciphertext blocks Ct0~Ct3, the same counter value is set as was done at CLK3. Because no new keys have been set, the key provided at CLK2 is used.

CLK138: Since Drcv=1, the first ciphertext block Ct0 is taken.

CLK139: Random number generation begins and BSY rises to 1. The second ciphertext block Ct1 is taken.

CLK140: The third ciphertext block Ct2 is taken.

CLK179: Random number generation has just finished and BSY becomes 0. Since only 3 ciphertext blocks have been entered, exporting plaintexts has not yet begun.

CLK180: The fourth ciphertext block Ct3 is taken. As all the 4 data input registers DI0reg~DI3reg have been filled, Drcv falls to 0, disabling further data entries.

CLK181~184: Dvld becomes 1. The 4 plaintext blocks Pt0~Pt3 come out continuously in sequence.

CLK185: Since all the data input registers become empty, Drcv rises to 1. The next random number generation begins, raising BSY to 1.

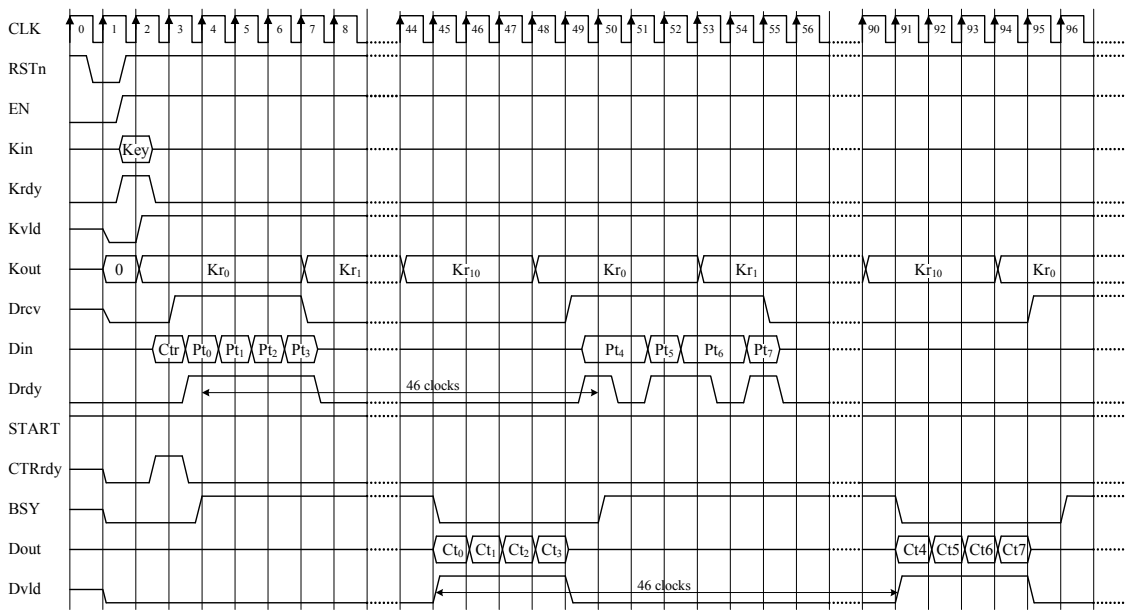


Figure 13-7-1 Timing Chart for AES5 Encryption/Decryption

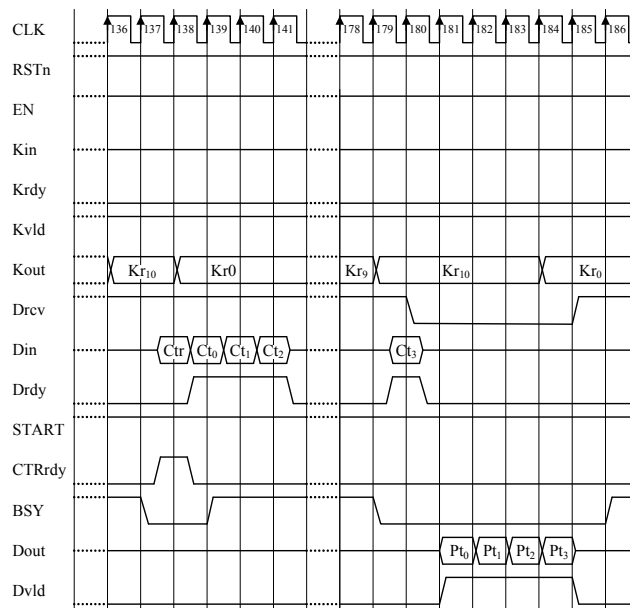


Figure 13-7-2 Timing Chart for AES5 Encryption/Decryption

Figure 13-8 shows the timing for AES5 performing encryption and decryption with a START signal control. If the START signal is fixed to 1 as illustrated in Figure 13-7, the macro computes XOR operations between the random numbers generated by the AES core and the 4 plaintext or ciphertext blocks that have been entered, and subsequently exports the resulting ciphertext and plaintext blocks. At the same time the next random number generation is initiated automatically in the AES core. However, for measurement of power traces or electro-magnetic waveforms in a side-channel attack experiment, the operation timing of the AES core needs to be controlled by the experimental system. The START signal is implemented for that purpose.

CLK1: RSTn=0 resets the control circuit.

CLK2: Krdy=1 transfers the 128-bit secret key Key presented on Kin to the key register Kreg.

CLK3: CTRrdy=1 transfers the 128-bit counter value Ctr presented on Din to the counter register CTRreg. The first round key Kr0 is identical to the secret key Key.

- CLK4:** Random number generation starts, raising the busy signal BSY to 1. Even though the data input enable signal Drcv=1, a plaintext block is not taken at this clock cycle. This differs from the version in Figure 13-7 where the START signal remains 1.
- CLK46:** Random number generation completes, turning BSY to 0. The first 128-bit plaintext block Pt0 is taken. The AES core will remain in an idle state until all the 4 plaintext blocks have entered. Although START turns to 1, it is not effective since the required 4 plaintexts have not yet been entered.
- CLK48,49:** The second and third plaintexts Pt1 and Pt2 are stored.
- CLK51:** The third plaintext Pt3 is stored.
- CLK52:** Because all 4 128-bit data input registers have captured plaintext blocks, the data input enable signal Drcv falls to 0.
- CLK53~56:** The ciphertext blocks Ct0~Ct3 corresponding to the 4 plaintext blocks Pt0~Pt3 come out in sequence.
- CLK56:** In preparation for the next random number generation, the round key is reset to Kr0 from Kr10.
- CLK99:** In preparation for decryption, the counter register is reset to the initial value Ctr during BSY=0.
- CLK100:** The round-key output port Kout indicates Kr0.
- CLK101:** Random number generation begins and BSY rises to 1.
- CLK142:** Random number generation has just finished and BSY becomes 0. Since no plaintext has been entered, the AES core enters an idle state.
- CLK144~147:** With Drdy set to 1, the 4 ciphertext blocks Ct0~Ct3 are taken.
- CLK148:** Since all the data input registers have been filled with ciphertext blocks, Drcv turns to 0.
- CLK149~152:** The 4 plaintext blocks Pt0~Pt3 come out continuously in sequence.
- CLK152:** In preparation for the next random number generation, the round-key register is reset to Kr0. Since START=0, the random number generation has not yet begun.
- CLK153:** Since all the 4 plaintext blocks have been exported and all the data input registers are empty, Drcv rises to 1.
- CLK154:** START=1 initiates random number generation.
- CLK156:** Random number generation begins and BSY rises to 1.

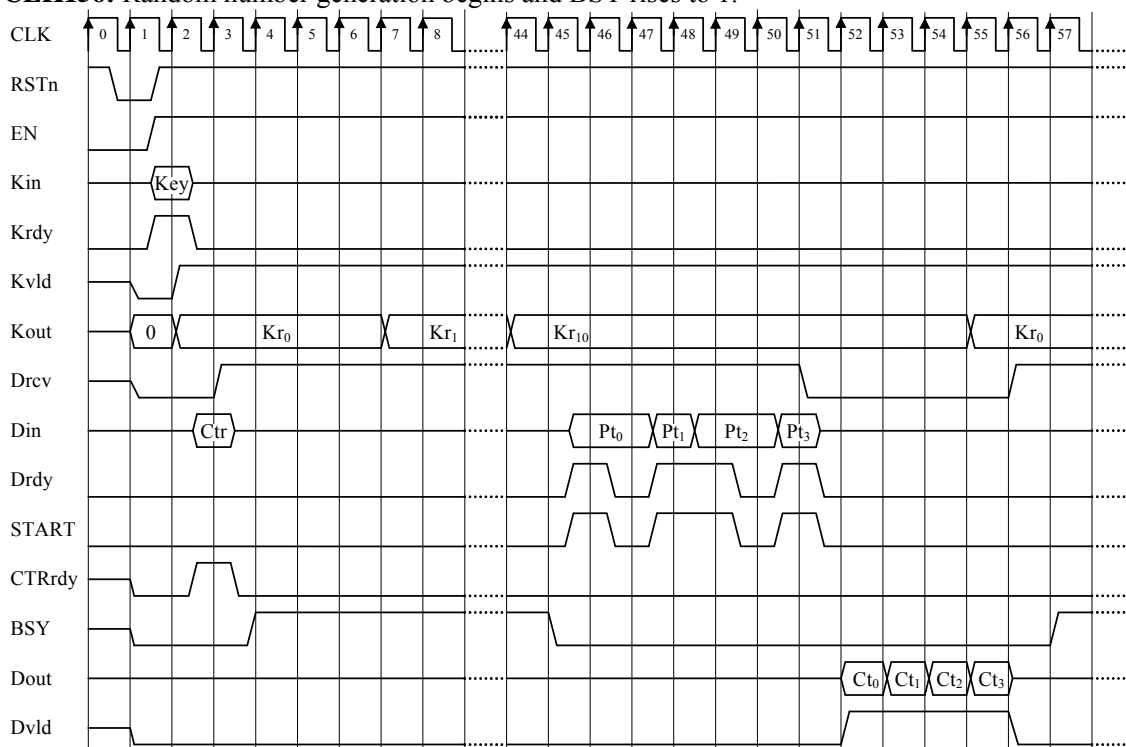


Figure 13-8-1 Timing Chart for AES5 with START Signal Control

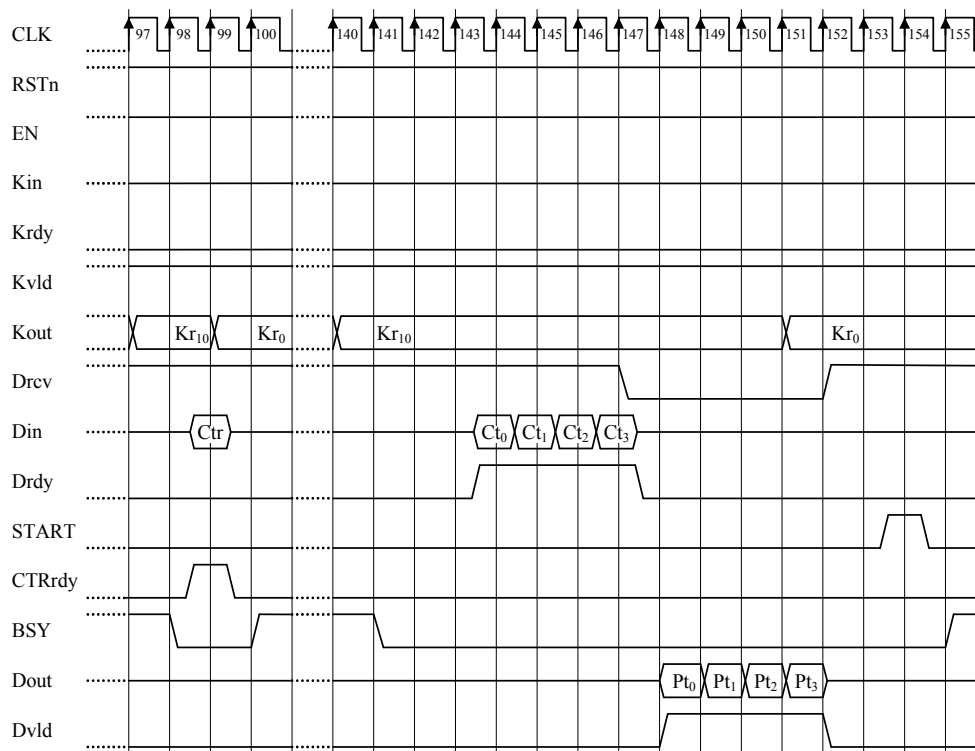


Figure 13-8-2 Timing Chart for AES5 with START Signal Control

13.4. AES6 (FA Countermeasure)

AES6 is a cryptographic circuit macro, which employs a countermeasure against Fault Injection Attacks (FA). Overview specifications and I/O ports of AES6 are shown in Table 13-7 and Table 13-8, respectively. The AES6 macro checks the intermediate value in encryption or decryption every half round by doing the following: it holds the intermediate value; a half round later, it performs decryption or encryption inversely on the intermediate value, and inspects whether the resulting plaintext or ciphertext is identical to the one a half round before. The key data is also checked to determine whether an error occurred on the final round key.

Table 13-7 Overview Specifications of AES6

Algorithm	AES
Data block length	128 bits
Key length	128 bits
Functions	Encryption/Decryption, Error detection
Mode of operation	Electronic Code Book (ECB)
Source file	AES_FA.v
Description language	Verilog-HDL
Top module	AES
S-box	Composite field $GF(((2^2)^2)^2)$ base
Throughput	128 bits / 21 clocks
Round key generation	On-the-fly

Table 13-8 I/O Ports of AES6

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 128-bit plaintext or ciphertext block given to Din is latched into the internal register on the rising clock edge, and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the AES6 macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 for one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0.
Err	Out	2	Err[0]=0: No data errors occurred. =1: Data error(s) occurred. Err[1]=0: No key errors occurred. =1 Key error(s) occurred

Figure 13-9 shows a typical architecture of AES, some of whose components, such as the inversion circuit over $GF(2^8)$ in the S-box and the common terms in the matrix functions MixColumns and InvMixColumns, are shared by the encryption and decryption parts. For such component sharing, the order of AddRoundKey and InvMixColumns (represented by InvMixCol in the figure) typically switches for decryption. To compensate for the reordering, the key scheduling part shown on the right side has additional MixColumns. However, as explained later, the AES6 macro does not employ such function reordering.

