ISO/IEC 18033-3 Standard Cryptographic LSI
~ with Side Channel Attack Countermeasures ~
Specification

- Version 1.0 -

September 25, 2009

Research Center for Information Security,
National Institute of Advanced Industrial Science and Technology

Index

Page

1. OVERVIEW

To evaluate Differential Power Analysis attacks and other implementation attacks, “The special purpose LSI implementing standard cryptographic algorithms” (hereinafter called the cryptographic LSI) implements the following cryptographic algorithms: public-key cryptography RSA, elliptic-curve cryptosystem (ECC), and every common-key cryptographic algorithm appearing in ISO/IEC 18033(Information technology – Security techniques – Encryption algorithms) Part 3: Block ciphers. The cryptographic LSI is manufactured using TSMC (Taiwan Semiconductor Manufacturing Company) 130nm or 90nm CMOS processes and packaged in a 160-pin ceramic QFP.

The LSI has the 9 algorithms including AES, which is implemented in 14 different ways, and houses 22 different cryptographic cores altogether. AES implementations No.8 through No.13 used custom logic synthesis. Because the LSI is intended not only for Japanese domestic use but also for use overseas, the key length for each algorithm is limited to meet the export control regulations; The upper 72 bits of the 128-bit secret keys in the block ciphers except for DES are fixed so that the user can only change the lower 56 bits. Similarly, RSA only supports a 512-bit key.

- AES (key length : 128 bits)
  - No.0: S-Box implemented using composite field. Encryption and decryption.
  - No.1: S-Box implemented using case statement. Encryption only.
  - No.2: S-Box implemented using AND-XOR (1-Stage). Encryption only.
  - No.3: S-Box implemented using AND-XOR (3-Stage). Encryption only.
  - No.4: S-Box implemented using composite field. Encryption only.
  - No.5: CTR mode supported. Pipelined.
  - No.6: For fault-injection-attack resistance evaluation.
  - No.7: Precalculation of round keys.
  - No.8: For DPA countermeasure evaluation (Masked AND Operation).
  - No.9: For DPA countermeasure evaluation (MDPL).
  - No.10: For DPA countermeasure evaluation (Threshold Implementation).
  - No.11: For DPA countermeasure evaluation (WDDL).
  - No.12: For DPA countermeasure evaluation (Pseudo RSL).
  - No.13: For DPA countermeasure evaluation (for Pseudo RSL effectiveness assessment).

The *n*-th implementation of AES will hereinafter be denoted as AESn. For example, No.5 of AES is AES5.

- Camellia (key length : 128 bits): Encryption and decryption.
- SEED: Encryption and decryption.
- MISTY1: Encryption and decryption.
- Triple-DES: 3 Key. Encryption and decryption.
- DES: Encryption and decryption.
- CAST128: Encryption and decryption.
- ECC (key length: 64 bits): Scalar multiplication on a point over a field of characteristic 2.
- RSA: 512-bit modular exponentiation operations.

The main functionalities of the LSI are as follows:

- Computes the cryptographic algorithms.
- Interfaces with Virtex-II PRO FPGA (xc2vp30) on SASEBO-R (Side-channel Attack Standard Evaluation Board) developed for mounting a cryptographic LSI in FY07.
- Generates a trigger signal for sampling information such as power consumption.
- Exports the pre-designated intermediate keys and algorithm values for fault attack evaluation (Only supported for AES6).
- Exports the intermediate keys and algorithm values on a fault event for fault attack evaluation (Only supported for AES6).
- Runs in the free-run mode that repeats cipher operations every 0.3 seconds (Only supported for AES0).

2. EXTERNAL INTERFACE

2.1 I/O Assignments

The I/O signals of the cryptographic LSI are listed in Table 2.1. Table 2.2 and Figure 2.1 show the assignments of the 160 pins of the 130nm LSI. Table 2.3 and Figure 2.2 represent the assignments of the 160 pins of the 90nm LSI. The 130nm and 90nm LSIs have identical package dimensions, pin assignments, and logic interfaces, but different core voltages (1.2 V ± 0.12V and 1.0V ± 0.1V). Parentheses in the "Signal Name" column denote the reserved signals for the future extension. The cryptographic LSI does not use those signals. For VSS/VDD pins, the "Signal Name" column lists the corresponding cell names. The LSI has the separated VDD/VSS pins for internal circuits and I/O buffers to reduce noise and to enable precise measurements of power consumption or electromagnetic emission while computing cryptographic algorithms.

Table 2.1 I/O Signals

Type (No. of signals)	Signal name	No. of signals	Active H/L	Direction	Purpose/Description
System (11)	CLKA	1	--	IN	Clock input of 24 MHz for intra-LSI circuit. Must be the same as CLKB or higher frequency.
	CLKB	1	--	IN	Clock input for LSI interface circuit
	HRST_N	1	L	IN	Reset signal generated by on-board reset circuit. Asynchronous reset input.
	LEDO[1:0]	2	L	OUT	LED driver outputs (NC pins)
	SWIN[3:0]	4	--	IN	Switch inputs (NC pins)
	PHIN[1:0]	2	--	IN	Pin header inputs (NC pins)
Bus control (4)	WR_N	1	L	IN	Write
	RD_N	1	L	IN	Read
	RSV0	1	--	IN	(NC pins)
	RSV1	1	--	IN	(NC pins)
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
Evaluation (13)	START_N	1	L	OUT	Start of target operation
	END_N	1	L	OUT	End of target operation
	(TRIG0)	1	--	OUT	(NC pins)
	(TRIG1)	1	--	OUT	(NC pins)
	EXEC	1	H	OUT	Indicates target in processing
	STATE[3:0]	4	--	OUT	Indicates selected IP
	MON[3:0]	4	--	OUT	Internal monitor use (TBD)
Total		77			

Table 2.2 Pin Assignments of the 130-nm LSI (1/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
1	PVSS1DGZ					core GND
2	PVSS1DGZ					core GND
3	PVSS2DGZ					I/O GND
4	(SWIN[3])					N.C
5	(SWIN[2])					N.C
6	(SWIN[1])					N.C
7	(SWIN[0])					N.C
8	PVDD2POC					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	PVSS2DGZ					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	PVDD2DGZ					I/O 3.3V
20	PVDD1DGZ					core 1.2V
21	PVSS1DGZ					core GND
22	PVSS2DGZ					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	PVSS2DGZ					I/O GND
29	A[15]	I	3.3V		PDIDGZ	Address Bus
30	A[14]	I	3.3V		PDIDGZ	Address Bus
31	A[13]	I	3.3V		PDIDGZ	Address Bus
32	A[12]	I	3.3V		PDIDGZ	Address Bus
33	PVDD2DGZ					I/O 3.3V
34	A[11]	I	3.3V		PDIDGZ	Address Bus
35	A[10]	I	3.3V		PDIDGZ	Address Bus
36	A[9]	I	3.3V		PDIDGZ	Address Bus
37	A[8]	I	3.3V		PDIDGZ	Address Bus
38	PVSS2DGZ					I/O GND
39	PVSS1DGZ					core GND
40	PVSS1DGZ					core GND

Table 2.2 Pin Assignments of the 130-nm LSI (2/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
41	PVDD1DGZ					core 1.2V
42	PVDD2DGZ					I/O 3.3V
43	A[7]	I	3.3V		PDIDGZ	Address Bus
44	A[6]	I	3.3V		PDIDGZ	Address Bus
45	A[5]	I	3.3V		PDIDGZ	Address Bus
46	A[4]	I	3.3V		PDIDGZ	Address Bus
47	PVSS2DGZ					I/O GND
48	PVDD1DGZ					core 1.2V
49	A[3]	I	3.3V		PDIDGZ	Address Bus
50	A[2]	I	3.3V		PDIDGZ	Address Bus
51	A[1]	I	3.3V		PDIDGZ	Address Bus
52	A[0]	I	3.3V		PDIDGZ	Address Bus
53	PVDD2DGZ					I/O 3.3V
54	PVSS1DGZ					core GND
55	PVSS2DGZ					I/O GND
56	CLKB	I	3.3V		PDISDGZ	Clock. Schmitt
57	PVSS2DGZ					I/O GND
58	CLKA	I	3.3V		PDISDGZ	Clock. Schmitt
59	PVSS2DGZ					I/O GND
60	PVDD1DGZ					core 1.2V
61	PVSS1DGZ					core GND
62	PVSS2DGZ					I/O GND
63	HRST_N	I	3.3V		PDISDGZ	Reset. Schmitt
64	PVSS2DGZ					I/O GND
65	WR_N	I	3.3V		PDIDGZ	Write
66	RD_N	I	3.3V		PDIDGZ	Read
67	PVDD2DGZ					I/O 3.3V
68	PVSS1DGZ					core GND
69	DO[15]	O	3.3V	8mA	PDO08CDG	Output Data
70	DO[14]	O	3.3V	8mA	PDO08CDG	Output Data
71	DO[13]	O	3.3V	8mA	PDO08CDG	Output Data
72	DO[12]	O	3.3V	8mA	PDO08CDG	Output Data
73	PVSS2DGZ					I/O GND
74	PVDD1DGZ					core 1.2V
75	DO[11]	O	3.3V	8mA	PDO08CDG	Output Data
76	DO[10]	O	3.3V	8mA	PDO08CDG	Output Data
77	DO[9]	O	3.3V	8mA	PDO08CDG	Output Data
78	DO[8]	O	3.3V	8mA	PDO08CDG	Output Data
79	PVDD2DGZ					I/O 3.3V
80	PVDD1DGZ					core 1.2V