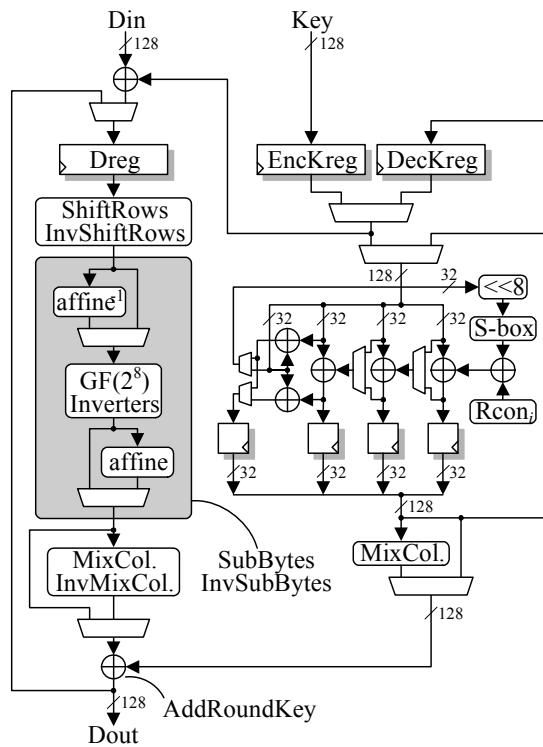


Figure 13-9 Typical Datapath Architecture of AES with the Core Components Shared by Encryption and Decryption

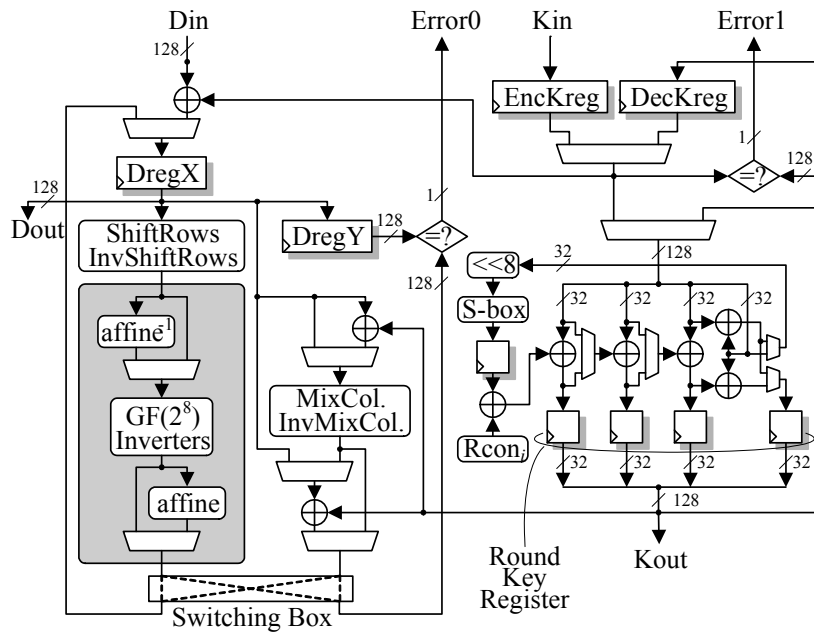


Figure 13-10 Encryption and Decryption Circuit of AES6

Figure 13-10 demonstrates the datapath of the AES6 macro. The encryption and decryption functions share some datapaths. The macro divides the sequence of the round functions of AES into two. One of them performs encryption or decryption, while the other bears the counterpart function for error detection. Unlike the architecture shown in Figure 13-9, the macro does not reorder AddRoundKey and InvMixColumns to share the XOR gates. Sharing XOR gates could shorten the critical path of

the round function block, but MixColumns would be necessary in the key scheduler instead. In contrast, the method that AES6 employs, where the round functions are divided into two sequences, takes advantage of a better trade-off between the circuit size and operation speed achieved by not incorporating additional MixColumns, instead of sharing the XOR gates. In addition to dividing the round function block, the key scheduling part is also divided into two parts to avoid being the critical path. With registers inserted between the divided parts, one round takes two clock cycles. Even if the round functions operate correctly, an error could occur at the key scheduler, or the control counter's failure could make the round loop complete at the 1st round instead of 10-time round repetition. To prevent such problems, the macro tests the sameness of the key generated on-the-fly with the stored key in the decryption key register DecKreg for encryption, or in the encryption key register EncKreg for decryption, at the completion of the last round. This practically ensures that the attacker will not be able to cheat the test for internal 128-bit data (unknown to him) in the key scheduler, even if he could skip the counter value.

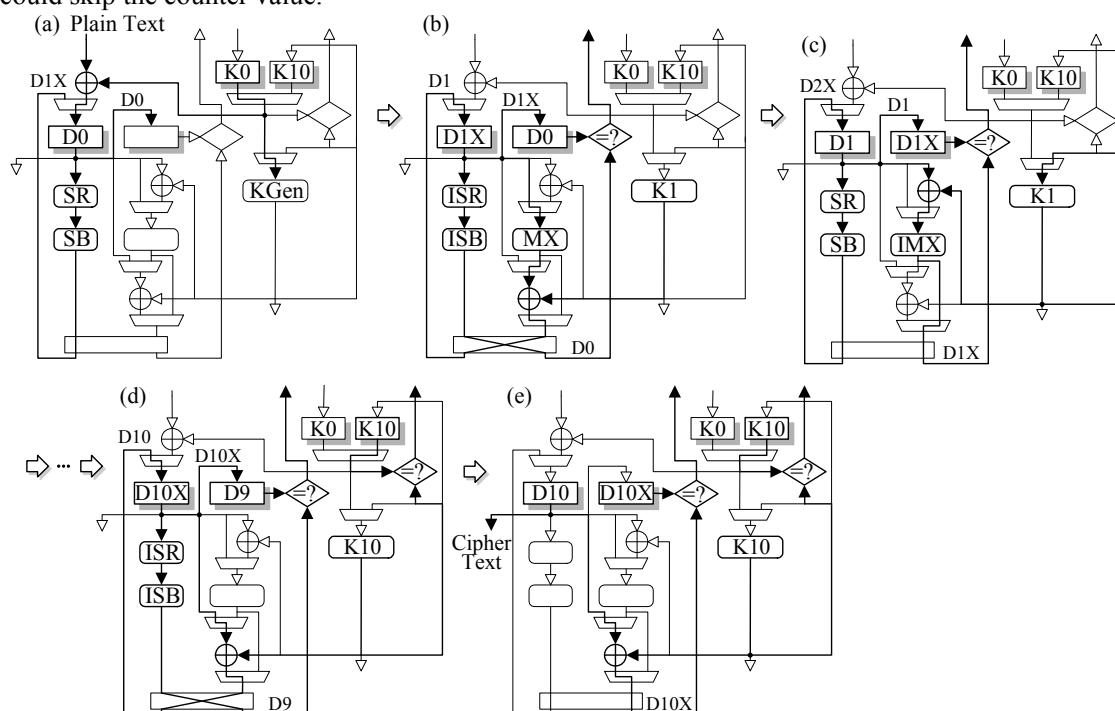


Figure 13-11 Encryption Operation Examples of AES6

Figure 13-11 illustrates an encryption operation of AES6 by example. Assume that the initial encryption key K0 written into EncKreg has been transformed into the initial decryption key (= final encryption key) K10 in the key scheduler, and K10 has been loaded into DecKreg already. In (a), initially the given plaintext is XORed with K0, resulting in D0 going into the register DregX. Subsequently the data goes through the path of ShiftRows and SubBytes in the first half round, being the feedback data D1X. Meanwhile D0 is written into the register DregY. The key scheduler generates the first round key K1 from the initial key K0 on-the-fly. In (b), the circuit decrypts the feedback data for verification through the datapath used for encryption in (a), and performs the last half round operation. The verification inversely transforms (i.e. decrypts) D1X latched by DregX, by using InvShiftRows and InvSubBytes, and compares the decrypted data with D0 kept in DregY. On the other hand, D1X also goes through the other path containing MixColumns and AddRoundKey, which XORs the falling through data with the round key K1, and transforms it into D1. In (c), D1, the latched value at DregX, transforms into D2X like (a). At the same time, D1 goes to the next path on the right and becomes back to D1X as the result of an inverse transform with InvMixColumns and the XOR in the path. The comparator verifies the result, which is expected to be the same as the value stored in DregY. These processes of encryption and verification will repeat until the 9th round. In (d), D10X goes through InvShiftRows and InvSubBytes for error detection. D10X is also XORed

with the 10th round key K10 to produce D10 for the last stage of the whole encryption. Since the last round of AES does not have MixColumns, its function block is bypassed in the path. Since this is the last round, completion of 10-round operations is verified by comparing the on-the-fly generated key K10 in the round key register with the pre-calculated key K10 in the EncKreg register. Although the ciphertext D10 could be output at this time, in practice it goes out after verifying that the inverse transform result matched with D10X as shown in (e). Because the next plaintext will not be input until the final verification completes, the entire encryption for a single block takes 21 clock cycles, broken down into 10 rounds x 2 cycles and 1 cycle for (e).

13.5. AES7 (Round Key Pre-calculation)

The AES cryptographic macro AES7 differs from the other AES macros that generate the round keys on-the-fly in that it calculates the round keys and stores them into 11 128-bit registers in advance.

Table 13-9 Overview Specifications of AES7

Algorithm	AES
Data block length	128 bits
Key length	128 bits
Function	Encryption
Mode of operation	Electronic Code Book (ECB)
Source file	AES_PreKeyGen.v
Description language	Verilog-HDL
Top module	AES_PKG
S-box	Composite field $GF(((2^2)^2)^2)$
Throughput	128 bits / 10 clocks
Round key generation	Pre-calculation

Table 13-10 I/O Ports of AES7

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a secret key is latched into the internal register, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a plaintext block is latched into the internal register, and encryption begins.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the AES7 macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 for one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption can be activated.
Dvld	Out	1	When encryption completes and the ciphertext is set on the data output port Dout, Dvld goes to 1 for one clock cycle and returns to 0.

Figure 13-12 represents the datapath architecture of AES7, which has 11 128-bit registers to store the round keys in addition to the same encryption circuit as that of AES0 shown in Figure 13-1. The entry of a secret key at K_{in} initiates key scheduling and the calculated round keys will be stored in the registers. When performing encryption, these registers supply $AddRoundKey$ with the round keys, without on-the-fly key scheduling.

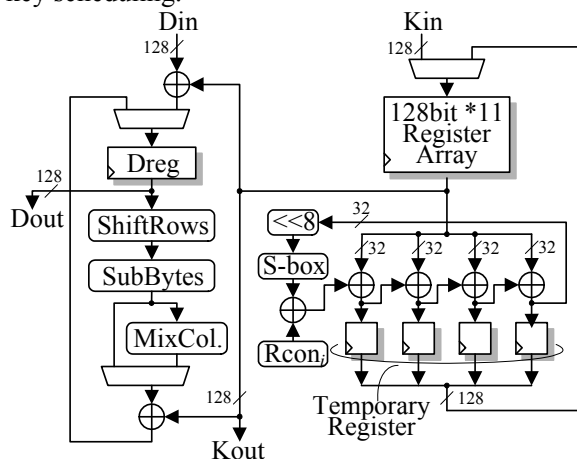


Figure 13-12 Datapath Architecture of AES7

The encryption timing for AES7 with the minimum possible cycles is shown in Figure 13-13. The operation(s) within each clock cycle follow:

- CLK1:** $RST_n=0$ resets the control circuit.
- CLK2:** $Krdy=1$ transfers the 128-bit secret key Key presented on K_{in} to the internal register.
- CLK3:** Key scheduling starts, turning the busy signal BSY to 1. $Krdy$ returns to 0.
- CLK14:** Key scheduling completes. The $Kvld$ flag goes to 1 to indicate the keys have become valid, while BSY turns to 0.
- CLK15:** From this cycle on, plaintext blocks can enter the circuit to encrypt. The plaintext Pt_0 presented on D_{in} is XORed with the first round key Kr_0 (This is the secret key Key input through K_{in} .) output on the key registers. $Drdy=1$ loads the result of XOR into the data register D_{reg} . The 128-bit port K_{out} outputs Kr_0 .
- CLK16:** Encryption begins, turning the busy signal BSY to 1. From this cycle on, K_{out} will be exporting the round keys every clock cycle starting with the second round key Kr_1 . Likewise, D_{out} outputs the intermediate values forwarded from D_{reg} . Thus, during the whole encryption process, the round keys and intermediate values are output every clock cycle.
- CLK17~26:** Encryption takes 10 clocks and completes at CLK25. D_{out} presents the ciphertext, BSY falls to 0, and $Dvld$ turns to 1 at CLK26. The next plaintext Pt_1 can be input at CLK26.

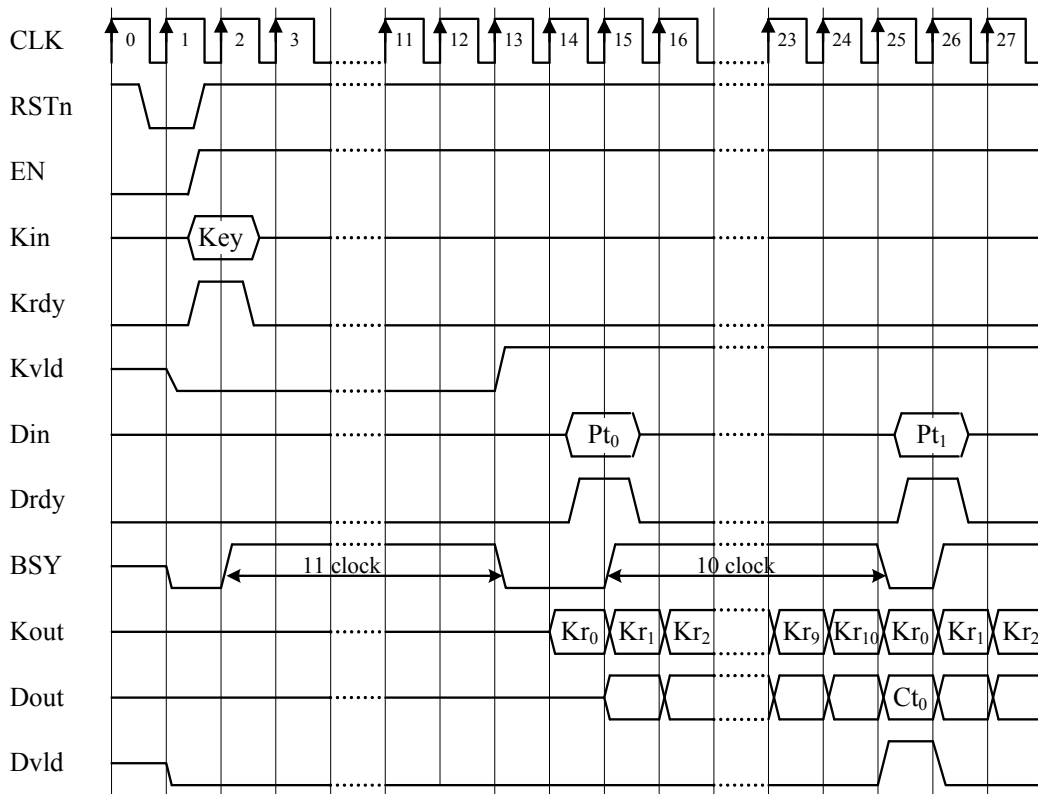


Figure 13-13 Timing Chart for Encryption on AES7

13.6. AES8 (MAO)

The AES8 macro implements the Masked-AND Operation (MAO)⁶⁾, the DPA countermeasure with randomized masking on AND operations proposed by Trichina et al. Figure 13-14 illustrates the basic structure of a Masked-AND gate. The original input data $\langle a, b \rangle$ are XOR-masked with random numbers $\langle m_a, m_b \rangle$ that are independent with each other, resulting in the gate inputs $\langle \tilde{a}, \tilde{b} \rangle$. The gate outputs $(a \cdot b) \oplus m$, which is the logical product of a and b masked with another independent random number input m . This operation does not involve the inputs of original operation $\langle a, b \rangle$ or output $a \cdot b$. However, it has been reported that the gate risks secret information leaking through power consumption due to glitches caused by signal delay variations.

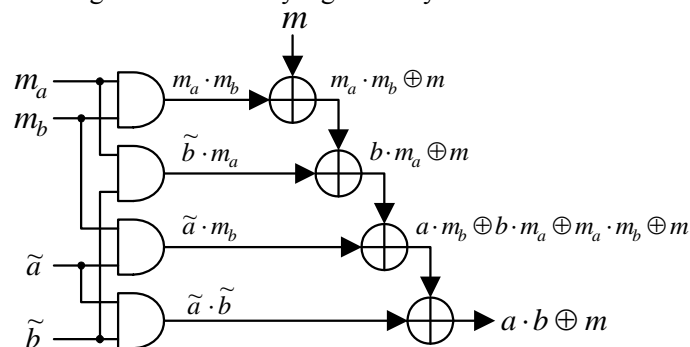


Figure 13-14 Masked-AND Gate

13.7. AES9 (MDPL)

AES9 adopts the DPA countermeasure proposed by Popp et al. It implements the Masked Dual-rail Precharge Logic (MDPL)⁸ that combines the after-mentioned WDDL⁹ with random number masking. Figure 13-15 shows the main components and the basic construction of the MDPL. Figure 13-15(a) shows a MAJ gate, the majority decision logic that outputs the logic level 0 or 1 depending which level is represented at more input ports. The MDPL-AND gate shown in (b) has two complementarily placed MAJ gates so that it performs the operation of the formulas shown below

for the masked inputs a_m, b_m , the mask m , and its inversion. The truth table for the MDPL-AND gate is shown in Table 13-11.

$$\begin{cases} q_m = MAJ(a_m, b_m, m) = MAJ(a \oplus m, b \oplus m, m) = a \cdot b \oplus m \\ \bar{q}_m = MAJ(\bar{a}_m, \bar{b}_m, \bar{m}) = MAJ(a \oplus \bar{m}, b \oplus \bar{m}, \bar{m}) = a \cdot b \oplus \bar{m} \end{cases}$$

The WDDL requires the capacitances of complementary wires be identical. On the contrary, the MDPL equalizes the power consumption regardless of the capacitance balance of complementary wires because the output of a MAJ gate transitions at random depending on the random number m (and \bar{m}) as shown in Figure 13-15(c). However, even though the MPDL yields less information leak than the WDDL does, it has been pointed out that the countermeasure is not able to completely prevent information leakage.

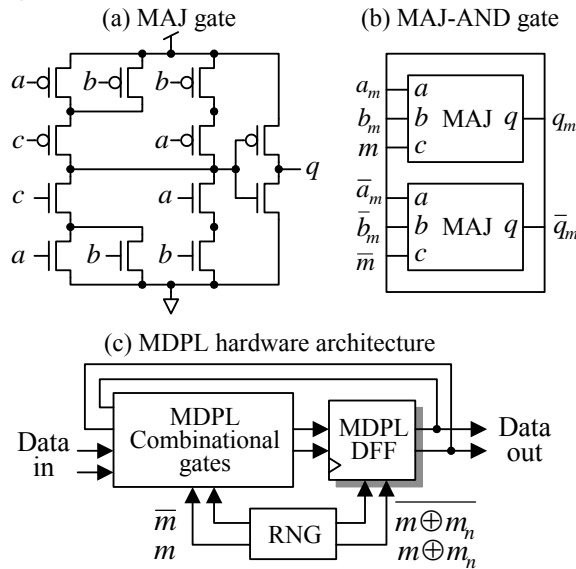


Figure 13-15 Masked Dual-rail Precharge Logic

Table 13-11 Truth Table for the MDPL-AND Gate

a	b	m	a_m	b_m	q_m	\bar{m}	\bar{a}_m	\bar{b}_m	\bar{q}_m
0	0	0	0	0	0	1	1	1	1
0	0	1	1	1	1	0	0	0	0
0	1	0	0	1	0	1	1	0	1
0	1	1	1	0	1	0	0	1	0
1	0	0	1	0	0	1	0	1	1
1	0	1	0	1	1	0	1	0	0
1	1	0	1	1	1	1	0	0	0
1	1	1	0	0	0	0	1	1	1

13.8. AES10 (Threshold Implementation)

AES10 employs the Threshold implementations⁷⁾, the DPA countermeasure proposed by Nikova et al that makes use of a plurality of random number masks. In this section, \oplus and \bigoplus denote

$$x = \bigoplus_{i=1}^n x_i$$

addition and summation over $\text{GF}(2^m)$, respectively. The input variables are represented as

$$\text{and } y = \bigoplus_{i=1}^n y_i, \text{ while the output variable is } z = \bigoplus_{i=1}^n z_i.$$

$$\begin{cases} z_1 = (x_3 \oplus x_4)(y_2 \oplus y_3) \oplus y_2 \oplus y_3 \oplus y_4 \oplus x_2 \oplus x_3 \oplus x_4 \\ z_2 = (x_1 \oplus x_3)(y_1 \oplus y_4) \oplus y_1 \oplus y_3 \oplus y_4 \oplus x_1 \oplus x_3 \oplus x_4 \\ z_3 = (x_2 \oplus x_4)(y_1 \oplus y_4) \oplus y_2 \oplus x_2 \\ z_4 = (x_1 \oplus x_2)(y_2 \oplus y_3) \oplus y_1 \oplus x_1 \end{cases}$$

These fundamental element formulas satisfy the following:

1. Every function is independent of at least one element (x_n, x_n) for each of the input variables x and y .

$$z_n = f(x_1, x_2, \dots, x_{n-1}, y_1, y_2, \dots, y_{n-1},)$$

2. The sum of the output elements gives the original output.

$$z = \bigoplus_{i=1}^n z_i = \bigoplus_{i=1}^n f_i(\dots) = N(x)$$

3. If $z = N(x, y, \dots)$ can be realized for all distributions of inputs x, y , the following is constant:

$$\Pr(\bar{z} = \bar{Z} \mid z = \bigoplus_{i=1}^n Z_i)$$

Thus, firstly, the input variables are not correlated with z_i . In other words, the operations are independent of the input and output variables. Secondly, since the above property 3 indicates that the transition probability of each function output for each element is constant, the power consumption for every cycle turns out to be constant. Therefore, this suggests that the countermeasure is promising against DPA because even if the power consumption due to glitches is captured, it is considered that secret information would not leak.

13.9. AES11 (WDDL)

AES11 implements Wave Dynamic Differential Logic (WDDL)⁹⁾, the DPA countermeasure proposed by Tiri et al. Figure 13-16 shows the basic structural element of WDDL, which applies the Sense Amplifier Based Logic (SABL), a dual-rail logic, to make the power consumption due to gate switching constant. While the precharge signal of the data input logic is 1, all the input data for the combinational logic stay 0s and the circuit is in the idle state. When the precharge signal turns to 0, the complementary input data (0, 1) or (1, 0) are sent into the combinational logic through the data input logic, and the operation starts. This method is considered to be effective against power analysis attacks because the switching count over the whole combinational logic does not depend on the input data and consequently the power consumption is constant. However, to be exact, there is a difference in the power consumptions of the AND gate and OR gate. In addition, the wiring capacitances of a pair of data lines have to be adjusted. These factors suggest that the input and output delay variations of the WDDL gates would cause the leakage of secret information.

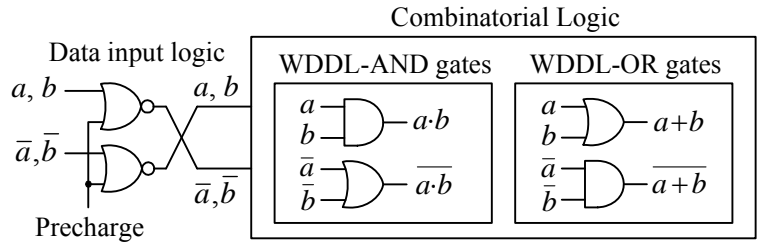


Figure 13-16 Wave Dynamic Differential Logic

13.10. AES12/AES13 (Pseudo RSL)

Random Switching Logic (RSL)¹⁰ is the transistor-level DPA countermeasure proposed by Mitsubishi Electric Corporation, which uses majority decision logic gates with output enable signals. Figure 13-17 shows a NAND gate with RSL. In a simple random masking countermeasure that does not take signal delays into account, transient transitions may leak information. On the contrary, the RSL gate prevents a transient transition by controlling the delays of the inputs (x_z, y_z), output enable \overline{en} , and random mask (r_z). Re-masking at every RSL gate makes it possible to resist even higher-order DPA or the like. The following illustrates the processes on the RSA-NAND gate:

$$\text{Input: } \overline{en}, \begin{cases} x = a \oplus r_x \\ y = b \oplus r_y \end{cases}, \begin{cases} r_z \\ r_{xz} = r_x \oplus r_z \\ r_{yz} = r_y \oplus r_z \end{cases} \quad \text{Output: } \overline{a \cdot b} \oplus r_z$$

Process 1: $\overline{en} = 1$ (Suppresses transient transition)

Process 2: $\begin{cases} x_z = x \oplus r_{xz} (= a \oplus r_z) & \text{(Remasks } x) \\ y_z = y \oplus r_{yz} (= b \oplus r_z) & \text{(Remasks } y) \end{cases}$

Process 3: RSL-NAND($x_z, y_z, r_z, \overline{en}$) (Applies the input data to the RSL-NAND gate)

Process 4 $\overline{en} = 0$ (Enables the output after stabilizing data)

While the RSL requires a dedicated cell library, the pseudo RSL, employed in the AES12 and AES13 macros, emulates the operations of the RSL gates with standard CMOS libraries. Figure 13-18 depicts a pseudo RSL-NAND gate utilizing a multi-input AND-OR gate as a majority decision logic. The NOR gate at the last stage controls the output to prevent a transient transition event from propagating out of the pseudo RSL gate.

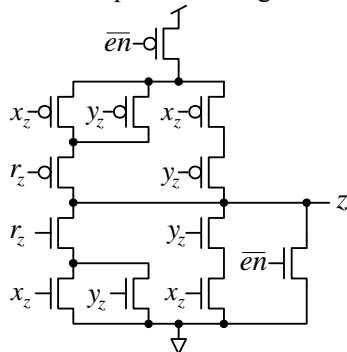


Figure 13-17 RSL-NAND Gate

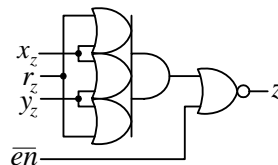


Figure 13-18 Pseudo RSL-NAND Gate

13.11. Camellia

The overview specifications and I/O ports of the cryptographic circuit macro for Camellia¹¹ are shown in Table 13-12 and Table 13-12, respectively. Camellia is a block cipher that has the Feistel structure and thus requires more cycles than AES. However, because Camellia can use the same

datapath for both encryption and decryption, it is better suited for a compact implementation than AES with its SPN structure.

Table 13-12 The Overview Specifications of Camellia

Algorithm	Camellia
Data block length	128 bits
Key length	128 bits
Function	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	Camellia.v
Description language	Verilog-HDL
Top module	Camellia
S-box	Table implementation
Throughput	128 bits / 23 clocks
Round key generation	Pre-calculation and On-the-fly

Table 13-13 I/O Ports of Camellia

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 128-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the Camellia macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0.

Figure 13-19 represents the datapath architecture of the Camellia macro. Each single round is processed in a single clock cycle; Encryption for a 128-bit plaintext and decryption for a 128-bit ciphertext each take 23 clock cycles. The secret key is latched into the key register K1, and processed for initial conversion at the data randomization part in the bottom of the figure. The converted key is eventually stored in the Ka register. The round keys are generated based on the Ka and K1 registers' data on-the-fly.

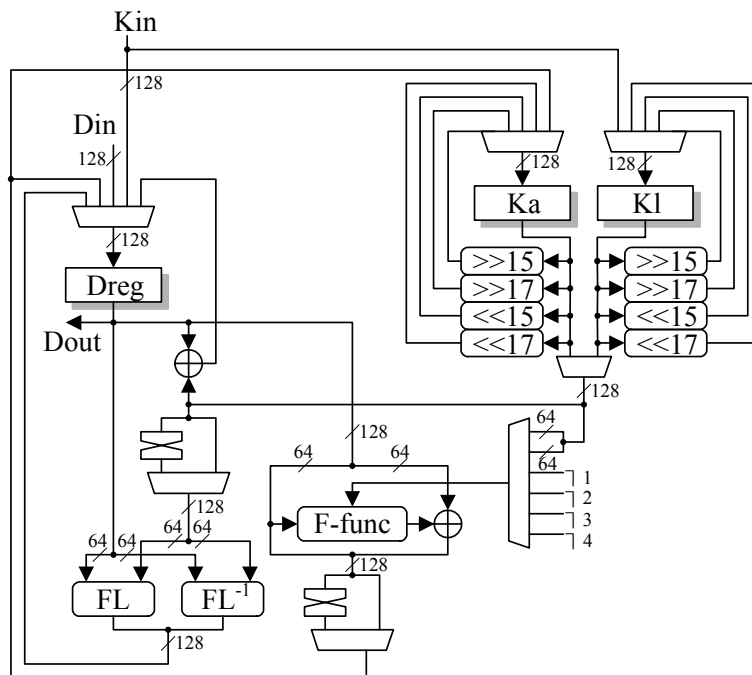


Figure 13-19 Datapath Architecture of Camellia

Figure 13-20 and Figure 13-21 illustrate the timings for key scheduling and encryption for Camellia each with the minimum possible cycles, respectively.

- CLK1:** $RST_n=0$ resets the control circuit.
- CLK2:** $Kr_{dy}=1$ transfers the 128-bit secret key presented on K_{in} to the internal register.
- CLK3:** Key scheduling starts, turning the busy signal BSY to 1 and Kr_{dy} to 0.
- CLK8:** Key scheduling completes in 8 clock cycles. The $Kvld$ flag goes to 1 to indicate the initial key has become valid, while BSY turns to 0.
- CLK9:** From this cycle on, plaintext or ciphertext blocks can enter the circuit. With $EncDec=0$ for encryption, $Dr_{dy}=1$ loads the plaintext presented on the 128-bit input port D_{in} into the data register D_{reg} .
- CLK10:** Encryption begins, turning the busy signal BSY to 1. From this cycle on, K_{out} will be exporting the round keys every clock cycle starting with the round keys Kw_1 and Kw_2 . Likewise, D_{out} outputs the intermediate values forwarded from D_{reg} . Thus, during the whole encryption process, the round keys and intermediate values are output every clock cycle.
- CLK32:** Encryption completes in 23 clock cycles. D_{out} presents the ciphertext and BSY falls to 0. $Dvld$ turns to 1 at this clock and returns to 0 at the next clock.
- CLK33:** $Dr_{dy}=1$ initiates the next operation. At this clock, with $EncDec=1$ for decryption, the 128-bit port D_{in} latches the ciphertext.
- CLK34:** Decryption begins, turning the busy signal BSY to 1. Similarly to encryption, K_{out} and D_{out} output the round keys and intermediate values every clock cycle, respectively.
- CLK57:** Decryption finishes in 23 clock cycles. D_{out} outputs the plaintext and BSY turns to 0. $Dvld$ turns 1 for a single clock cycle.

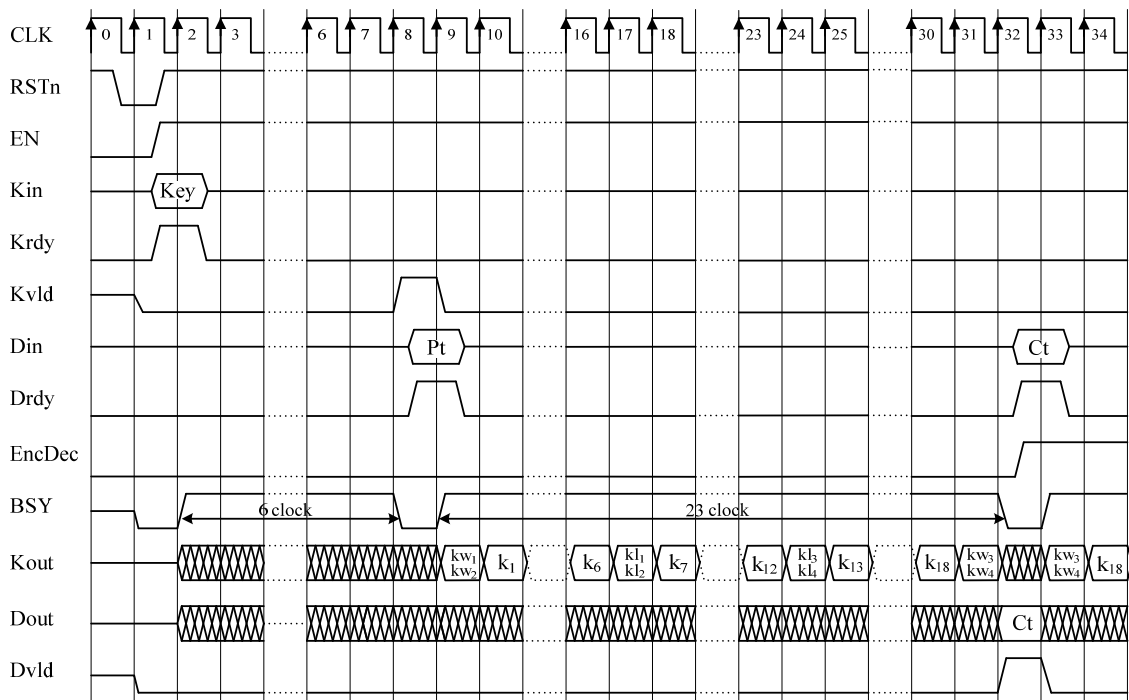


Figure 13-20 Timing Chart for Key Scheduling and Encryption of Camellia

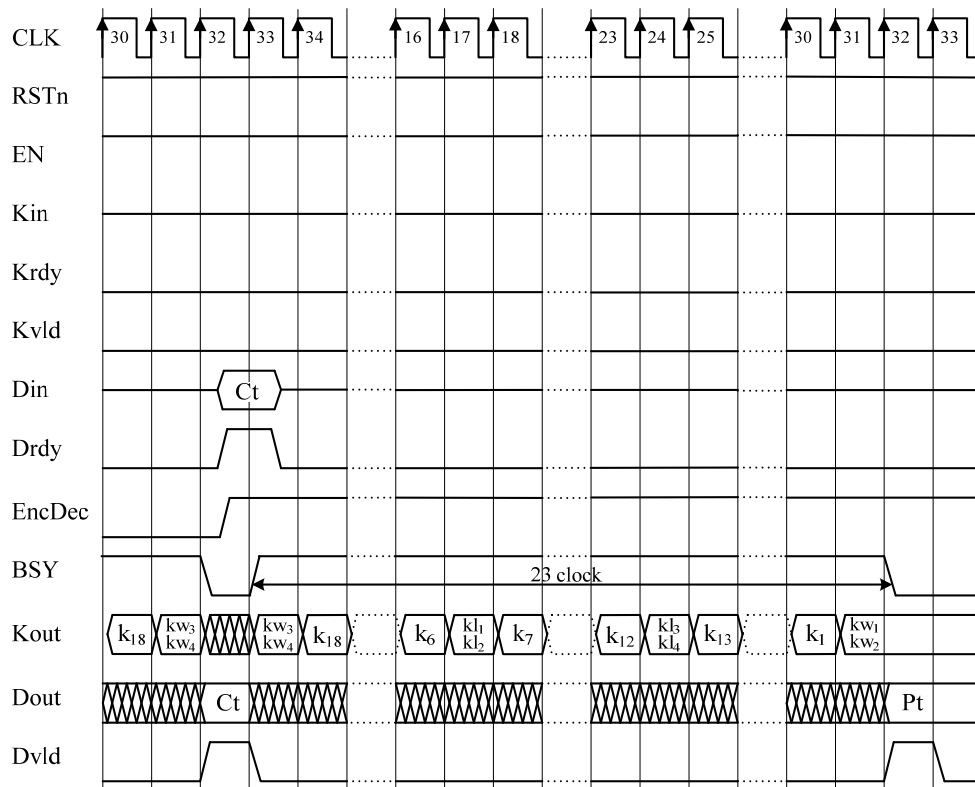


Figure 13-21 Decryption Timing Chart for Camellia

13.12. CAST-128

CAST-128⁽¹²⁾ is a block cipher that takes a 64-bit data block and 128-bit key. The overview specifications and I/O ports of the cryptographic circuit macro for CAST-128 are shown in Table