Table 2.2 Pin Assignments of the 130-nm LSI (3/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
81	PVSS1DGZ					core GND
82	PVSS1DGZ					core GND
83	PVSS2DGZ					I/O GND
84	DO[7]	O	3.3V	8mA	PDO08CDG	Output Data
85	DO[6]	O	3.3V	8mA	PDO08CDG	Output Data
86	DO[5]	O	3.3V	8mA	PDO08CDG	Output Data
87	DO[4]	O	3.3V	8mA	PDO08CDG	Output Data
88	PVDD2DGZ					I/O 3.3V
89	DO[3]	O	3.3V	8mA	PDO08CDG	Output Data
90	DO[2]	O	3.3V	8mA	PDO08CDG	Output Data
91	DO[1]	O	3.3V	8mA	PDO08CDG	Output Data
92	DO[0]	O	3.3V	8mA	PDO08CDG	Output Data
93	PVSS2DGZ					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	PVDD2DGZ					I/O 3.3V
100	PVDD1DGZ					core 1.2V
101	PVSS1DGZ					core GND
102	PVSS2DGZ					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	PVSS2DGZ					I/O GND
109	DI[0]	I	3.3V		PDIDGZ	Input Data
110	DI[1]	I	3.3V		PDIDGZ	Input Data
111	DI[2]	I	3.3V		PDIDGZ	Input Data
112	DI[3]	I	3.3V		PDIDGZ	Input Data
113	PVDD2DGZ					I/O 3.3V
114	DI[4]	I	3.3V		PDIDGZ	Input Data
115	DI[5]	I	3.3V		PDIDGZ	Input Data
116	DI[6]	I	3.3V		PDIDGZ	Input Data
117	DI[7]	I	3.3V		PDIDGZ	Input Data
118	PVSS2DGZ					I/O GND
119	PVSS1DGZ					core GND
120	PVSS1DGZ					core GND

Table 2.2 Pin Assignments of the 130-nm LSI (4/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
121	PVDD1DGZ					core 1.2V
122	PVDD2DGZ					I/O 3.3V
123	DI[8]	I	3.3V		PDIDGZ	Input Data
124	DI[9]	I	3.3V		PDIDGZ	Input Data
125	DI[10]	I	3.3V		PDIDGZ	Input Data
126	DI[11]	I	3.3V		PDIDGZ	Input Data
127	PVSS2DGZ					I/O GND
128	PVDD1DGZ					core 1.2V
129	DI[12]	I	3.3V		PDIDGZ	Input Data
130	DI[13]	I	3.3V		PDIDGZ	Input Data
131	DI[14]	I	3.3V		PDIDGZ	Input Data
132	DI[15]	I	3.3V		PDIDGZ	Input Data
133	PVDD2DGZ					I/O 3.3V
134	PVSS1DGZ					core GND
135	(LED[0])					N.C
136	(LED[1])					N.C
137	END_N	O	3.3V	8mA	PDO08CDG	End of Operation
138	START_N	O	3.3V	8mA	PDO08CDG	Start of Operation
139	PVSS2DGZ					I/O GND
140	PVDD1DGZ					core 1.2V
141	PVSS1DGZ					core GND
142	PVSS2DGZ					I/O GND
143	STATE[0]	O	3.3V	8mA	PDO08CDG	Selected IP
144	STATE[1]	O	3.3V	8mA	PDO08CDG	Selected IP
145	STATE[2]	O	3.3V	8mA	PDO08CDG	Selected IP
146	STATE[3]	O	3.3V	8mA	PDO08CDG	Selected IP
147	PVDD2DGZ					I/O 3.3V
148	PVSS1DGZ					core GND
149	(MON[0])					N.C
150	(MON[1])					N.C
151	(MON[2])					N.C
152	(MON[3])					N.C
153	PVSS2DGZ					I/O GND
154	PVDD1DGZ					core 1.2V
155	EXEC	O	3.3V	8mA	PDO08CDG	In Processing
156	N.C					N.C
157	N.C					N.C
158	N.C					N.C
159	PVDD2DGZ					I/O 3.3V
160	PVDD1DGZ					core 1.2V

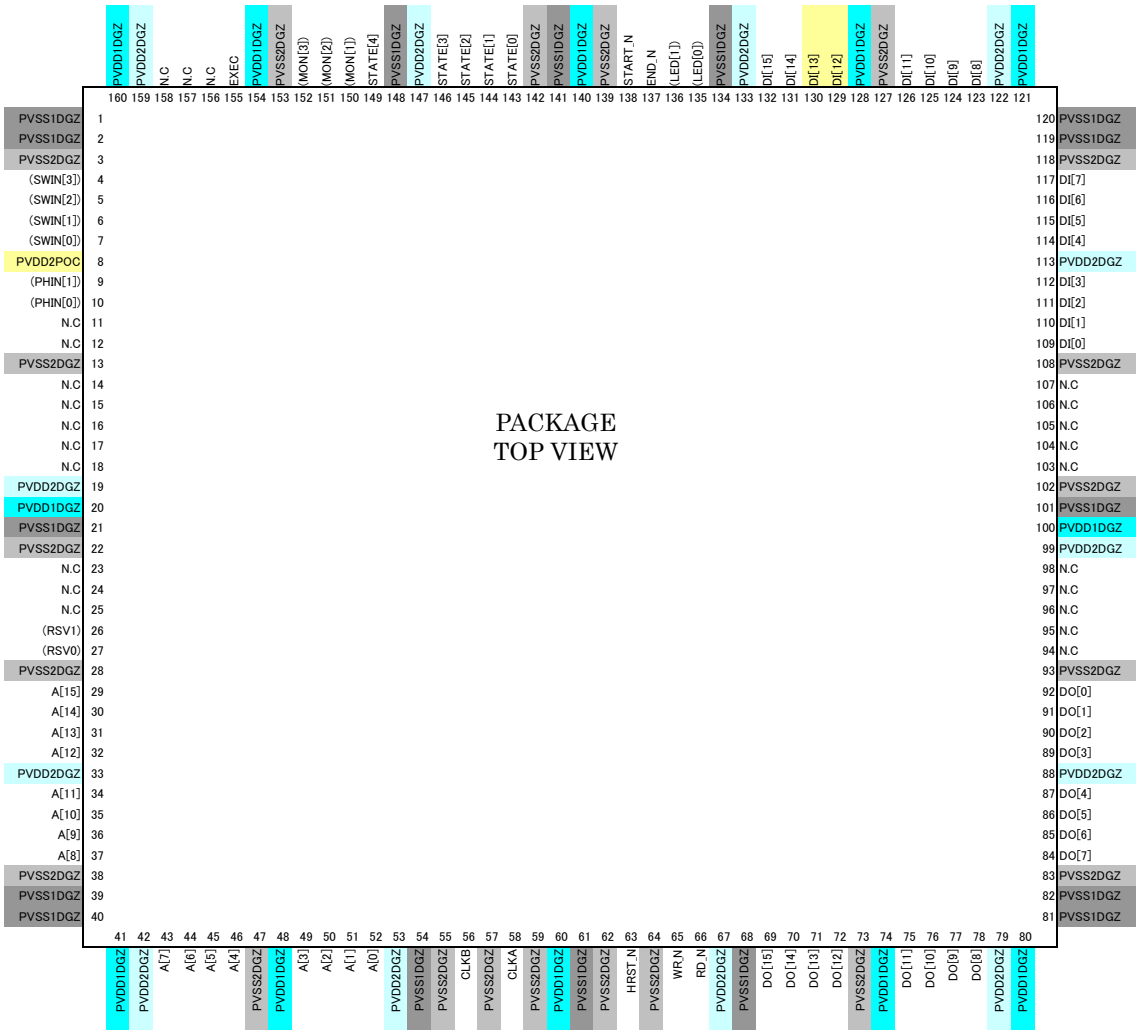


Figure 2.1　Pin Assignment Image of the 130nm Cryptographic LSI

Table 2.3 Pin Assignment of the 90-nm LSI (1/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
1	PVSS1CDG_33					core GND
2	PVSS1CDG_33					core GND
3	PVSS2CDG_33					I/O GND
4	(SWIN[3])					N.C
5	(SWIN[2])					N.C
6	(SWIN[1])					N.C
7	(SWIN[0])					N.C
8	PVDD2POC_33					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	PVSS2CDG_33					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	PVDD2CDG_33					I/O 3.3V
20	PVDD1CDG_33					core 1.0V
21	PVSS1CDG_33					core GND
22	PVSS2CDG_33					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	PVSS2CDG_33					I/O GND
29	A[15]	I	3.3V		PDC0816CDG_33	Address Bus
30	A[14]	I	3.3V		PDC0816CDG_33	Address Bus
31	A[13]	I	3.3V		PDC0816CDG_33	Address Bus
32	A[12]	I	3.3V		PDC0816CDG_33	Address Bus
33	PVDD2CDG_33					I/O 3.3V
34	A[11]	I	3.3V		PDC0816CDG_33	Address Bus
35	A[10]	I	3.3V		PDC0816CDG_33	Address Bus
36	A[9]	I	3.3V		PDC0816CDG_33	Address Bus
37	A[8]	I	3.3V		PDC0816CDG_33	Address Bus
38	PVSS2CDG_33					I/O GND
39	PVSS1CDG_33					core GND
40	PVSS1CDG_33					core GND

Table 2.3 Pin Assignment of the 90-nm LSI (2/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
41	PVDD1CDG_33					core 1.0V
42	PVDD2CDG_33					I/O 3.3V
43	A[7]	I	3.3V		PDC0816CDG_33	Address Bus
44	A[6]	I	3.3V		PDC0816CDG_33	Address Bus
45	A[5]	I	3.3V		PDC0816CDG_33	Address Bus
46	A[4]	I	3.3V		PDC0816CDG_33	Address Bus
47	PVSS2CDG_33					I/O GND
48	PVDD1CDG_33					core 1.2V
49	A[3]	I	3.3V		PDC0816CDG_33	Address Bus
50	A[2]	I	3.3V		PDC0816CDG_33	Address Bus
51	A[1]	I	3.3V		PDC0816CDG_33	Address Bus
52	A[0]	I	3.3V		PDC0816CDG_33	Address Bus
53	PVDD2CDG_33					I/O 3.3V
54	PVSS1CDG_33					core GND
55	PVSS2CDG_33					I/O GND
56	CLKB	I	3.3V		PDS0816CDG_33	Clock. Schmitt
57	PVSS2CDG_33					I/O GND
58	CLKA	I	3.3V		PDS0816CDG_33	Clock. Schmitt
59	PVSS2CDG_33					I/O GND
60	PVDD1CDG_33					core VDD (1.2V/1.0V)
61	PVSS1CDG_33					core GND
62	PVSS2CDG_33					I/O GND
63	HRST_N	I	3.3V		PDS0816CDG_33	Reset. Schmitt
64	PVSS2CDG_33					I/O GND
65	WR_N	I	3.3V		PDC0816CDG_33	Write
66	RD_N	I	3.3V		PDC0816CDG_33	Read
67	PVDD2CDG_33					I/O 3.3V
68	PVSS1CDG_33					core GND
69	DO[15]	O	3.3V	8mA	PDC0816CDG_33	Output Data
70	DO[14]	O	3.3V	8mA	PDC0816CDG_33	Output Data
71	DO[13]	O	3.3V	8mA	PDC0816CDG_33	Output Data
72	DO[12]	O	3.3V	8mA	PDC0816CDG_33	Output Data
73	PVSS2CDG_33					I/O GND
74	PVDD1CDG_33					core VDD (1.2V/1.0V)
75	DO[11]	O	3.3V	8mA	PDC0816CDG_33	Output Data
76	DO[10]	O	3.3V	8mA	PDC0816CDG_33	Output Data
77	DO[9]	O	3.3V	8mA	PDC0816CDG_33	Output Data
78	DO[8]	O	3.3V	8mA	PDC0816CDG_33	Output Data
79	PVDD2CDG_33					I/O 3.3V
80	PVDD1CDG_33					core VDD (1.2V/1.0V)

Table 2.3 Pin Assignment of the 90-nm LSI (3/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
81	PVSS1CDG_33					core GND
82	PVSS1CDG_33					core GND
83	PVSS2CDG_33					I/O GND
84	DO[7]	O	3.3V	8mA	PDC0816CDG_33	Output Data
85	DO[6]	O	3.3V	8mA	PDC0816CDG_33	Output Data
86	DO[5]	O	3.3V	8mA	PDC0816CDG_33	Output Data
87	DO[4]	O	3.3V	8mA	PDC0816CDG_33	Output Data
88	PVDD2CDG_33					I/O 3.3V
89	DO[3]	O	3.3V	8mA	PDC0816CDG_33	Output Data
90	DO[2]	O	3.3V	8mA	PDC0816CDG_33	Output Data
91	DO[1]	O	3.3V	8mA	PDC0816CDG_33	Output Data
92	DO[0]	O	3.3V	8mA	PDC0816CDG_33	Output Data
93	PVSS2CDG_33					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	PVDD2CDG_33					I/O 3.3V
100	PVDD1CDG_33					core VDD 1.0V
101	PVSS1CDG_33					core GND
102	PVSS2CDG_33					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	PVSS2CDG_33					I/O GND
109	DI[0]	I	3.3V		PDC0816CDG_33	Input Data
110	DI[1]	I	3.3V		PDC0816CDG_33	Input Data
111	DI[2]	I	3.3V		PDC0816CDG_33	Input Data
112	DI[3]	I	3.3V		PDC0816CDG_33	Input Data
113	PVDD2CDG_33					I/O 3.3V
114	DI[4]	I	3.3V		PDC0816CDG_33	Input Data
115	DI[5]	I	3.3V		PDC0816CDG_33	Input Data
116	DI[6]	I	3.3V		PDC0816CDG_33	Input Data
117	DI[7]	I	3.3V		PDC0816CDG_33	Input Data
118	PVSS2CDG_33					I/O GND
119	PVSS1CDG_33					core GND
120	PVSS1CDG_33					core GND

Table 2.3 Pin Assignment of the 90-nm LSI (4/4)

Pin No	Signal name	I/O	I/F voltage	Output	I/O buffer	Function
121	PVDD1CDG_33					core 1.0V
122	PVDD2CDG_33					I/O 3.3V
123	DI[8]	I	3.3V		PDC0816CDG_33	Input Data
124	DI[9]	I	3.3V		PDC0816CDG_33	Input Data
125	DI[10]	I	3.3V		PDC0816CDG_33	Input Data
126	DI[11]	I	3.3V		PDC0816CDG_33	Input Data
127	PVSS2CDG_33					I/O GND
128	PVDD1CDG_33					core 1.0V
129	DI[12]	I	3.3V		PDC0816CDG_33	Input Data
130	DI[13]	I	3.3V		PDC0816CDG_33	Input Data
131	DI[14]	I	3.3V		PDC0816CDG_33	Input Data
132	DI[15]	I	3.3V		PDC0816CDG_33	Input Data
133	PVDD2CDG_33					I/O 3.3V
134	PVSS1CDG_33					core GND
135	(LED[0])					N.C
136	(LED[1])					N.C
137	END_N	O	3.3V	8mA	PDC0816CDG_33	End of Operation
138	START_N	O	3.3V	8mA	PDC0816CDG_33	Start of Operation
139	PVSS2CDG_33					I/O GND
140	PVDD1CDG_33					core 1.0V
141	PVSS1CDG_33					core GND
142	PVSS2CDG_33					I/O GND
143	STATE[0]	O	3.3V	8mA	PDC0816CDG_33	Selected IP
144	STATE[1]	O	3.3V	8mA	PDC0816CDG_33	Selected IP
145	STATE[2]	O	3.3V	8mA	PDC0816CDG_33	Selected IP
146	STATE[3]	O	3.3V	8mA	PDC0816CDG_33	Selected IP
147	PVDD2CDG_33					I/O 3.3V
148	PVSS1CDG_33					core GND
149	STATE[4]				PDC0816CDG_33	N.C
150	(MON[1])					N.C
151	(MON[2])					N.C
152	(MON[3])					N.C
153	PVSS2CDG_33					I/O GND
154	PVDD1CDG_33					core 1.0V
155	EXEC	O	3.3V	8mA	PDC0816CDG_33	In Processing
156	N.C					N.C
157	N.C					N.C
158	N.C					N.C
159	PVDD2CDG_33					I/O 3.3V
160	PVDD1CDG_33					core 1.0V