13-14 and Table 13-15, respectively.

Table 13-14 Overview Specifications of CAST-128

Algorithm	CAST-128
Data block length	64 bits
Key length	128 bits
Function	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	CAST128.v
Description language	Verilog-HDL
Top module	CAST
S-box	Table implementation
Throughput	64 bits / 17 clocks
Round key generation	Pre-calculation

Table 13-15 I/O Ports of CAST-128

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output. Only Kri (5 bits) and Kmi (32 bits) in the lower bits are valid. The upper 91 bits are padded with 0s.
Din	In	64	Data input.
Dout	Out	64	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 64-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the CAST-128 macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0 at the next clock.

Figure 13-22 represents the datapath architecture of the CAST-128¹³⁾ macro. CAST-128 is a block cipher with the Feistel structure and suits software implementation on a 32-bit processor. However, because it requires a 32-bit adder-subtractor and 8 different S-boxes with 8-bit inputs and 32-bit outputs represented as random tables, a hardware implementation will have a large circuit size. Besides, the key scheduling part has additional large register arrays to hold pre-calculated round keys so that the macro can operate at 1 round per clock cycle. The two round keys, 5-bit Kr_i and

32-bit Km_i , are placed in the lower side of the external 128-bit port $Kout$, and the upper 91 bits of the port are padded with 0s.

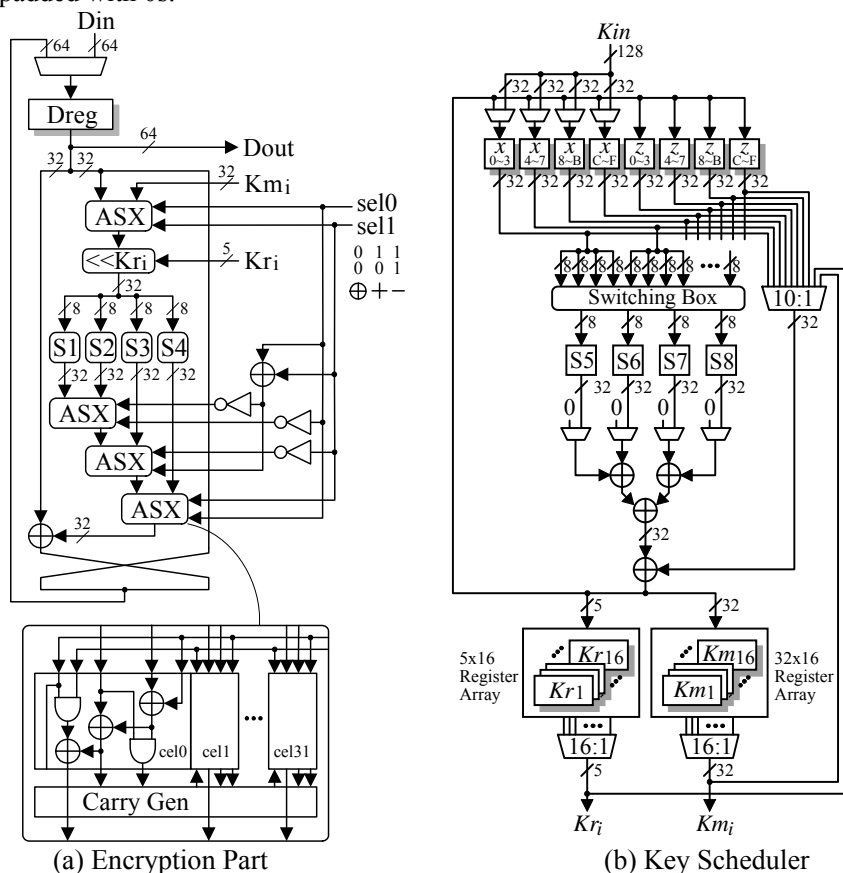


Figure 13-22 Datapath Architecture of CAST-128

Figure 13-23 illustrates the timings for key scheduling, encryption, and decryption for CAST-128 each with the minimum possible cycles. The operation(s) within each clock cycle follow:

- CLK1:** $RSTn=0$ resets the control circuit.
- CLK2:** $Kr_{dy}=1$ transfers the 128-bit secret key presented on Kin to the internal register.
- CLK3:** Key scheduling starts, turning the busy signal BSY to 1 and Kr_{dy} to 0.
- CLK130:** Key scheduling completes in 128 clock cycles. The $Kvld$ flag goes to 1 to indicate that the initial key has become valid, while BSY turns to 0.
- CLK131:** From this cycle on, plaintext or ciphertext blocks can enter to the circuit. With $EncDec=0$ for encryption, $Dr_{dy}=1$ loads the plaintext presented on the 64-bit input port Din into the data register $Dreg$.
- CLK132:** Encryption begins, turning the busy signal BSY to 1. From this cycle on, $Kout$ will be exporting the round keys Kr_i and Km_i every clock cycle. Likewise, the 64-bit port $Dout$ outputs the intermediate values forwarded from $Dreg$. Thus, during the whole encryption process, the round keys and intermediate values are output every clock cycle.
- CLK148:** Encryption completes in 16 clock cycles. $Dout$ presents the 64-bit ciphertext and BSY falls to 0. $Dvld$ turns to 1 at this clock and returns to 0 at the next clock.
- CLK49:** $Dr_{dy}=1$ initiates the next operation. At this clock, with $EncDec=1$ for decryption, the 64-bit port Din latches the ciphertext.
- CLK150:** Decryption begins, turning the busy signal BSY to 1. Similarly to encryption, $Kout$ and $Dout$ output the round keys and intermediate values every clock cycle, respectively.
- CLK165:** Decryption finishes in 163 clock cycles. $Dout$ outputs the 64-bit plaintext and BSY turns to 0. $Dvld$ turns 1 for a single clock cycle.

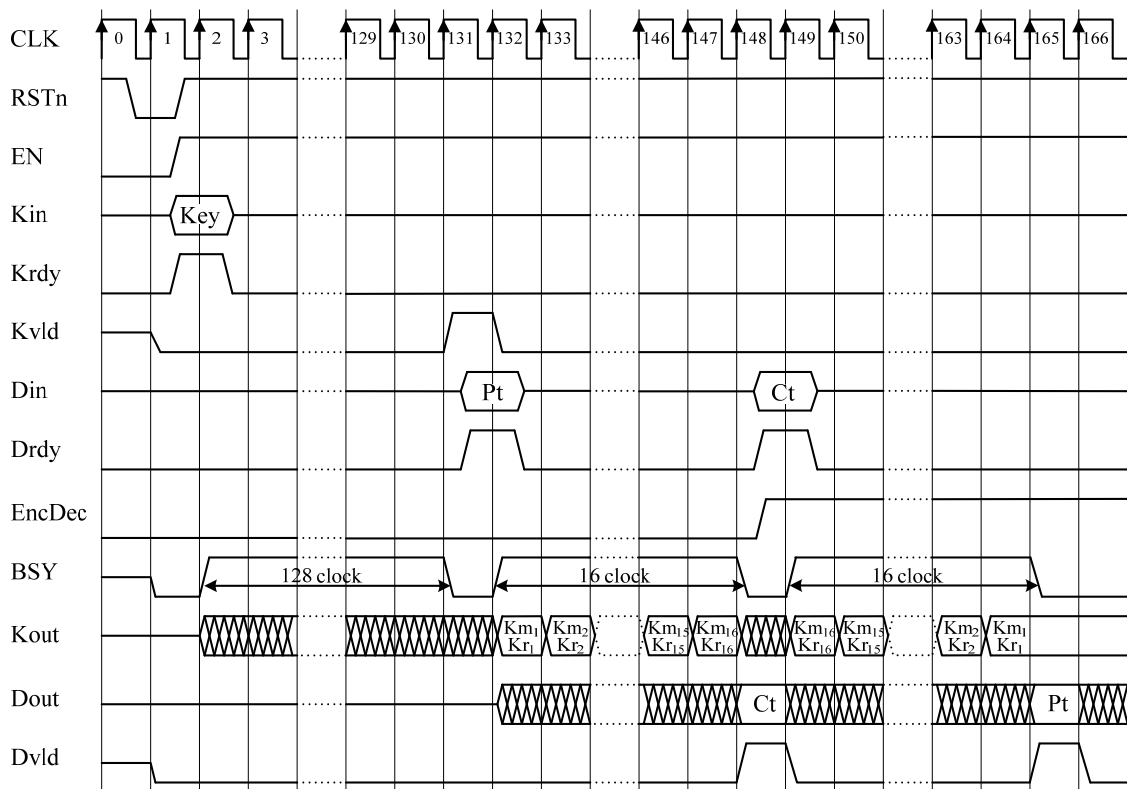


Figure 13-23 Timing Chart for CAST-128

13.13. DES

The overview specifications and I/O ports of the cryptographic circuit macro for DES¹⁴⁾ are shown in Table 13-16 and Table 13-17, respectively. DES is a block cipher with the Feistel structure, and is thus suitable for small-footprint implementations.

Table 13-16 Overview Specifications of DES

Algorithm	DES
Data block length	64 bit
Key length	64 bit (Key 56bit+Parity 8bit)
Function	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	DES.v
Description language	Verilog-HDL
Top module	DES
S-box	Table implementation
Throughput	64 bits / 16 clocks
Round key generation	On-the-fly

Table 13-17 I/O Ports of DES

Port name	Direction	Bit width	Description
Kin	In	64	Key input.
Kout	Out	128	48-bit round key output. The upper 80 bits are padded with 0s.
Din	In	64	Data input.
Dout	Out	64	Data output.
Krdy	In	1	When Krdy=1, a 64-bit secret key given to Kin is latched into the internal register on the rising clock edge. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 64-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
STn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the DES macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	After a key is input, and the converted key is set into the internal register, Kvld goes to 1 for a single clock cycle and returns to 0 at the next clock. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0 at the next clock.

Figure 13-24 represents the datapath architecture of the DES macro. It employs a simple implementation that repeatedly uses the 32-bit round function block. Kreg takes the 56-bit key out of the 64-bit key excluding the 8 parity bits. Parity check is not performed. Key scheduling takes place on-the-fly; The 48-bit round key is exported from the 128-bit port Kout, with the upper 80 bits padded with 0s, during encryption or decryption.

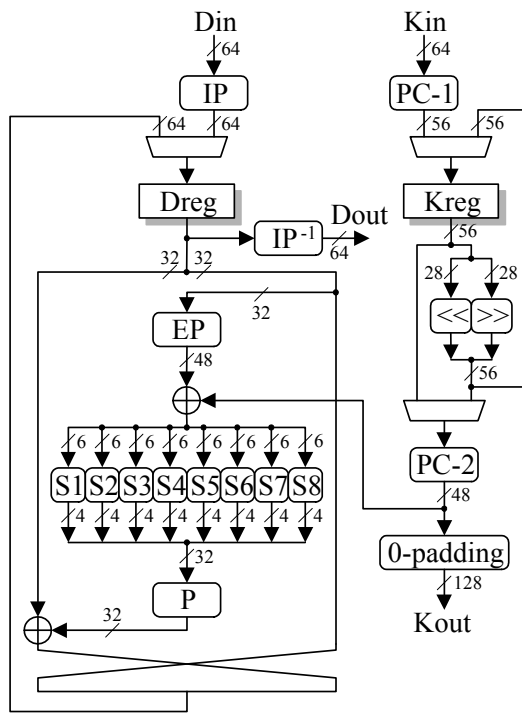


Figure 13-24 Datapath Architecture of DES

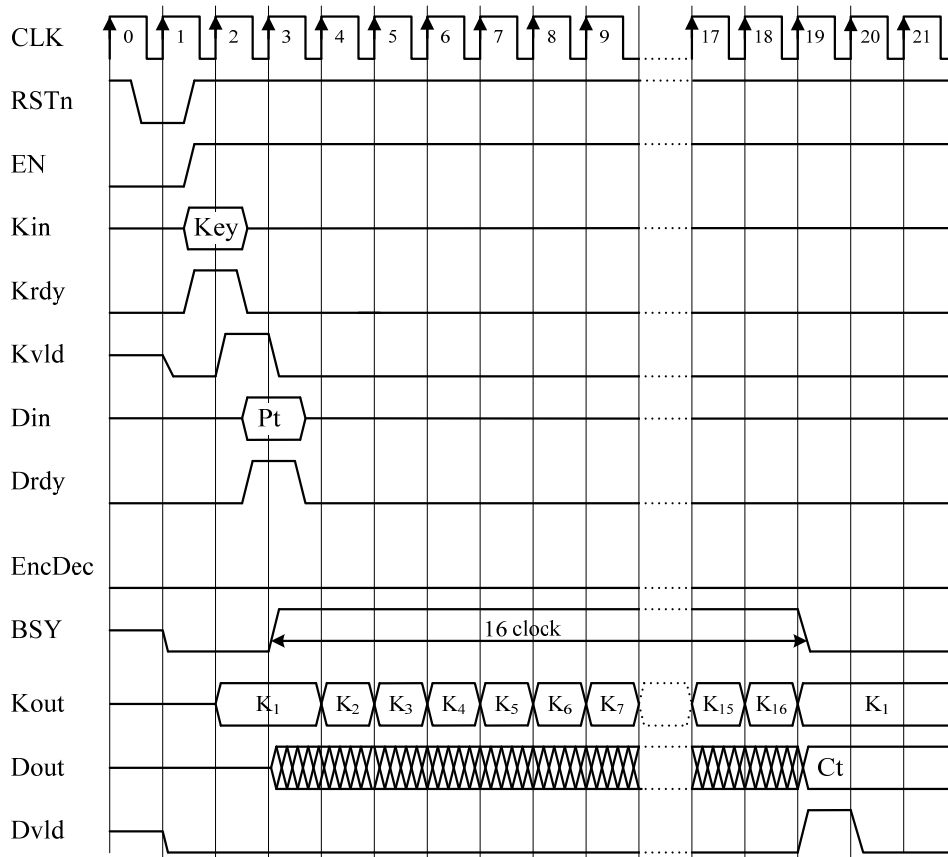


Figure 13-25 Timing Chart for DES

Figure 13-25 illustrates the encryption timing for DES with the minimum possible cycles. The decryption timing is the same as that for encryption except that the round keys are used in sequence from K16 to K1. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.

CLK2: Krdy=1 transfers the secret key presented on the 64-bit port Kin to the internal register.

CLK3: Because no advance key scheduling is involved, the Kvld flag goes to 1 immediately to indicate that the key has become valid. With EncDec=0 for encryption, Drdy=1 loads the plaintext presented on the 64-bit port Din into the data register Dreg.

CLK4: Encryption begins, turning the busy signal BSY to 1. From this cycle on, Kout will be exporting the round keys every clock cycle, starting with K1. Likewise, the 64-bit port Dout outputs the intermediate results forwarded from Dreg. Thus, during the whole encryption process, the round keys and intermediate results are output every clock cycle.

CLK20: Encryption completes in 16 clock cycles. Dout presents the ciphertext and BSY falls to 0. Dvld turns to 1 at this clock and returns to 0 at the next clock.

13.14. ECC

The overview of specifications, I/O ports, and the 192bit×16word memory map of internal variable of the cryptographic circuit macro for ECC are shown in Table 13-18 to Table 13-20. The ECC macro computes elliptic scalar multiplications of points on the elliptic curve:

$$E : y^2 + xy = x^3 + ax^2 + b$$

over the finite field GF(2¹⁹¹) defined with the irreducible polynomial:

$$f(x) = x^{191} + x^9 + 1$$

The key is restricted to 64bit.

Table 13-18 Summary of the ECC module.

Algorithm	Montgomery Power Ladder Method with Randomizable Data Value and Address Value
Data Block Length	192 bit
Key Length	64 bit (with restriction)
Function	Elliptic scalar multiplication over GF(2 ¹⁹¹)
Source file	uec_2nd_ECC_OS.v
Description Language	Verilog-HDL
Top Module	uec_2nd_ECC_OS

Table 13-19 I/O ports of the ECC module.

Port name	Direction	Bit width	Description
Kin	In	32	Key Input. 64bit key (scalar) and 64bit random number data, total 128bit data are inputted in 4 clocks by sending 32bit per each clock using burst transmission.
Din	In	32	Data input including the initial point. 192 bits initial point 's x coordinate in Affine coordinate system, 192 bits initial point's z coordinate (z≠0) in Projective coordinate system, 192 bits the elliptic curve parameter b, total 572 bit data are inputted in 18 clocks by sending 32 bits per each clock using burst transmission.
Dout	Out	32	The output of the result elliptic scalar multiplication. After Dvld=1 is outputted, 192 bit data of the x coordinate of the result of elliptic scalar multiplication are outputted in 6

			clocks by sending 32 bits per each clock using burst transmission.
Krdy	In	1	At the clock when Krdy turns to 1, the secret key data and the random number data, total 128 bits data are inputted from Kin to internal register in 4 clocks using burst transmission.
Drdy	In	1	At the clock when Drdy turns to 1, 192 bits initial point's x coordinate on Affine coordinate system, 192 bits z coordinate ($z \neq 0$) on Projective coordinate system, 192 bits the elliptic curve's parameter b, in total 18 clocks with bursts transmission, are inputted from Din to internal registers. Subsequently, the elliptic scalar multiplication takes place.
RSTn	In	1	Rest signal. If "0" is inputted on this port, the control circuit and the internal registers are reset. The reset is proceed even when the enable signal EN=0, as long as the system clock CLK is supplied.
EN	In	1	Enable signal. When EN=1, the ECC macro becomes active. When EN=0, it returns to the initial state.
CLK	In	1	System clock. All internal registers captures data synchronously on the rising edge of this clock.
BSY	Out	1	The busy status flag. It turns to "1" during the calculation of scalar multiplication or data loading.
Kvld	Out	1	When the key loading completes, Kvld becomes "1" for only one clock cycle.
Dvld	Out	1	When the loading of initial point completes, Dvld becomes "1" (High) for only one clock cycle.

Table 13-20 Memory map of the ECC module.

Address	Purpose	Address	Purpose
0	0	8	Z_2
1	Reserved	9	Z_0
2	$R^2 \bmod M(x) = 0x402$	A	Z_1
3	X	B	1
4	B	C	X_2
5	t_1	D	X_0
6	t_2	E	X_1
7	t_3	F	Reserved

The ECC macro adopts, for elliptic scalar multiplication, the improved version of the López and Dahab's algorithm¹⁶⁾, which is the Montgomery powering ladder¹⁵⁾ in projective coordinates, using randomized address algorithm proposed by Ito et al.¹⁷⁾. By inputting the random number to Z coordinate of projective coordinates, the data randomizing is possible. The following demonstrates these algorithms as Algorithm 1, Algorithm 2, and Algorithm 3 respectively. All data is represented by polynomial representation.

- Algorithm 1: Montgomery Powering Ladder Method

Input: A point on the elliptic curve P , Positive integer $d = (1d_{k-2} \cdots d_1 d_0)_2$

Output: x coordinate of dP : $x(dP)$

```

1:  $P_1 \leftarrow P, P_2 \leftarrow 2P$ 
2: for  $i=k-2$  downto 0 do
3:   if  $d_i=1$  then
4:      $x(P_1) \leftarrow x(P_1) + x(P_2), x(P_2) \leftarrow x(2P_2)$ 
5:   else
6:      $x(P_2) \leftarrow x(P_2) + x(P_1), x(P_1) \leftarrow x(2P_1)$ 
7:   end if
8: end for
9: return  $x(P_1)$ 

```

- Algorithm 2: López and Dahab Algorithm. Montgomery Powering Ladder in Projective Coordinates

Input: $P_1 = (X_1, Z_1), P_2 = (X_2, Z_2), x = X(P_2 - P_1)$

Input: $P_1 = (X_1, Z_1)$

Output: $P_1 = P_1 + P_2$

Output: $P_1 = 2P_1$

```

1:  $X_1 \leftarrow X_1 Z_2$ 
2:  $Z_1 \leftarrow X_2 Z_1$ 
3:  $t_1 \leftarrow X_1 Z_1$ 
4:  $Z_1 \leftarrow X_1 + Z_1$ 
5:  $Z_1 \leftarrow Z_1 Z_1$ 
6:  $X_1 \leftarrow x Z_1 + t_1$ 
7: return  $P_1$ 

```

```

1:  $t_2 \leftarrow X_1 X_1$ 
2:  $t_3 \leftarrow Z_1 Z_1$ 
3:  $Z_1 \leftarrow t_2 t_3$ 
4:  $t_2 \leftarrow t_2 t_2$ 
5:  $t_3 \leftarrow t_3 t_3$ 
6:  $X_1 \leftarrow b t_3 + t_2$ 
7: return  $P_1$ 

```

- Algorithm 3: Montgomery powering ladder method with randomized address.

Input: $P, k = (1, k_{n-2}, \dots, k_0)_2, r = (r_{n-2}, \dots, r_0)_2$

Output: $x(kP)$

```

1:  $R[r_{n-1}] \leftarrow x(2P)$ 
2:  $R[1 \oplus r_{n-1}] \leftarrow x(P)$ 
3: for  $i = n-2$  downto 0 do
4:    $R[2] \leftarrow \text{PD}(R[k_{i+1} \oplus k_i \oplus r_{i+1}])$ 
5:    $R[1 \oplus r_i] \leftarrow \text{PA}(R[0], R[1])$ 
6:    $R[r_i] \leftarrow R[2]$ 
7: end for
8: return  $R[k_0 \oplus r_0]$ 

```

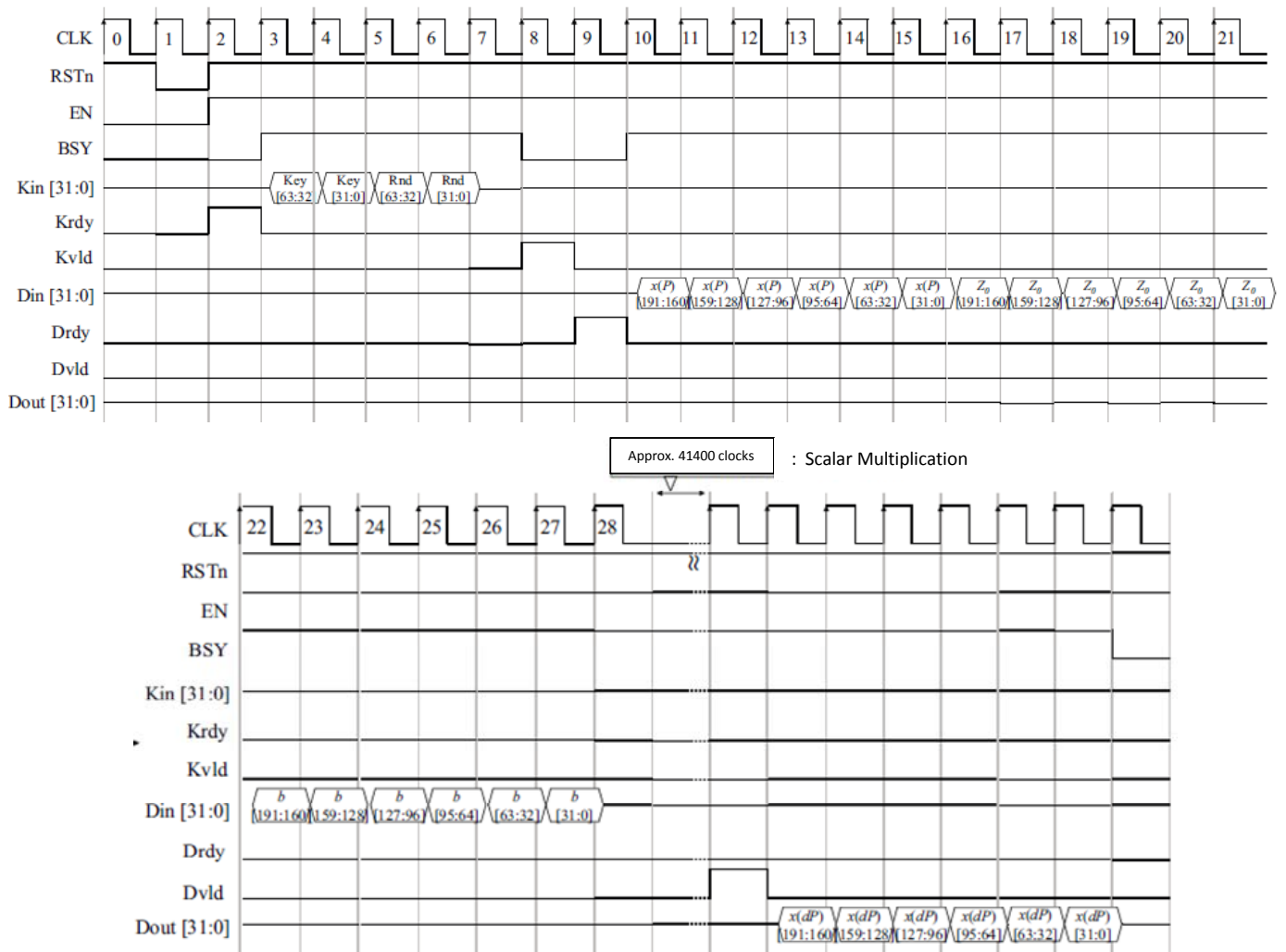


Figure 13-26 Timing chart of the ECC module.

The timing chart of the ECC circuit is illustrated in Figure 13-26. The operation(s) within each clock cycle is as follows.

CLK1: RSTn=0 resets the control circuit.

CLK2: EN=1, Krdy=1, from the next clock, the key(scalar) will be loaded.

CLK3~8: 64 bits key data and random data are loaded in 4 clocks with 32 bit per clock cycle using burst transmission, and they are stored in internal key register and random number register respectively. After the key data and random data loading completes, i.e., in **CLK8** Kvld turns to "1" (Kvld=1) for only 1 clock cycle. Also, the busy flag becomes "1" (BSY=1).

CLK9~27: To load the data of initial point in CLK9, Drdy is set to "1" (Drdy=1). 192 bits initial point's x coordinate on Affine coordinates, 192 bits z coordinate (z≠0) on Projective coordinates, 192 bits the elliptic curve's parameter b, are loaded into internal memory in total 18 clocks using bursts transmission.

CLK28~: The scalar multiplication is performed in approximately 414000 clocks. After the process completes, for only 1 clock Dvld turns to "1" (Dvld=1), and after that, 192 bits result of the elliptic scalar multiplication are outputted in 6 clocks with 32 bits per clock cycle using burst transmission.

13.15. MISTY1

The overview specifications and I/O ports of the cryptographic circuit macro MISTY1¹⁸⁾ are shown

in Table 13-21 and Table 13-22, respectively. MISTY1 is a block cipher with a nested Feistel structure.

Table 13-21 Overview Specifications of MISTY1

Algorithm	MISTY1
Data block length	64 bits
Key length	128 bits
Functions	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	MISTY1_1clk.v
Description language	Verilog-HDL
Top module	MISTY1
S-box	Table implementation
Throughput	64 bits / 9 clocks
Round key generation	On-the-fly

Table 13-22 I/O Ports of MISTY1

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	256	Outputs the 128-bit secret key concatenated with the 128-bit intermediate key.
Din	In	64	Data input.
Dout	Out	64	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 128-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the MISTY1 macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0 at the next clock.

Figure 13-27 represents the datapath architecture of the MISTY1 macro. Each single round is processed in a single clock cycle; Encryption and decryption for a 64-bit data block each take 9

clock cycles. Immediately after the 128-bit secret key's entry through the port Kin, the data randomizing part generates the intermediate keys in 8 clock cycles. After the key initialization, a plaintext or ciphertext block enters in the 64-bit port Din, and encryption or decryption begins, and subsequently the computed ciphertext or plaintext exits through the 64-bit port Dout.

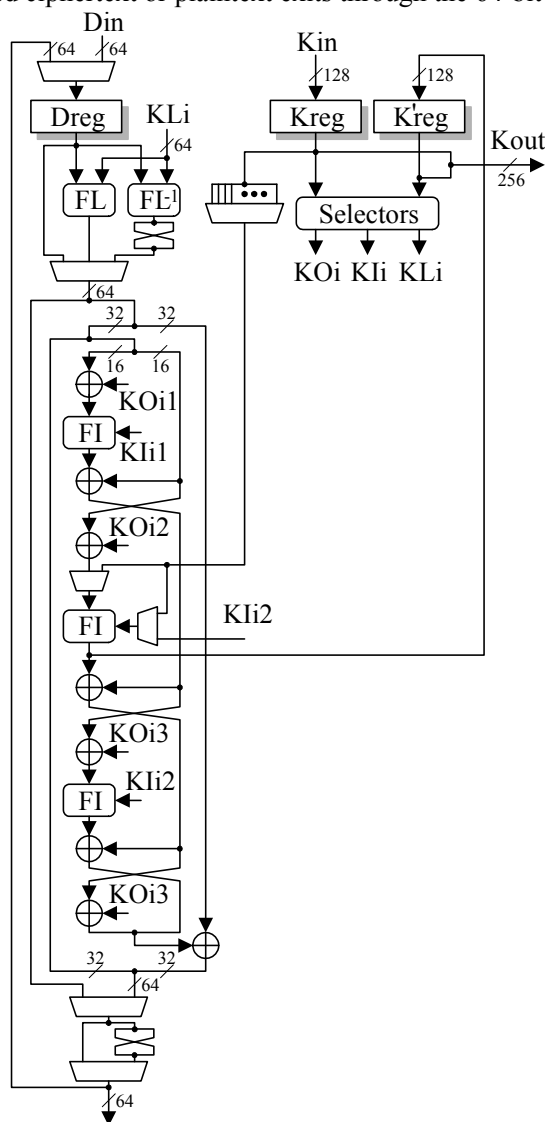


Figure 13-27 Datapath Architecture of MISTY1

Figure 13-28 illustrates the timings for key scheduling, encryption, and decryption for MISTY1 each with the minimum possible cycles. The operation(s) within each clock cycle follow:

- CLK1:** RSTn=0 resets the control circuit.
- CLK2:** Krdy=1 transfers the 128-bit secret key presented on Kin to the internal register Kreg.
- CLK3:** Key generation process starts to generate an intermediate key, turning the busy signal BSY to 1.
- CLK10:** Intermediate key generation completes. The intermediate key is set to the register Kreg. The Kvld flag goes to 1 for one clock cycle. BSY turns to 0.
- CLK11:** Drdy=1 loads the 64-bit plaintext PT presented on Din into the internal register Dreg.
- CLK12:** With EncDec=0, encryption begins, and the busy signal BSY turns to 1. From this clock on, Dout will be exporting the intermediate results every clock cycle. Likewise, Kout outputs the round keys.