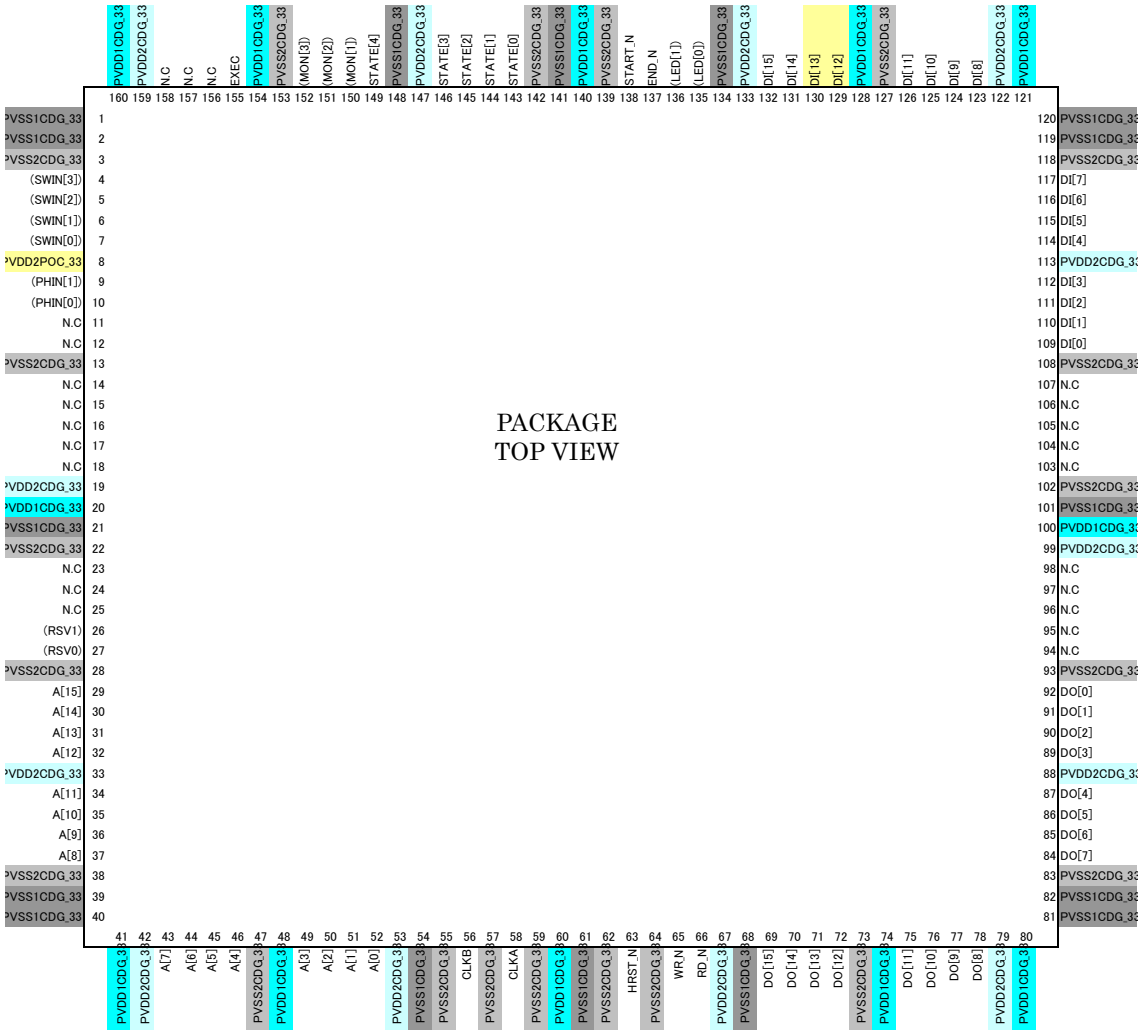


Figure 2.2 Pin Assignment Image of the 90nm Cryptographic LSI

2.2 Control Interface

Table 2.4 lists the interface registers and address map of the cryptographic LSI. Figure 2.3, Figure 2.4, and Figure 2.5 chart the data read and write timings on the registers. Through the interface registers, the sequence described below controls each cryptographic IP core. Refer to Section 3.3 for a detailed description of the interface registers.

- Cryptographic algorithm cores other than AES5
  - (a) Operating IP selection : Set the corresponding bits on the IP selection register (IPSEL0, 1).
  - (b) Selected IP reset : Write 1 then 0 to CONT[IPRST].
  - (c) Output IP selection : Set the corresponding bit on the output selection register (OUTSEL0, 1).
  - (d) Mode setting : Set desired operating modes on the mode register(MODE). (*1)
  - (e) Key setting :
    1. Set KEY0~7 for the common-key ciphers, EXP0~31 and MOD0~31 for RSA, or IDATA0~3 for ECC.
    2. Set CONT[KSET], then wait until this bit is cleared.
  - (f) Initial value (IV) setting : Set IV0~7. (*2)

(g) Random number (seed) setting : Set RAND0~7. (*3)
(h) Cryptographic operation : Repeat the following sequence.
1. Set ITEXT0~7(*4) for the common-key ciphers, IDATA0~31 for RSA, and IDATA8~19 for ECC.
2. Set CONT[RUN], then wait until this bit is cleared.
3. Read OTEXT0~7(*5) for the common-key ciphers, ODATA0~31 for RSA, and ODATA0~3 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.

When AES6 is selected, the intermediate value either at the round designated in the round selection register or at fault is accessible on RDATA0~7. Likely, the intermediate key is accessible on RKEY0~7.

Settings can be changed as follows:

- To change the cryptographic core, perform the sequence from (a) through (h) again.
- To change the operating modes of the already selected cryptographic core, perform the sequence from (d) through (h) again.
- To change the key of the already selected cryptographic core, perform the sequence from (e) through (h) again.
- To change the initial value of the already selected cryptographic core, perform the sequence from (f) through (h) again.
- To change the random number of the already selected cryptographic core, perform the sequence from (g) through (h) again.

- AES5 (CTR mode + 4-stage pipeline implementation)

(a)~(e) Same as the previous sequence.
(f) Initial value (IV) setting
1. Set IV0~7.
2. Set 1 to the control register CONT[RUN], then wait until this bit is cleared.
(g) Random number (seed) setting : No need to set.
(h) Cryptographic operation : Repeat the following sequence.
1. Set ITEXT0~31.
2. Set 1 to the control register CONT[RUN], then wait until this bit is cleared.
3. Read OTEXT0~31.

To change the initial value, perform the sequence from (f) through (h) again.

Table 2.4 Interface Registers (1/3)

Type	Address	Register name	Mnemonic	R/W	Function/Description
System control	0000	(Reserved)		—	
	0002	Control register	CONT	R/W	Starts operation(W)/Notifies completion(R). Starts key generation(W) / Notifies completion(R). Controls resetting the cryptographic IP(W).
	0004	IP selection register 0	IPSEL0	R/W	Designates the operating cryptographic IP.
	0006	IP selection register 1	IPSEL1	R/W	Designates the operating cryptographic IP.
	0008	Output selection register 0	OUTSEL0	R/W	Designates the data-exporting cryptographic IP.
	000A	Output selection register 1	OUTSEL1	R/W	Designates the data-exporting cryptographic IP.
	000C	Mode register	MODE	R/W	Designates the operating modes, key length, and encryption/decryption.
	000E	Round selection register	RSEL	R/W	Designates the intermediate value keeping round.
	0010	Test register 1	TEST1	R/W	Custom core operation control 1
	0012	Test register 2	TEST2	R/W	Custom core operation control 2
	⋮ 00FE	(Reserved)			

Table 2.4 Interface Registers (2/3)

Type		Address	Register name	Mnemonic	R/W	Function/Description
Common-key cryptography	Secret key (→LSI)	0100	Key register 0	KEY0	W	Common-key algorithm's key (Top 16 bits)
		0102	Key register 1	KEY1	W	Common-key algorithm's key (Next 16 bits to KEY0)
		:	:	:	:	:
		010E	Key register 7	KEY7	W	Common-key algorithm's key (Bottom 16 bits)
	IV (→LSI)	0110	IV register 0	IV0	W	Designates IV (Top 16 bits).
		0112	IV register 1	IV1	W	Designates IV (next 16 bits to IV0).
		:	:	:	:	:
		011E	IV register 7	IV7	W	Designates IV (Bottom 16 bits).
	Input text (→LSI)	0120	Input text register 0	ITEXT0	W	Designates input text (Top 16 bits)
		0122	Input text register 1	ITEXT1	W	Designates input text (Next 16 bits to ITEXT0)
		:	:	:	:	:
		015E	Input text register 31	ITEXT31	W	Designates input text (Bottom 16 bits)
	Random number (→LSI)	0160	Random number register 0	RAND0	W	Designates random number (Top 16 bits)
		0162	Random number register 1	RAND1	W	Designates random number (Next 16 bits to RAND0)
		:	:	:	:	:
		016E	Random number register 7	RAND7	W	Designates random number (Bottom 16 bits)
	(Reserved)	: 017E	(Reserved)			
	Output text (←LSI)	0180	Output text register 0	OTEXT0	R	Reads output text (Top 16 bits)
		0182	Output text register 1	OTEXT1	R	Reads output text (Next 16 bits to OTEXT0)
		:	:	:	:	:
		01BE	Output text register 31	OTEXT31	R	Reads output text (Bottom 16 bits)
	Intermediate data (←LSI)	01C0	Intermediate data register 0	RDATA0	R	Reads intermediate data (Top 16 bits)
		01C2	Intermediate data register 1	RDATA1	R	Reads intermediate data (Next 16 bits to RDATA0)
		:	:	:	:	:
		01CE	Intermediate data register 7	RDATA7	R	Reads intermediate data (Bottom 16 bits)
	Intermediate key (←LSI)	01D0	Intermediate key register 0	RKEY0	R	Reads intermediate key (Top 16 bits)
		01D2	Intermediate key register 1	RKEY1	R	Reads intermediate key (Next 16 bits to RKEY0)
		:	:	:	:	:
		01DE	Intermediate key register 7	RKEY7	R	Reads intermediate key (Bottom 16 bits)
	(Reserved)	: 01FE	(Reserved)			