- CLK13~20:** Encryption completes in 9 clock cycles. Dout presents the 64-bit ciphertext CT and BSY falls to 0. Dvld turns to 1 for one clock cycle.
- CLK21:** Drdy=1 with EncDec=1 loads the 64-bit ciphertext CT presented on Din into the internal register Dreg.
- CLK22:** With EncDec=1, decryption begins, and the busy signal BSY turns to 1. From this clock on, Dout will be exporting the intermediate results every clock cycle. Likewise, Kout outputs the round keys.
- CLK23~30:** Decryption completes in 9 clock cycles. Dout presents the 64-bit plaintext PT and BSY falls to 0. Dvld turns to 1 for one clock cycle.

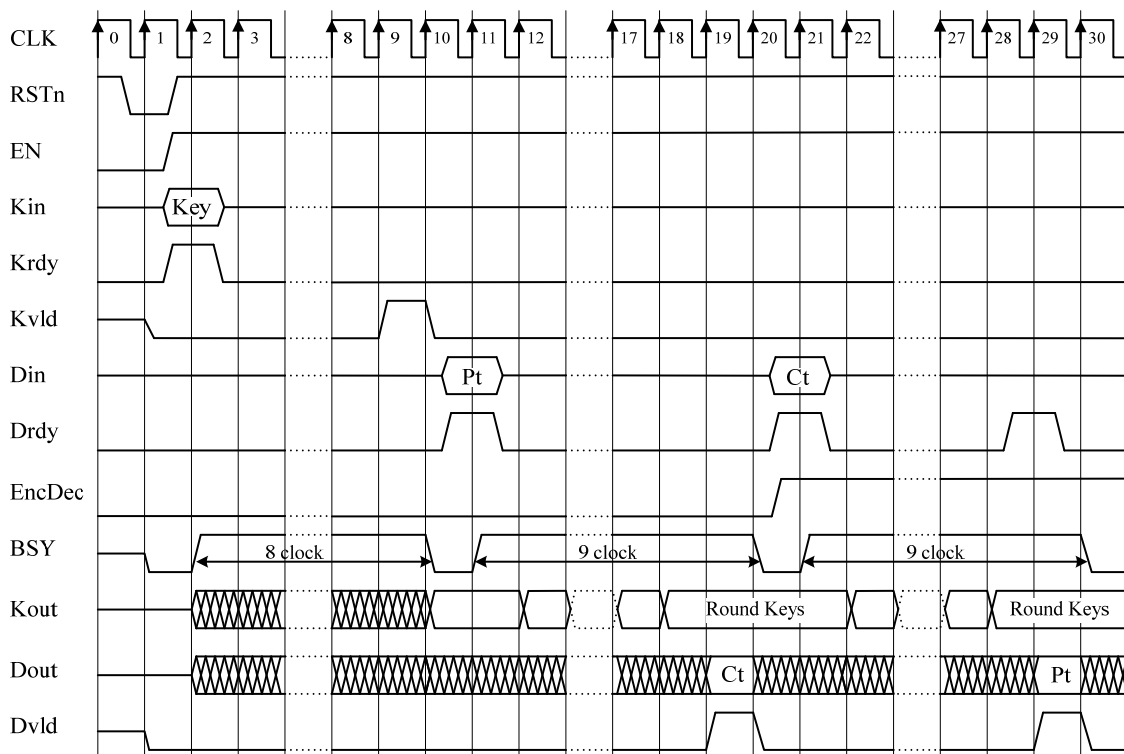


Figure 13-28 Timing Chart for MISTY1

13.16. RSA

The overview specifications and I/O ports of the cryptographic circuit macro for RSA¹⁹⁾ are shown in Table 13-23 and Table 13-24 respectively. The macro performs 512-bit encryption and decryption for the RSA cryptography with 6 modular exponentiation algorithms. In addition to a basic implementation of the binary method (left and right binary methods²⁰⁾), it employs the following countermeasures against side-channel attacks: the square-and-multiply always method (countermeasure with dummy operation)²¹⁾, Montgomery Powering Ladder²²⁾, and Square-Multiply exponentiation method²³⁾. Furthermore, it has an acceleration mode with the Chinese Remainder Theorem (CRT)²⁴⁾. Consequently, the macro supports 12 different combinations of operations. For multiply-add operation, it employs the Finely Integrated Operand Scanning (FIOS)²⁵⁾, a high-radix Montgomery multiplication algorithm.

Table 13-23 Overview Specifications of RSA

Algorithm	RSA
Data block length	512 bits
Key length	512 bits
Function	<ul style="list-style-type: none"> • CRT mode (non-CRT/CRT) • Modular exponentiation operations 0) Left binary method 1) Right binary method 2) Left binary method with dummy multiplication 3) Right binary method with dummy multiplication 4) Montgomery powering ladder 5) Square-multiply exponentiation method
Source file	RSA.v
Description language	Verilog-HDL
Top module	RSA
Throughput	non-CRT: 512 bits / approx. 452K clocks – 0) 1) 512 bits / approx. 599K clocks – 2) 3) 4) 5) CRT: 512 bits / approx. 135K clocks – 0) 1) 512 bits / approx. 176K clocks – 2) 3) 4) 5)

Table 13-24 I/O Ports of RSA

Port name	Direction	Bit width	Description
Kin	In	32	Key input. The 512-bit key data are taken in 32 bit blocks from the LSB sequentially in 16 cycles. When CRT is used, the two 256-bit keys are taken every 32 bits, each in 8 cycles continuously.
Min	In	32	Modulus input. The 512-bit modulus data $N (=pq)$ are taken every 32 bits from LSB sequentially in 16 cycles. When CRT is used, the two modules are taken every 32 bits, each in 8 cycles continuously. Subsequently, the preprocessed data $U=q^{-1} \bmod p$ is taken in 8 cycles.
Din	In	32	Data input. The 512-bit data are taken every 32 bits from the LSB sequentially in 16 cycles.
Dout	Out	32	Data output. After Dvld goes to 1, the 512-bit data are exported every 32 bits from the LSB sequentially in 16 cycles.
Krdy	In	1	After Krdy=1, a key divided into 32-bit pieces will be latched into the internal register. If 1s are given to both Mrdy and Krdy at the same time, only Krdy is effective.
Mrdy	In	1	After Mrdy=1, a modulus divided into 32-bit pieces will be loaded into the internal memory.
Drdy	In	1	After Drdy=1, data divided into 32-bit pieces will be loaded into the internal memory. Subsequently, encryption begins.
RSTn	In	1	A 0 on the reset signal RSTn resets the sequencer block and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the RSA macro.
CRT	In	1	CRT=1 specifies the CRT acceleration (CRT), while CRT=0 indicates that no CRT acceleration is used (non-CRT).

MODE	In	3	MODE specifies the operation mode for modular exponentiation. MODE=0, 1, 2, 3, 4, and 5 indicate 0) left binary, 1) right binary, 2) left binary with dummy multiplication, 3) right binary with dummy multiplication, 4) Montgomery Powering Ladder, and 5) square-multiply exponentiation methods, respectively.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, or data loading, the busy status flag BSY indicates 1. Drdy, Mrdy, and Krdy will be ignored while BSY=1.
Kvld	Out	1	When 512-bit key loading completes, Kvld goes to 1 for one clock cycle and returns to 0 at the next clock.
Mvld	Out	1	When modulus loading completes, Mvld goes to 1 for one clock cycle and returns to 0 at the next clock.
Dvld	Out	1	When the whole modular exponentiation completes, Dvld goes to 1 for one clock cycle and returns to 0 at the next clock.

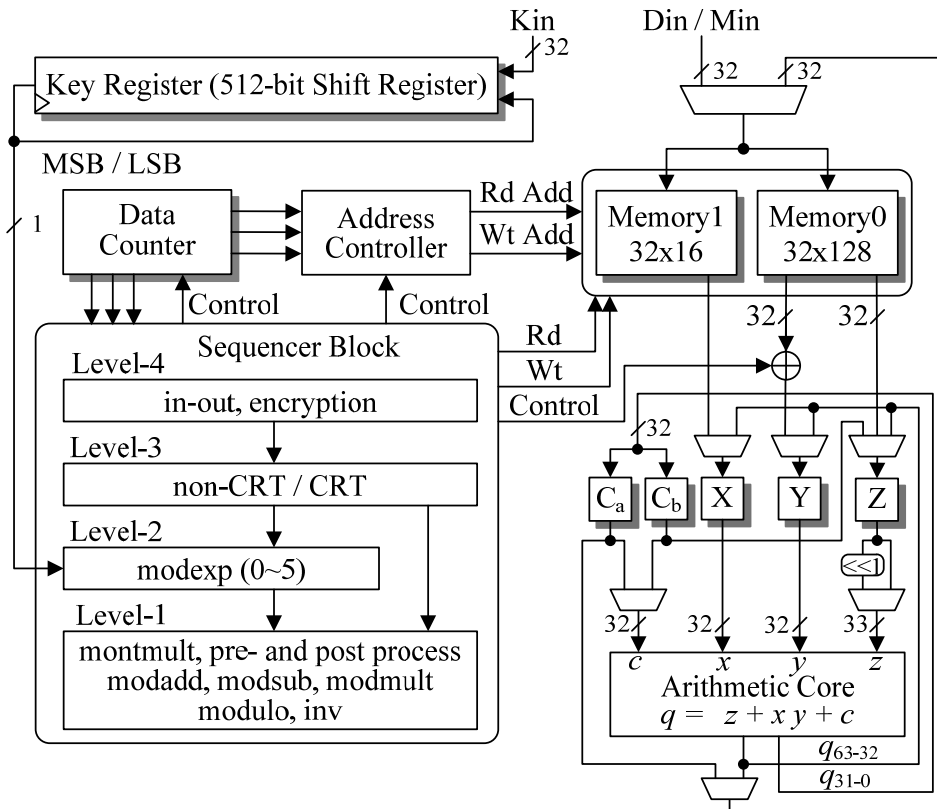


Figure 13-29 Circuit Architecture of RSA

Figure 13-29 shows the circuit architecture of the RSA macro. The macro consists of a key register, sequencer block, multiplication block, data counter, memory blocks, and address controller. The key register is a shift register storing a 512-bit key, which forwards the key information one bit a time to the sequencer block along with the modular exponentiation sequence. The data counter is comprised of three registers (two 9-bit registers and a 4-bit register) for holding data and a 9-bit adder. Two register arrays form the memory block. The address controller generates the addresses for the register arrays.

The sequencer block is made up of the 4 layers from Level 1 to Level 4. Level 4 controls the

input and output. Level 3 and Level 2 control the CRT mode and the 6 modular exponentiation sequences, respectively. Level 1 serves the sequence of each function called by the modular exponentiation operations and CRT. The controlled operations include: the Montgomery multiplication (montmult), pre-processing operations for the Montgomery multiplication (montredc, inv), some multiple-precision modular operations (modular operation (modulo), modular addition (modsub), modular subtraction (modsub)), multiple-precision multiplication (mult), data move, and data copy.

Figure 13-30 illustrates the timing for the RSA macro without CRT acceleration, in other words, with the non-CRT mode (CRT=0). In this chart, the left binary method modexp0 (MODE=0) is specified for the modular exponentiation algorithm. All the input signals are provided to minimize the cycles. The right binary method and the algorithms with countermeasures run with similar timings, except for the number of cycles of encryption; The right binary method takes approximately 452K cycles, while every algorithm with a countermeasure takes about 599K cycles.

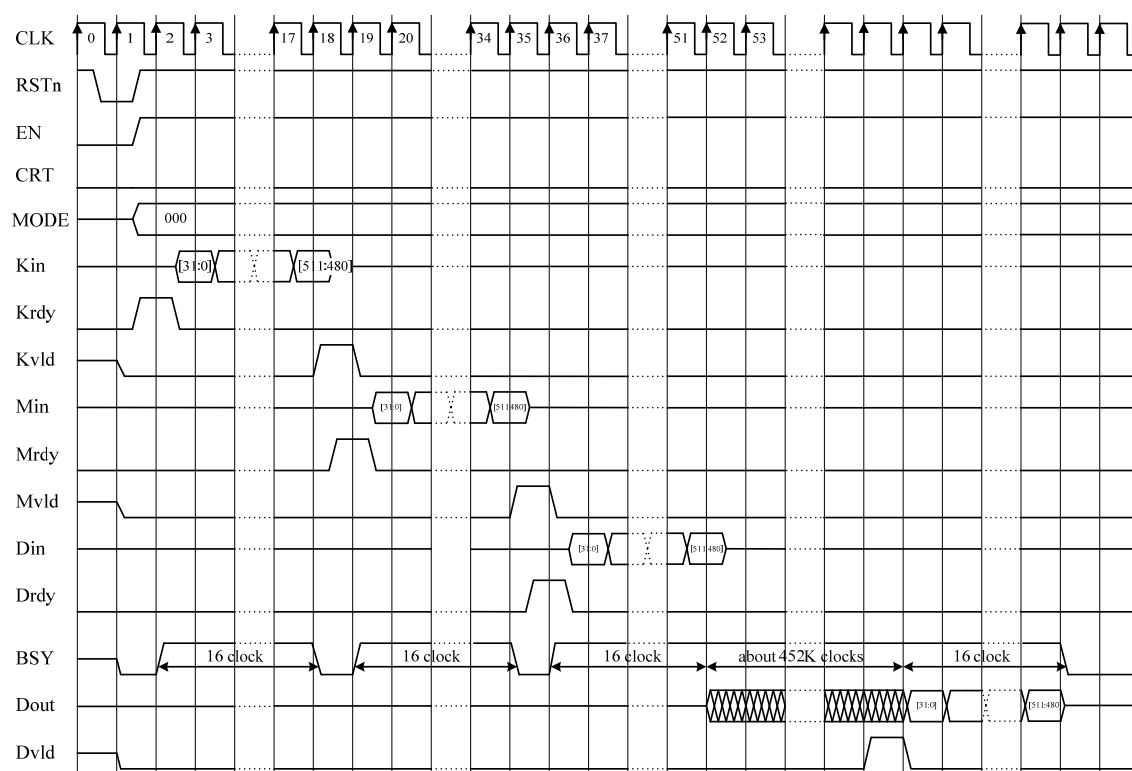


Figure 13-30 Timing Chart for RSA (non-CRT)

CLK1: RSTn=0 resets the sequencer block and registers.

CLK2: EN=1, CRT=0, and MODE=000 are set.

CLK2~18: After providing Krdy=1, the 512-bit key data are transferred through the 32-bit input port to the internal key register every 32 bits from the LSB sequentially. BSY=1 during the transfer. After 16 cycles, the sequencer block goes to the idle state, and BSY returns to 0. At CLK18, Kvld goes to 1 for a single clock cycle.

CLK19~35: After providing Mrdy=1, the 512-bit modulus data are transferred to the memory every 32 bits from the LSB sequentially in the same way as the key data. BSY=1 during the transfer. After 16 cycles, the sequencer block goes to the idle state, and BSY returns to 0. At CLK35, Mvld goes to 1 for a single clock cycle.

CLK36~52: With the key and modulus stored, Drdy=1 initiates loading the 512-bit plaintext to the memory. The plaintext data enter every 32 bits from the LSB sequentially. BSY=1 during the transfer. Immediately after the transfer completes, encryption begins.

CLK53~: Taking approximately 452K clock cycles, the modular exponentiation process runs. When

the whole process completes, Dvld goes to 1 for one clock cycle. After that, the ciphertext data are exported every 32 bits from the LSB sequentially in 16 clock cycles. Subsequently, BSY returns to 0 and the sequencer block moves into the idle state.