Table 2.4 Interface Registers (3/3)

Type		Address	Register Name	Mnemonic	R/W	Function/Description
Public-key Cryptography	Exponent /Key (←LSI)	0200	Exponent register 0	EXP0	W	Sets exponent (Top 16 bits)
		0202	Exponent register 1	EXP1	W	Sets exponent (Next 16 bits to EXP00)
		:	:	:	:	:
		023E	Exponent register 31	EXP31	W	Sets exponent (Bottom 16 bits)
		:	:	:	:	:
	Modulus (→LSI)	0300	Modulus register 0	MOD0	W	Sets modulus (Top 16 bits)
		0302	Modulus register 1	MOD1	W	Sets modulus (Next 16 bits to MOD00)
		:	:	:	:	:
		033E	Modulus register 31	MOD31	W	Sets modulus (Bottom 16 bits)
	Preprocessed data input (→LSI)	0340	Preprocessed data register 0	PREDAT0	W	Reads preprocessed data (Top 16 bits)
		0342	Preprocessed data register 1	PREDAT1	W	Reads preprocessed data (Next 16 bits to PREDAT00)
		:	:	:	:	:
		035E	Preprocessed data register 15	PREDAT15	W	Reads preprocessed data (Bottom 16 bits)
		: 03FE	(Reserved)			
	Input data (→LSI)	0400	Input data register 0	IDATA0	W	Sets input data (Top 16 bits)
		0402	Input data register 1	IDATA1	W	Sets input data (Next 16 bits to IDATA00)
		:	:	:	:	:
		043E	Input data register 31	IDATA31	W	Sets input data (Bottom 16 bits)
		: 04FE	(Reserved)			
	Output data (←LSI)	0500	Output data register 0	ODATA0	R	Reads output data (Top 16 bits)
		0502	Output data register 1	ODATA1	R	Reads output data (Next 16 bits to ODATA00)
		:	:	:	:	:
		053E	Output data register 31	ODATA31	R	Reads output data (Bottom 16 bits)
		: 05FE	(Reserved)			
(Reserved)		0600 : FFEE				
LSI information (0xFFF0 ~0xFFFF)		FFF0 :	(Reserved)			
		FFFC	Version register	VER	R	
		FFFE	(Reserved)		- -	

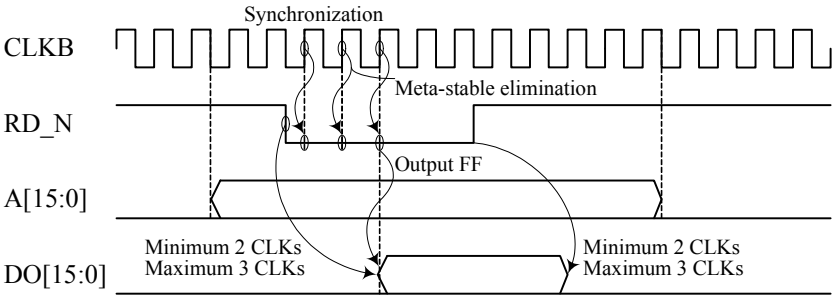


Figure 2.3 : Timing Chart of the Read Cycle

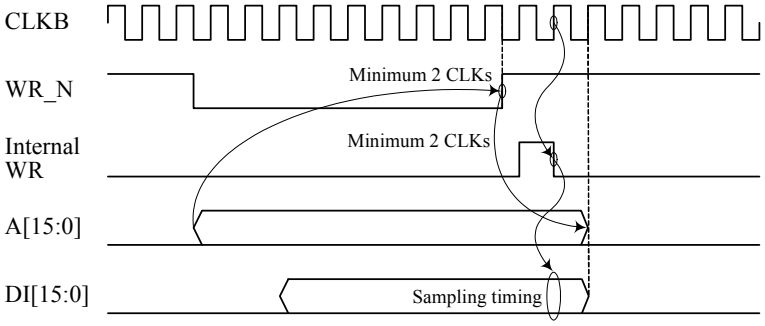
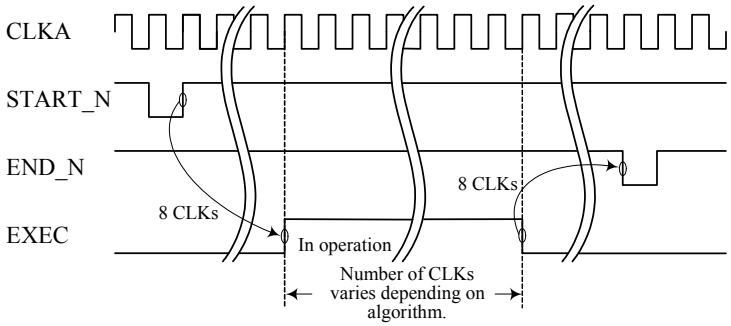


Figure 2.4 Timing Chart of the Write Cycle



START_N: Start of algorithm operation (Active L)
END_N: End of algorithm operation (Active L)
EXEC: In operation (Active H)

Figure 2.5 Timing Chart of the Cryptographic Operations

3. INTERNAL SPECIFICATIONS

3.1 Internal Structure of LSI

Figure 3.1 illustrates the block diagram of the cryptographic LSI. Figure 3.2 shows the hierarchical structure of the source code for each cryptographic IP. The cryptographic LSI consists of the 22 cryptographic IP cores listed in Table 3.1 and the interface circuit.

Table 3.1 Cryptographic IP Cores

IP No.	IP core	HDL source top module	Description
1	AES0 (Composite field S-box)	AES_Comp	AES with composite field S-box. Supports both encryption and decryption with 128-bit key.
2	AES1 (Table S-box)	AES_TBL	AES with S-box described with case statements. Supports only encryption with 128-bit key.
3	AES2 (1-stage PPRM S-box)	AES_PPRM1	AES with PPRM(Positive Prime Reed-Muler)-based S-box using single-stage AND-XOR logic. Supports only encryption with 128-bit key.
4	AES3 (3-stage PPRM S-box)	AES_PPRM3	PPRM-based S-box using 3-stage AND-XOR logic. Supports only encryption with 128-bit key.
5	AES4 (Composite field S-box)	AES_Comp_ENC_top	AES including only the encryption part extracted from AES_Comp.AES with
6	AES5 (CTR mode)	AES_CTR_PIPE	AES with 4-stage pipeline supporting the CTR mode. Used composite field S-box.
7	AES6 (FA countermeasure implemented)	AES_FA	AES with FA(Fault injection Attack) countermeasure with internal data error detection. Supports both encryption and decryption. Used composite field S-Box.
8	AES7 (Round key preprocess)	AES_PKG	AES that preprocesses the 11 round keys and stores them in a register file.
9	AES8 (MAO)	U_YNU_MA_AESTOP	AES with DPA countermeasure with Masked And Operation (MAO).
10	AES9 (MDPL)	U_YNU_ML_AESTOP	AES with DPA countermeasure with Dual-rail Precharge Logic (MDPL).
11	AES10 (Threshold)	U_YNU_TI_AESTOP	AES with DPA countermeasure with Threshold implementation.
12	AES11 (WDDL)	U_YNU_WL_AESTOP	AES with DPA countermeasure with Wave Dynamic Differential Logic (WDDL).
13	AES12 (Pseudo RSL)	JIP_PR_AESTOP	AES as in AES_Comp_ENC_top except for DPA countermeasure with pseudo RSL(Random Switching Logic) imitated using the standard cell library.
14	AES13 (Pseudo RSL)	JIP_WO_AESTOP	AES using the same RTL source code as AES_Comp. Synthesized into a netlist with a special constraint to contain equivalent nodes as in FPGA(Xilinx Virtex2).
15	Camellia		128-bit block cipher Camellia with case statement based S-box.

IP No.	IP core	HDL source top module	Description
16	CAST-128		CAST128, a 64-bit block cipher with 128-bit key.
17	DES		Single DES, known as a 64-bit block cipher with 56-bit key.
18	ECC		ECC with scalar multiplication on an elliptic curve over GF(2^{64}).
19	MISTY1		MISTY1, a 64-bit block cipher with 128-bit key. Adopted the case statement approach for S-box S7 and S9.
20	RSA		RSA cryptography using the Montgomery multiplication with a 32-bit multiplier.
21	SEED		SEED, a 64-bit block cipher with 128-bit key.
22	TDES		3-key Triple DES

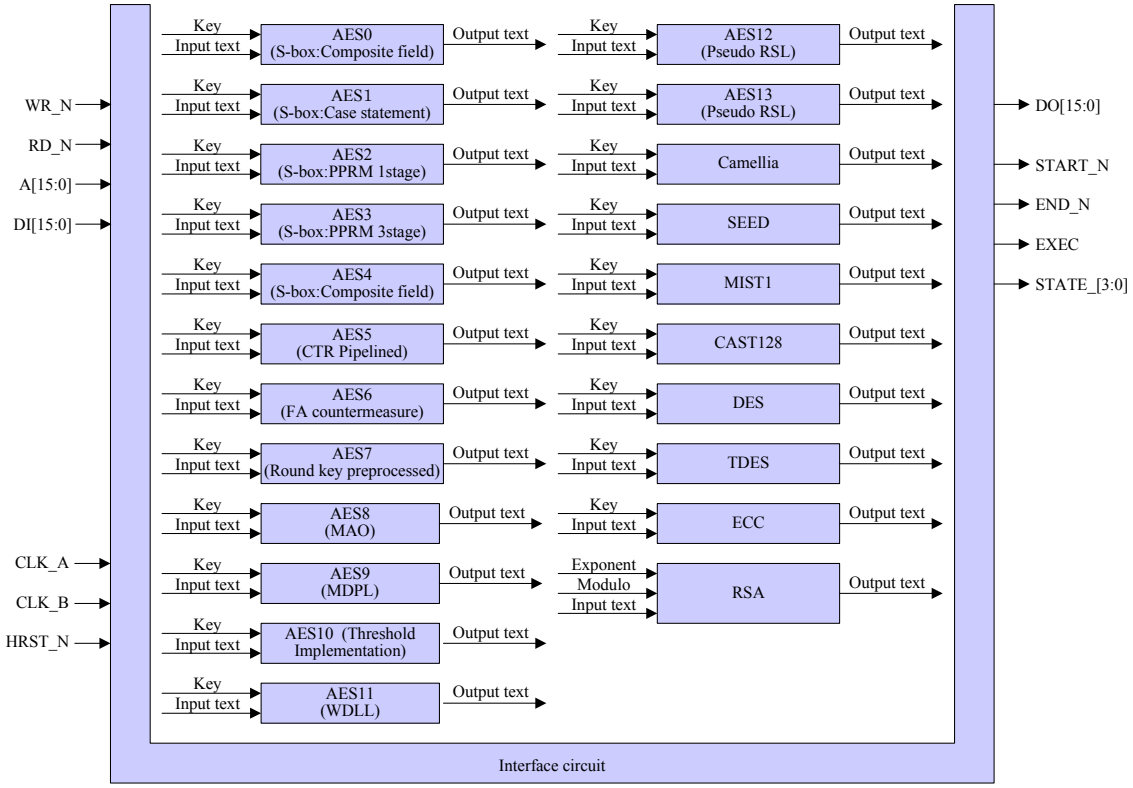


Figure 3.1 Block Diagram of the LSI

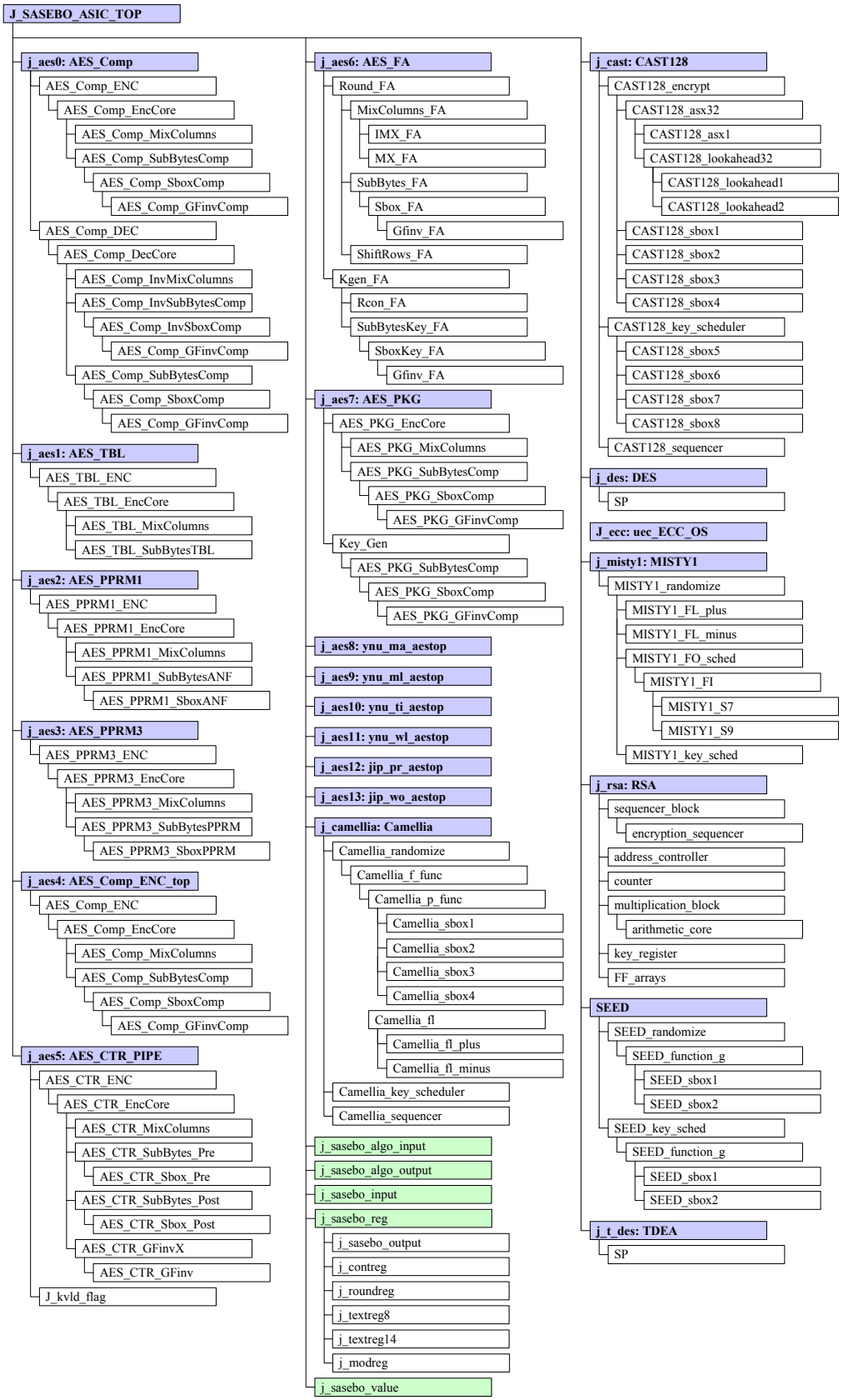
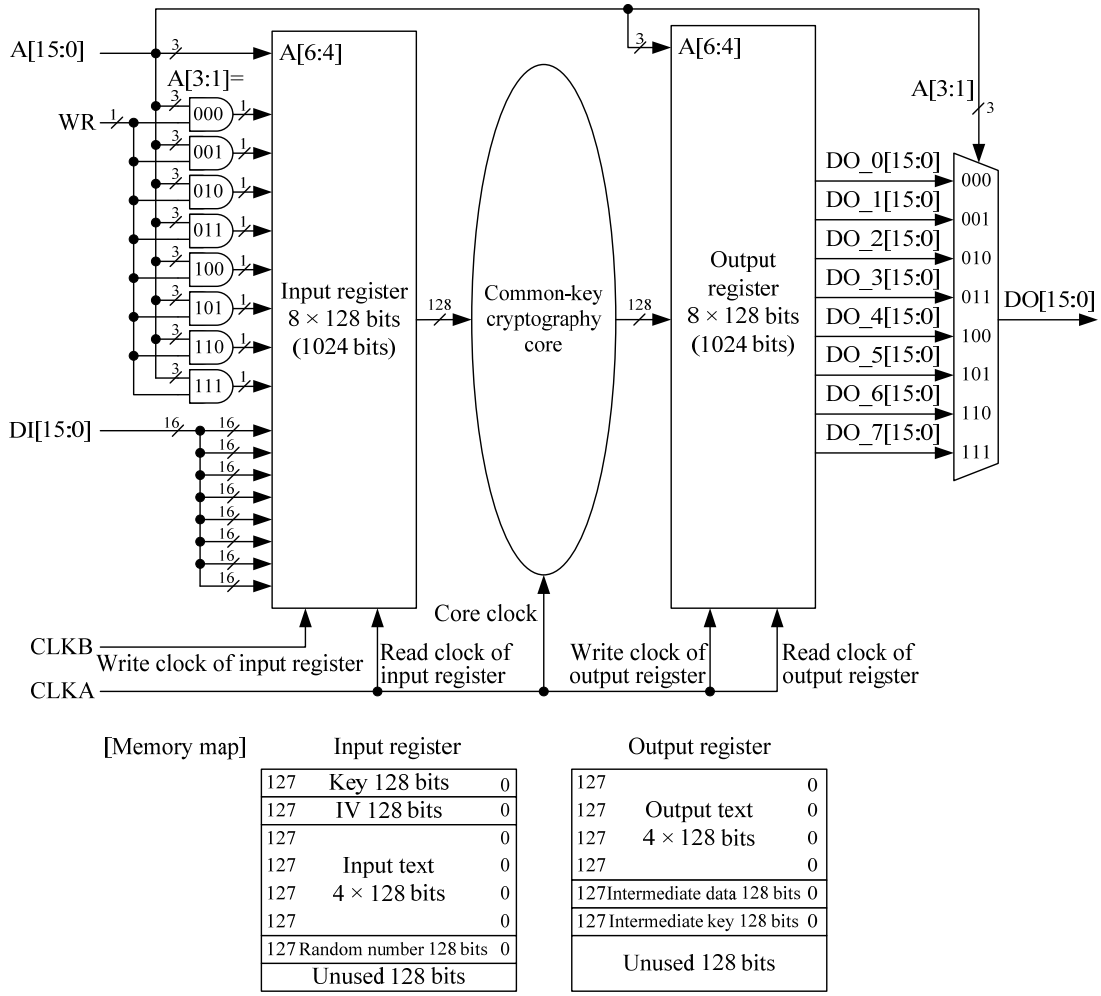