Figure 13-31 illustrates the timing for the RSA macro in the CRT mode. The timing is almost the same as in Figure 13-30 except for the input sequences of the modulus and key and for the number of cycles of encryption. The CRT mode requires the modulus and keys each be separated into two parts, and thus the key and modulus are loaded during the cycles between CLK2 and CLK18, and between CLK19 and CLK35, respectively, with each 256-bit part divided every 32 bits in 8 cycles. After the modulus transfer, during the cycles between CLK36 and 43, the pre-processed data ($U = q^{-1} \bmod p$, where $N = pq$) are loaded in 8 clock cycles. Thus, Mrdy=1 causes the input processes for 24 clock cycles in total. With the CRT mode, the macro takes 135K and 176K clock cycles for the binary methods and countermeasure algorithms, respectively.

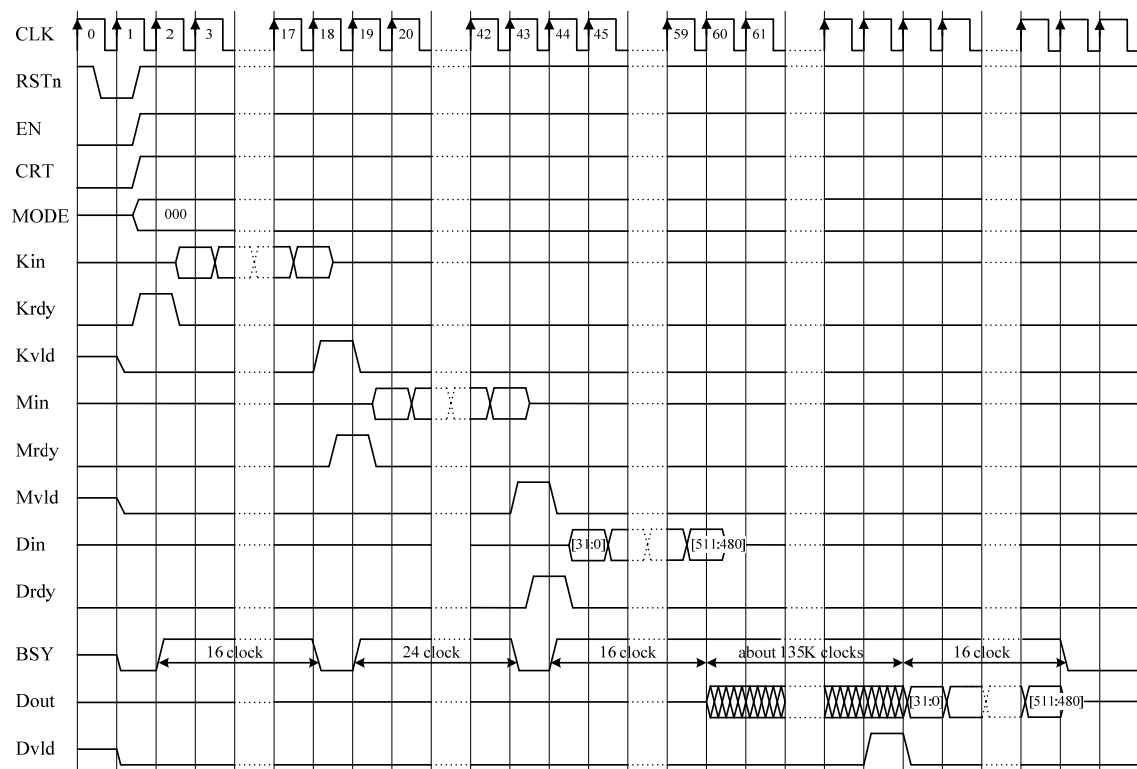


Figure 13-31 Timing Chart for RSA (CRT)

13.17. SEED

The overview specifications and I/O ports of the cryptographic circuit macro for SEED²⁶⁾ are shown in Table 13-25 and Table 13-26 respectively. SEED is a block cipher with the Feistel structure, proposed by KISA (Korea Information Security Agency).

Table 13-25 Overview Specifications of SEED

Algorithm	SEED
Data block length	128 bits
Key length	128 bits
Functions	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	SEED.v
Description language	Verilog-HDL
Top module	SEED
S-box	Table implementation
Throughput	128 bit / 23 clock
Round key generation	Pre-calculation and On-the-fly

Table 13-26 I/O Ports of SEED

Port name	Direction	Bit width	Description
Kin	In	128	Key input.
Kout	Out	128	Round key output.
Din	In	128	Data input.
Dout	Out	128	Data output.
Krdy	In	1	When Krdy=1, a 128-bit secret key given to Kin is latched into the internal register on the rising clock edge, and key initialization begins. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 128-bit plaintext or ciphertext given to Din is latched into the internal register on the rising clock edge and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the SEED macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When key initialization completes, Kvld goes to 1 during one clock cycle and returns to 0. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0.

Figure 13-32 represents the datapath architecture of the SEED macro. Each single round is processed in a single clock cycle; Encryption for a 128-bit plaintext and decryption for a 128-bit ciphertext each take 16 clock cycles. The 64-bit round function has the nested structure that iteratively uses a set of the 32-bit G function, XOR, and addition (or subtraction), three times, similarly to that of MISTY1. The G function consists of the 4 8-bit S-boxes and a 32-bit Permutation function. The 128-bit secret key is processed through the circular shifter, adders, subtractors, and eventually G function, to generate the 16 64-bit round keys K1~K16 on-the-fly.

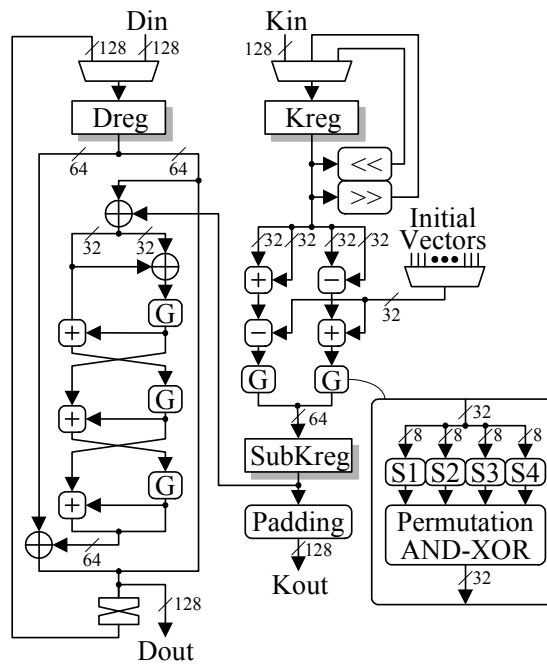


Figure 13-32 Datapath Architecture of SEED

Figure 13-33 illustrates the timings for key scheduling, encryption, and decryption for the SEED macro each with the minimum possible cycles. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.

CLK2: Krdy=1 with EncDec=0 transfers the 128-bit secret key presented on Kin to the internal register for encryption.

CLK3: Key scheduling starts, turning the busy signal BSY to 1. This process completes within this clock cycle. Kvld goes to 1 for one clock cycle. If switching from encryption to decryption by alternating the logic level of EncDec, another key scheduling has to be performed. At this time, the 128-bit port Kout presents the first round key K_1 for encryption.

CLK4: Drdy=1 stores the plaintext Pt presented on the 128-bit port Din into the data register Dreg. BSY falls to 0 as key scheduling completes.

CLK5: Encryption begins, turning BSY to 1. From this clock on, Dout will be exporting the intermediate results forwarded from Dreg every clock cycle. Likewise, Kout outputs the round keys starting with K_2 .

CLK20: Encryption completes in 16 clock cycles. Dout presents the ciphertext Ct and BSY falls to 0. Dvld turns to 1 for one clock cycle.

CLK21: Krdy=1 with EncDec=1 transfers the secret key Key to the internal register for decryption.

CLK22: Key scheduling starts, turning the busy signal BSY to 1. This process completes within this clock cycle. Kvld turns to 1 for one clock cycle. At this time, the 128-bit port Kout presents the first round key K_{16} for decryption.

CLK23: Drdy=1 stores the ciphertext Ct presented on the 128-bit port Din into the data register Dreg. BSY falls to 0 as key scheduling completes.

CLK24: Decryption begins, turning BSY to 1. Similarly to encryption, Dout and Kout output the intermediate results and round keys every clock cycle.

CLK39: Decryption completes in 16 clock cycles. Dout presents the plaintext Pt and BSY falls to 0. Dvld turns to 1 for one clock cycle.

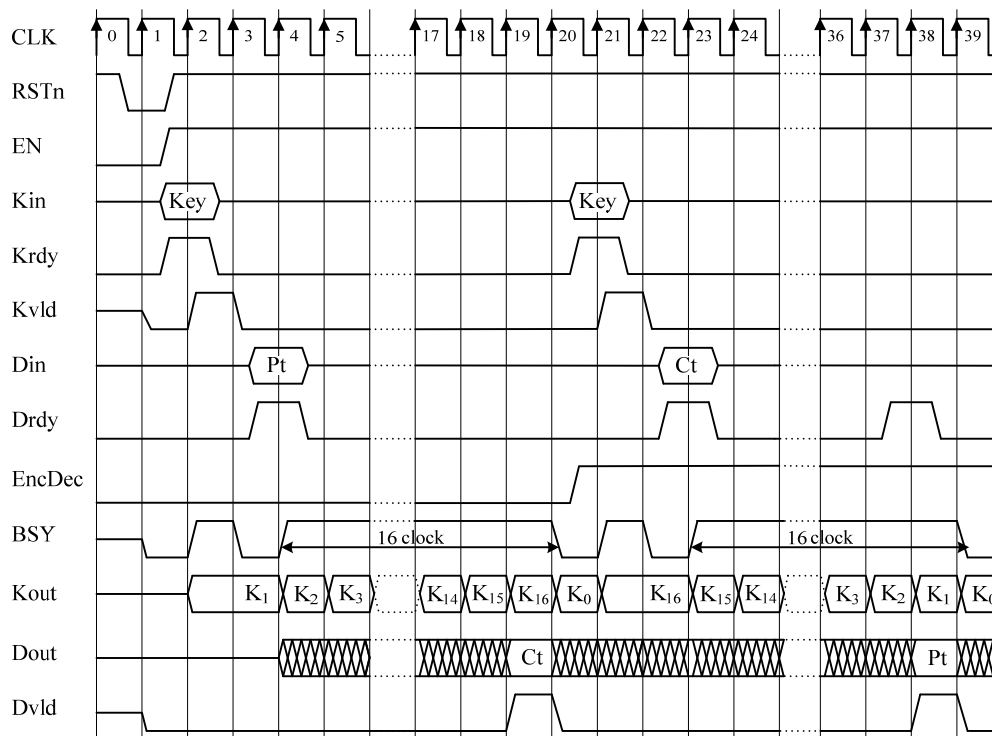


Figure 13-33 Timing Chart for SEED

13.18. TDES

The overview specifications and I/O ports of the cryptographic circuit macro for TDES (Triple-DES)¹⁴⁾ are shown in Table 13-27 and Table 13-28, respectively. TDES performs DES, the 56-bit-key block cipher, three times by changing the keys (3 x 16 cycles = 48 cycles); TDES encryption is broken down to [DES encryption]-[DES-decryption]-[DES-encryption], while TDES decryption is [DES decryption]-[DES-encryption]-[DES-decryption]. The TDES macro supports the 3-key Triple-DES, which uses three different keys. The macro takes the three keys in 3 cycles continuously. If the first and last keys provided are the same, the macro operates as the 2-key Triple-DES. If the three keys are all the same, DES encryption and DES decryption cancel each other out; Thus, the macro runs as a simple equivalent DES operation, although the number of cycles to complete the whole operation remains 48 for the three elementary DES operations.

Table 13-27 Overview Specifications of TDES

Algorithm	3-key Triple-DES
Data block length	64 bits
Key length	3 x 64 bits (56-bit key + 8-bit parity)
Functions	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	TDEA.v
Description language	Verilog-HDL
Top module	TDEA
S-box	Table implementation
Throughput	64 bits / 48 clocks
Round key generation	On-the-fly

Table 13-28 I/O Ports of TDES

Port name	Direction	Bit width	Description
Kin	In	64	Key input.
Kout	Out	128	48-bit round key output with upper 80 bits padded with 0s.
Din	In	64	Data input.
Dout	Out	64	Data output.
Krdy	In	1	When Krdy=1, 3 64-bit secret keys given to Kin are latched into the internal registers in 3 clock cycles. If 1s are given to both Drdy and Krdy at the same time, only Krdy is effective.
Drdy	In	1	When Drdy=1, a 64-bit plaintext or ciphertext given to Din is latched into the internal register, and encryption or decryption begins.
EncDec	In	1	EncDec=0 specifies that encryption is the operation to run when Drdy=1, while EncDec=1 means decryption.
RSTn	In	1	A 0 on the reset signal RSTn resets the control circuit and internal registers synchronously with CLK regardless of the logic level at EN.
EN	In	1	A 1 on the enable signal EN enables the TDES macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	During an active encryption, decryption, or key initialization process, the busy status flag BSY indicates 1. Drdy and Krdy will be ignored while BSY=1.
Kvld	Out	1	When the three key loading completes, Kvld goes to 1 during one clock cycle and returns to 0 at the next clock. Immediately after Kvld turns to 0, encryption or decryption can be activated.
Dvld	Out	1	When encryption or decryption completes and the ciphertext or plaintext is set on the data output port Dout, Dvld goes to 1 during one clock cycle and returns to 0 at the next clock.

Figure 13-34 represents the datapath architecture of the TDES macro. The TDES macro differs from the DES macro only in the number of key registers, which is increased to three. It has the same single 32-bit round function block as that of the DES macro, and uses it 48 times repeatedly. The key registers Kreg1~3 each load a 56-bit key as the result of trimming the 8-bit parity off from the 64-bit key input. Parity check is not performed, same as in the DES macro. Key scheduling takes place on-the-fly. The 48-bit round key is exported from the 128-bit port Kout, with the upper 80 bits padded with 0s, during encryption or decryption.