Figure 3.2 Hierarchical Architecture of the Source Code

3.2 Cryptographic Circuit Interface

Figure 3.3 and Figure 3.4 show external interface circuits of common-key cryptographic algorithms (AES, DES, MISTY1, Camellia, SEED, and CAST128) and public-key cryptographic algorithms (RSA and ECC), respectively.



[Memory map]

Input register

127	Key 128 bits	0
127	IV 128 bits	0
127	Input text 4 × 128 bits	0
127	Random number 128 bits	0
	Unused 128 bits	

Output register

127	Output text 4 × 128 bits	0
127	Intermediate data 128 bits	0
127	Intermediate key 128 bits	0
	Unused 128 bits	

Figure 3.3 Interface Circuit of the Common-key Cryptographic Algorithms

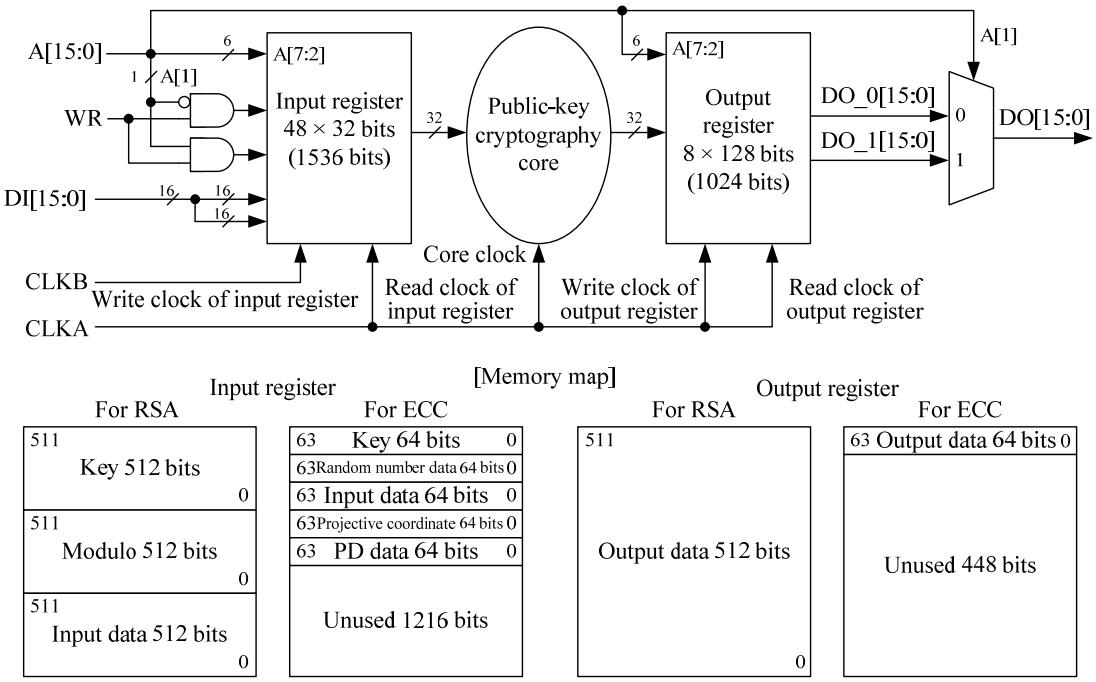


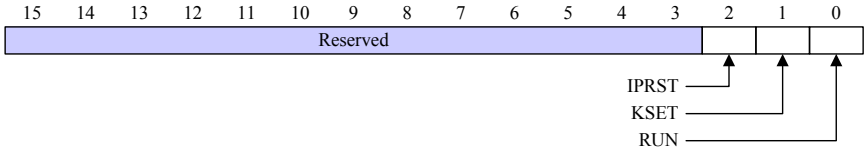
Figure 3.4 Interface Circuit of the Public-key Cryptographic Algorithms

3.3 Interface Registers

This section details the interface registers.

- **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 selection 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 selection 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

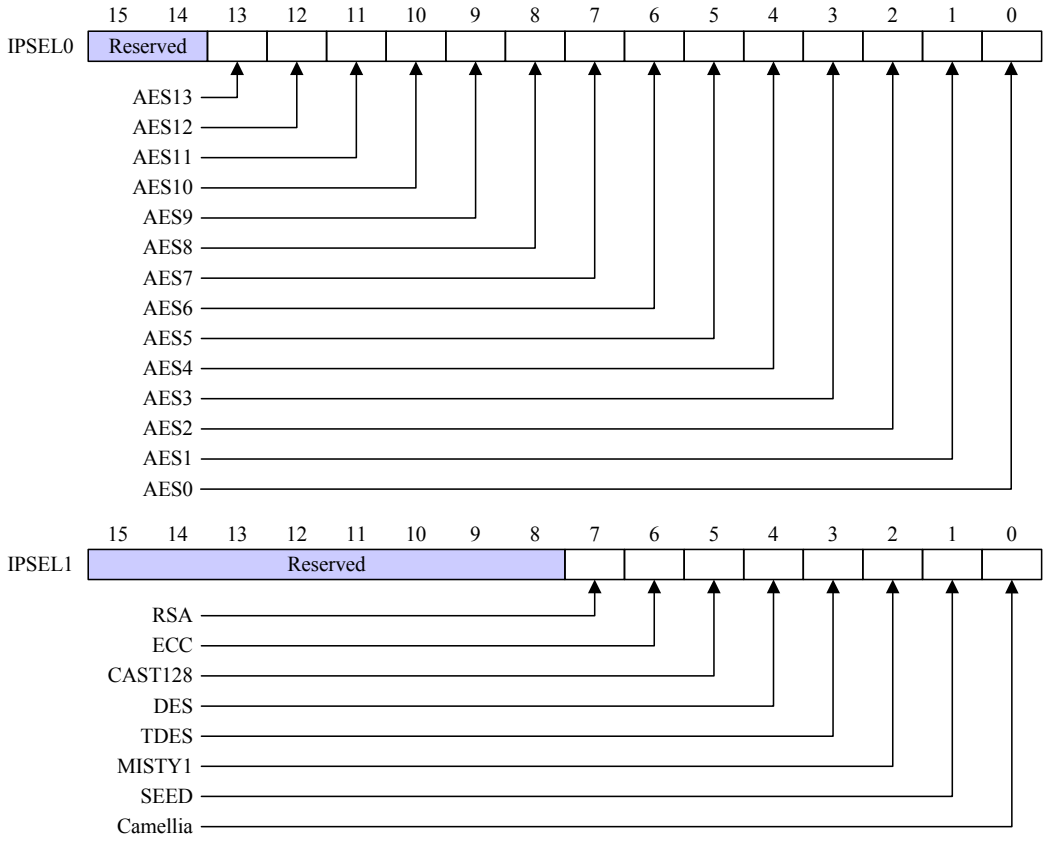
Write ‘1’ to this bit, and a key is generated, in accordance with the mode register (MODE), within the cryptographic IP designated by the IP selection register (IPSEL). When the key generation in the cryptographic IP designated by the output selection register (OUTSEL) is completed and a cryptographic 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’.