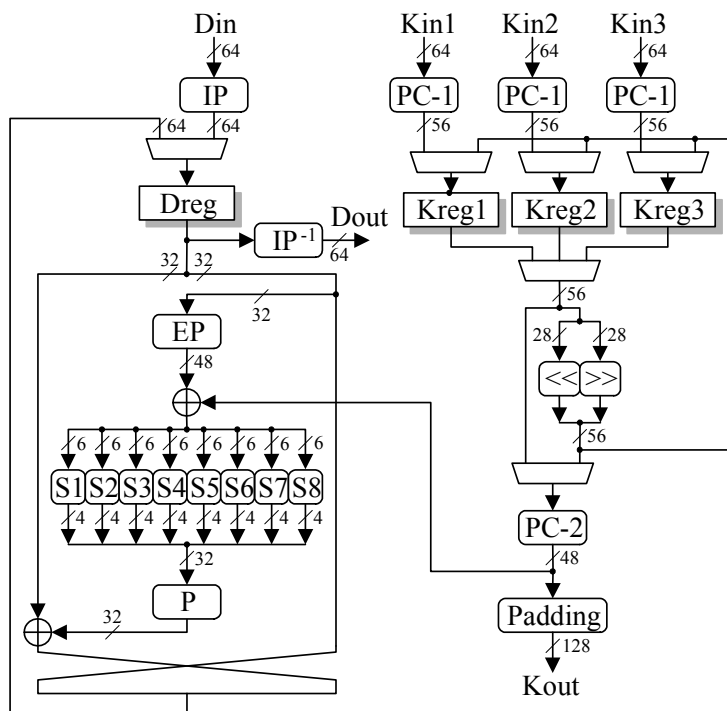


Figure 13-34 Datapath Architecture of TDES

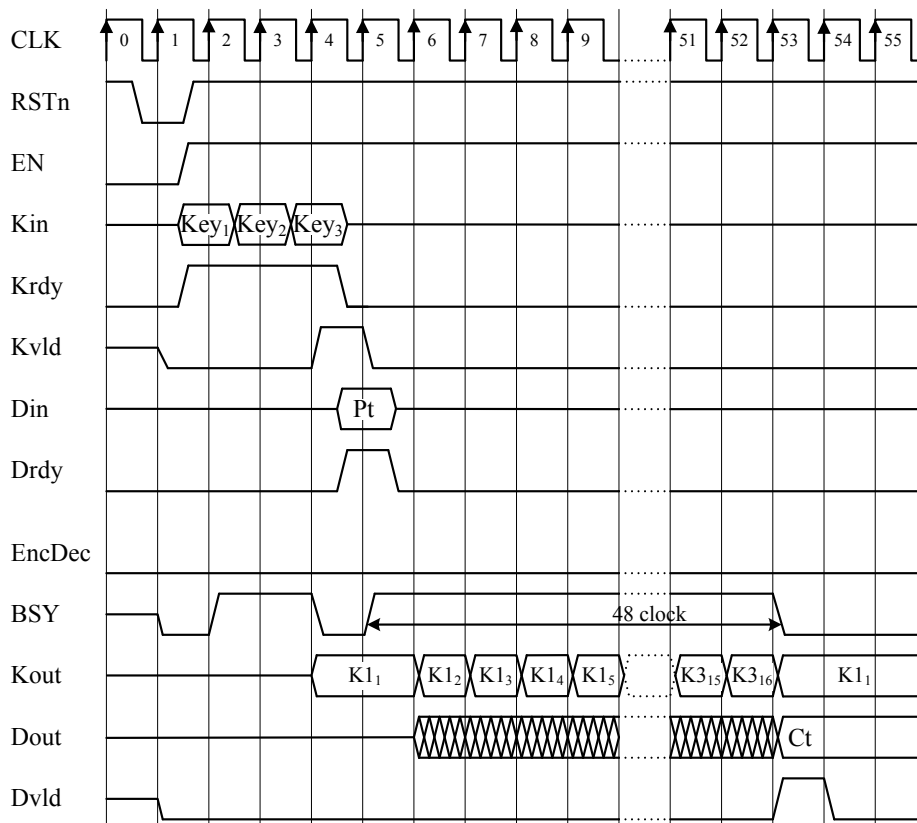


Figure 13-35 Timing Chart for TDES

Figure 13-35 illustrates the encryption timing for TDES with the minimum possible cycles. The decryption timing is the same as that for encryption except that the round keys are used in sequence from K16 to K1. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.

CLK2~4: With Krdy=1, the internal registers Kreg1~3 load the three secret keys Key₁~Key₃ presented on the 64-bit port Kin in sequence.

CLK5: Because no advance key scheduling is involved, the Kvld flag goes to 1 immediately to indicate that the keys have become valid. With EncDec=0 for encryption, Drdy=1 loads the plaintext Pt presented on the 64-bit port Din into the data register Dreg.

CLK6: Encryption begins, turning the busy signal BSY to 1. From this cycle on, Kout will be exporting the round keys every clock cycle, starting with K1₁ corresponding to the first secret key Key₁. Likewise, the 64-bit port Dout outputs the intermediate results forwarded from Dreg. Thus, during the whole encryption process, the round keys and intermediate results are output every clock cycle.

CLK54: Encryption completes in 48 clock cycles. Dout presents the ciphertext Ct and BSY falls to 0. Dvld turns to 1 at this clock and returns to 0 at the next clock.

13.19. CLEFIA

The overview specification and I/O ports of CLEFIA^{27), 28)} hardware macro are summarized in Table 13-29 and Table 13-30, respectively. 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. Note that this hardware macro is based on the paper [29].

Table 13-29 Summary of CLEFIA

Algorithm	CLEFIA
Data block size	128 bit
Key size	128 bit
Mode of operation	Electronic Code Book (ECB)
Source file name	CLEFIA_Comp.v
Description Language	Verilog-HDL
Top module name	CLEFIA_Comp
Throughput	128 bit / 18 clock (Encryption) 128 bit / 19 clock (Decryption)
Round keys	On-the-fly

Table 13-30 I/O ports of CLEFIA

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. An input data should be kept

			while the encryption/decryption process is running.
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 being completed, this signal becomes 1 during the final one clock cycle of the process, and then it goes 0.
Dvld	Out	1	This signal is set to 1 when the data in the port Dout is valid and ready to output. This signal becomes 1 during the final one clock cycle of the encryption/decryption process (See also timing chart in Fig. 5.

Datapath of the CLEFIA macro is shown in Figure 13-36. This macro executes 1-round operation of the GFN in 1 clock cycle. The process of one message block (128 bits) requires 18 cycles for encryption, and 19 cycles for decryption.

A secret key is contained in an internal register K through a 128-bit port Kin in the key-scheduling. Then the intermediate key generation is started in the data randomization block. The generated intermediate key is set to an internal register L after 13 cycles. During the encryption/decryption process, the round keys are generated on the fly using the data contained in the registers K and L.

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

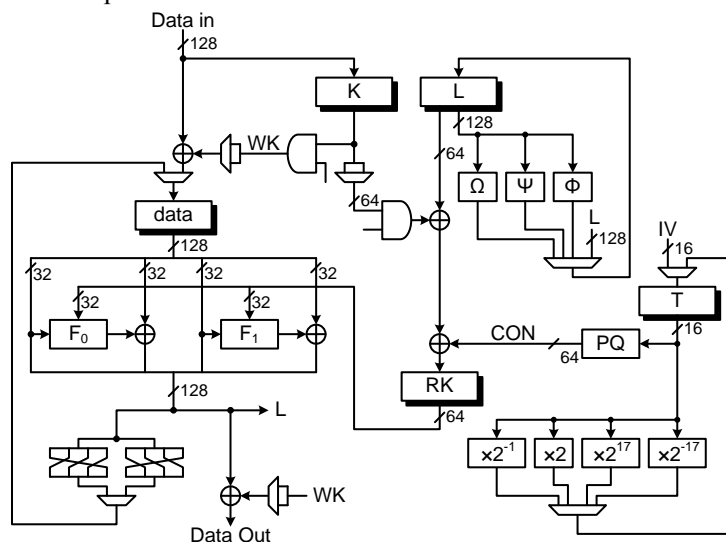


Figure 13-36 Data path architecture of CLEFIA

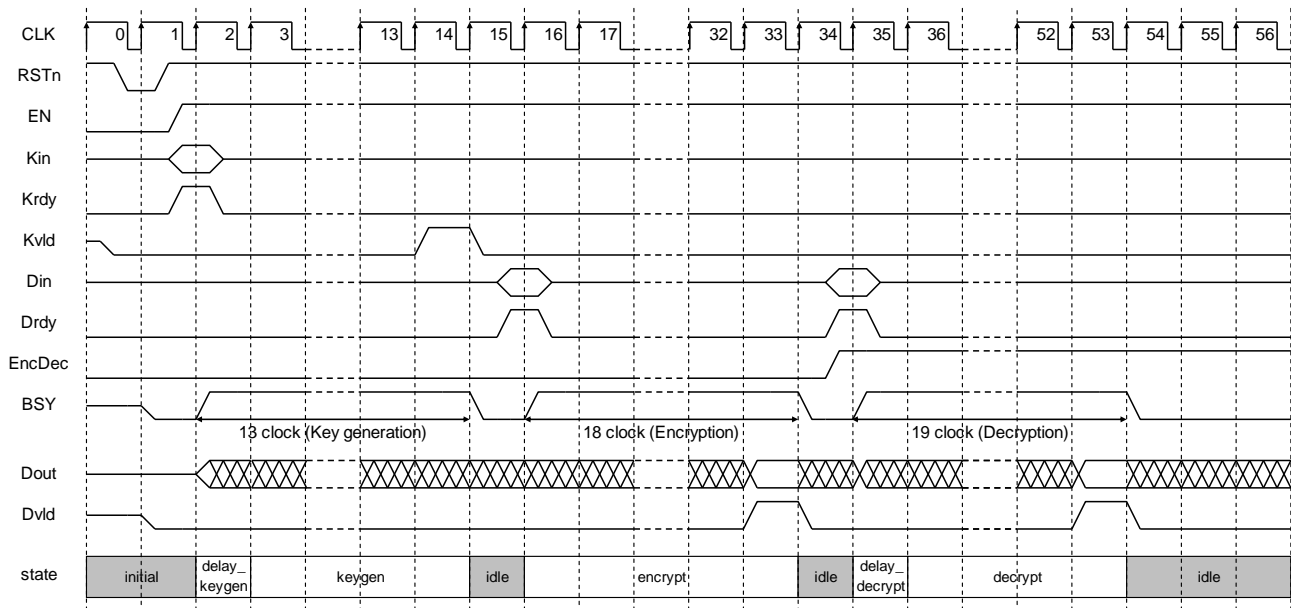


Figure 13-37 Timing chart of CLEFIA

Figure 13-37 shows the timing chart of the key scheduling, encryption, and decryption process for the CLEFIA macro in the minimum cycles for the control signals. The operations 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~CLK14:** The key scheduling process takes 13 cycles. Kvld is set to 1 in the final cycle of the process (i.e. CLK14). The sequencer returns to the idling state “IDLE” and BSY returns to 0 in the successive cycle
- CLK16:** By asserting Drdy=1, the 128-bit input (plaintext) is stored into an internal register. The encryption process is started in accordance with EncDec=0.
- CLK17~33:** The encryption process requires 18 cycles, and thus it is completed in CLK33. The output data (ciphertext) is valid only in the final cycle of the process (i.e. CLK33) and Dvld is set to 1 in the corresponding cycle. The sequencer is set to “IDLE” and BSY goes to 0 in the successive cycle.
- CLK35:** By asserting Drdy=1, the next operation is started. The decryption process is started in accordance with EncDec=1, and BSY is set to 1.
- CLK36~53:** The decryption process takes 19 clocks. and thus it is completed in CLK53. The output data (plaintext) is valid only in the final cycle (i.e. CLK53) and Dvld is set to 1 in the corresponding cycle. The sequencer is set to “IDLE” and BSY goes to 0 in the successive cycle.

References

- 1) ISO/IEC 18033-3 “Information technology – Security techniques – Encryption algorithm – Part 3: Block ciphers,” Jul. 2005.
<http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37972>
- 2) National Institute of Standards and Technology, “FIPS PUB 197: ADVANCED ENCRYPTION STANDARD (AES),” Nov. 2001.
- 3) A. Satoh, S. Morioka, K. Takano, S. Munetoh, “A Compact Rijndael Hardware Architecture with S-Box Optimization,” *Advances in Cryptology (ASIACRYPT 2001)*, LNCS 2248, pp. 239-254, Springer-Verlag, Dec. 2001.
- 4) S. Morioka, A. Satoh, “An Optimized S-Box Circuit Architecture for Low Power AES Design,” *Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*, LNCS 2523, pp. 271-295, Springer-Verlag, Aug. 2002.
- 5) NIST, “Recommendation for Block Cipher Modes of Operation: Methods and Techniques,” Special Publication 800-38A, Dec. 2001.
http://csrc.nist.gov/CryptoToolkit/modes/800-38_Series_Publications/SP800-38A.pdf
- 6) E. Trichina, “Combinational Logic Design for AES SubByte Transformation On masked Data,” *Cryptology ePrint Archive*, 2003/236, 2003.
- 7) S. Nikova and C. Rechberger, and V. Rijmen, “Threshold Implementations Against Side-Channel Attacks and Glitches,” *The 8th International Conference on Information and Communications Security (ICICS 2006)*, LNCS4307, pp. 529-545, Springer-Verlag, Dec. 2006.
- 8) T. Pop and S. Mangard, “Masked Dual-Rail Pre-charge Logic: DPA-Resistance without Routing Constrain,” *Workshop on Cryptographic Hardware and Embedded Systems (CHES2005)*, LNCS 3659, pp. 172-186, Springer-Verlag, Aug. 2005.
- 9) K. Tiri and I. Verbauwhede, “A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation,” *Proc. 2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004)*, pp. 246-251, Feb. 2004.
- 10) D. Suzuki, M. Saeki, and T. Ichikawa, “Random Switching Logic: A New Countermeasure against DPA and Second-Order DPA at the Logic Level,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 1, pp. 160-168, Jan. 2007.
- 11) 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>
- 12) C. Adams, “The CAST-128 Encryption Algorithm,” RFC2144 (Informational), May 1997.
- 13) T. Sugawara, N. Homma, T. Aoki, and A. Satoh, “A High-Performance ASIC Implementation of the 64-bit Block Cipher CAST-128,” *Proc. 2007 IEEE International Symposium on Circuits and Systems (ISCAS2007)*, pp. 1859-1862, May 2007.
- 14) NIST, “FIPS PUB 46-3: DATA ENCRYPTION STANDARD (DES),” Oct. 1999.
- 15) P. L. Montgomery, “Speeding the Pollard and elliptic curve methods of factorization,” *Mathematics of Computation*, vol. 48, no.177, pp. 243-264, 1987.
- 16) J. López, and R. Dahab, “Fast multiplication on elliptic curves over $GF(2^m)$,” *Workshop on Cryptographic Hardware and Embedded Systems (CHES '99)*, LNCS 1717, pp. 316-327, Springer-Verlag, Aug. 1999.
- 17) K. Itoh, T. Izu, and M. Takenaka, “A practical countermeasure against Address-Bit Differential Power Analysis,” in *Cryptographic Hardware and Embedded Systems (CHES '03)*, LNCS 2779, pp. 382–396, Springer, 2003.
- 18) M. Matsui, “Specification of MISTY1 - a 64-bit Block Cipher,” NESSIE Project.
<https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions.html>

- 19) R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.
- 20) J. A. Menezes, C. P. Oorschot, and A. S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997.
- 21) J. S. Coron: "Resistance against differential power analysis for elliptic curve cryptosystems", Workshop on Cryptographic Hardware and Embedded Systems (*CHES '99*), LNCS 1717, pp. 192-302, Springer-Verlag, Aug. 1999.
- 22) M. Joye and S. M. Yen, "The Montgomery powering ladder", Workshop on Cryptographic Hardware and Embedded Systems (*CHES2002*), LNCS 2523, pp. 291-302, Springer-Verlag, 2003.
- 23) M. Joye, "Highly Regular Right-to-Left Algorithms for Scalar Multiplication", Workshop on Cryptographic Hardware and Embedded Systems (*CHES2007*), LNCS 4727, pp. 135-147, Springer-Verlag, Sep. 2007.
- 24) J. J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public-key cryptosystem," *Electronics Letters*, vol. 18, no. 21, pp. 905-907, Oct. 1982.
- 25) C.K. Koc, T. Acar, and J. Burton S. Kaliski, "Analyzing and comparing Montgomery multiplication algorithms," *IEEE Micro*, vol. 16, no. 3, pp. 26-33, Jun 1996.
- 26) "SEED Algorithm Specification"
http://www.kisa.or.kr/seed/data/Document_pdf/SEED_Specification_english.pdf
- 27) Sony Corporation, "The 128-bit Block Cipher CLEFIA Algorithm Specification," Jun. 2007,
<http://www.sony.co.jp/Products/clefiat/technical/data/clefiat-spec-1.0.pdf>.
- 28) Sony Corporation, "The 128-bit Block Cipher CLEFIA Security and Performance Evaluations," Jun. 2007,
<http://www.sony.co.jp/Products/clefiat/technical/data/clefiat-eval-1.0.pdf>.
- 29) T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "High-performance ASIC Implementations of the 128-bit Block Cipher CLEFIA," *Proceedings of 2008 International Symposium on Circuits and Systems (ISCAS2008)*, pp. 2925--2928, May 2008

This LSI was developed by AIST (the National Institute of Advanced Industrial Science and Technology) in undertaking projects sponsored by METI (Ministry of Economy, Trade and Industry, Japan)

1. The copyright of this LSI belongs to AIST, and the copyright of each cryptographic hardware IPs belongs to each institute (AIST, Tohoku University, Yokohama University, or University of Electro-Communications).
2. Copying this document, in whole or in part, is prohibited without written permission from the copyholders.
3. Only personal or research use of this document and product is granted. Any other use of this document and LSI is not allowed without written permission from the copyholders.
4. The specifications of this LSI are subject to revision without notice.

Technical inquiries:

National Institute of Advanced Industrial Science and Technology (AIST)
Research Center for Information Security (RCIS)
Akihabara-Daiburu 10F Room 1003
1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
TEL: +81-3-5298-4722
FAX: +81-3-5298-4522