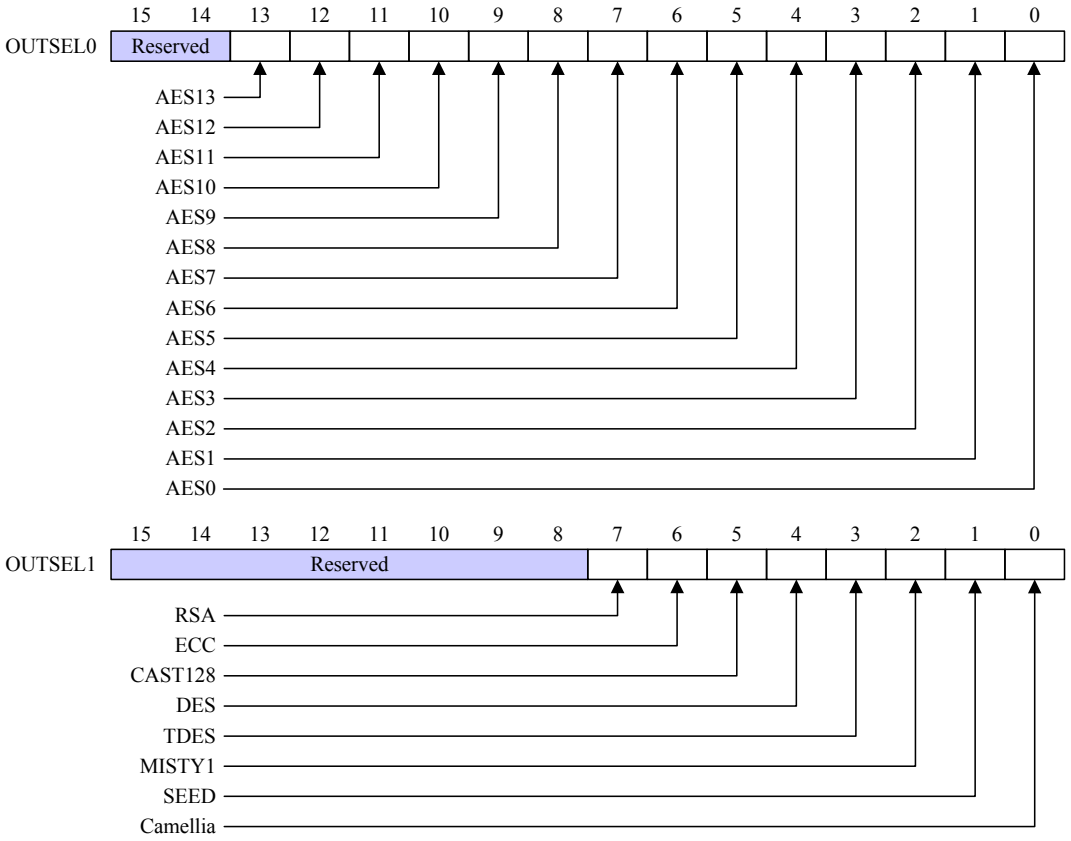
- **IP selection register : IPSEL**

Among the 22 cryptographic IPs, the IPs whose corresponding bits are set to ‘1’ enter the active state. The other IPs are not provided with the clock signal.



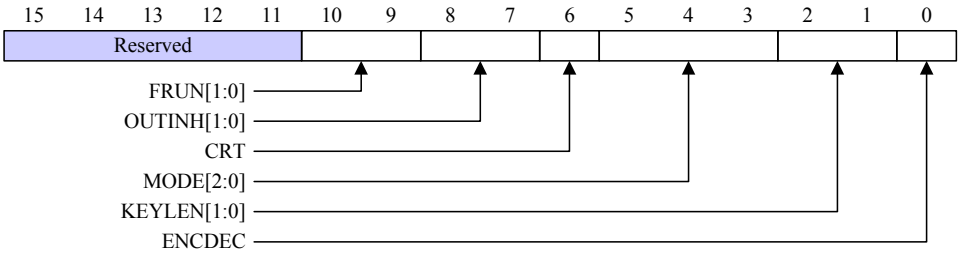
- **Output selection register : OUTSEL**

Write ‘1’ to one of the bits corresponding to the IP selection 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 selection register are set to ‘1’.



- **Mode register : MODE**

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



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	Free-run mode OFF
	1	Free-run 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 common-key cryptographic algorithms, the modes of operation (e.g. ECB and CTR) or operating modes are fixed with the specific values depending on the IP, except for AES12 where the test register TEST2 controls the operating mode. (The TEST2 register's specification is not available.)

Bit 2-1 : KEYLEN[1:0]

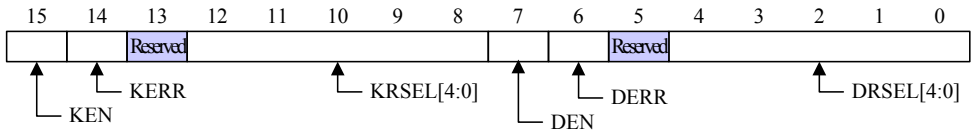
The value is fixed to 00. The actual key length is already specified for each IP.

Bit 0 : ENCDEC

Specifies encryption by 0 or decryption by 1. For the IPs that only support encryption, this bit is not valid.

- **Round selection register : RSEL**

Designates the rounds where the intermediate data register (RDATA0~7) and intermediate key register (RKEY0~7) take the corresponding values. DRSEL, RDATA0~7, KRSEL, and RKEY0~7 are valid only while AES6, which is the cryptographic IP equipped with a fault injection attack countermeasure, is selected.



Bit 15 : KEN

0: Deactivates the intermediate key register by not supplying the clock signal. The register continues to store the intermediate key, but cannot latch a new intermediate key.

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.)

Bit 12-8 : KRSEL[4:0]

Designates the round number where the intermediate key register (RKEY0~7) 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.)

Bit 4-0 : DRSEL[4:0]

Designates the round number where the intermediate data register (RDATA0~7) takes the intermediate data.

- **Test register 1 : TEST1**

Write '1' to Bit 0 of the test register TEST1, and encryption will use an internal key instead of externally applied keys. (The specification for the internal keys is not available.) 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 the 14 AES cryptographic IP cores.

- **Test register 2 : TEST2**

The test register TEST2 is the debugging register for controlling AES12, the AES core with the pseudo RSL countermeasure. The test register functionality is not available.

- **Key register for common-key cryptography: 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:

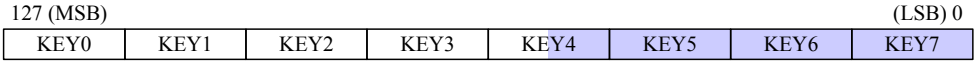
[191:64] : 0x000102030405060708090a0b0c0d0e0f (Fixed value)

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

Others: Of the 128 bits of the key, the upper 72 bits are fixed as shown below, and the rest 56 bits are set in the lower 8 bits of KEY4 and whole KEY5~7 by the user.

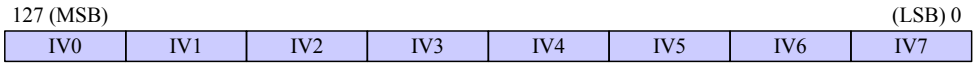
[127:56] : 0x00010203040506070 8

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



- **Initial vector register for the GCM mode: IV0~7**

The initial vector register holds the 128-bit initial vector IV for the GCM mode supported only in AES5.



- **Input text register for common-key cryptography: ITEXT0~31**

The input text register holds the input text used by the IP that the IP selection register IPSEL designates. Note that each cryptographic core has different data size and different input text register positions used as follows:

AES5(AES_CTR_PIPE) :	Takes 128 bits × 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

127 (MSB) (LSB) 0

ITEXT0	ITEXT1	ITEXT2	ITEXT3	ITEXT4	ITEXT5	ITEXT6	ITEX7
ITEXT8	ITEXT9	ITEXT10	ITEXT11	ITEXT12	ITEXT13	ITEXT14	ITEX15
ITEXT16	ITEXT17	ITEXT18	ITEXT19	ITEXT20	ITEXT21	ITEXT22	ITEX23
ITEXT24	ITEXT25	ITEXT26	ITEXT27	ITEXT28	ITEXT29	ITEXT30	ITEX31

- **Random number register: RAND0~7**

The random number register holds the random seed used by AES8, AES9, and AES10, the cryptographic cores equipped with side-channel attack countermeasures. Once a seed is set to the random number register, the register will repeat updating its content with new random numbers. AES9 uses only a 32-bit random number set in RAND0~1.

127 (MSB) (LSB) 0

RAND0	RAND1	RAND2	RAND3	RAND4	RAND5	RAND6	RAND7

- **Output text register for common-key cryptography: OTEXT0~31**

The output text register holds the output text exported by the IP that the output selection register OUTSEL designates. Note that each cryptographic core has different output data size and different output text register positions used as follows:

AES5(AES_CTR_PIPE) : Exports 128 bits × 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	Unused

128-bit block ciphers

OTEXT0~7	128-bit output
OTEXT8~31	Unused

127 (MSB) (LSB) 0

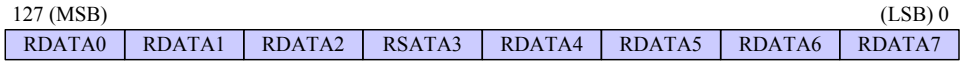
OTEXT0	OTEXT1	OTEXT2	OTEXT3	OTEXT4	OTEXT5	OTEXT6	OTEX7
OTEXT8	OTEXT9	OTEXT10	OTEXT11	OTEXT12	OTEXT13	OTEXT14	OTEX15
OTEXT16	OTEXT17	OTEXT18	OTEXT19	OTEXT20	OTEXT21	OTEXT22	OTEX23
OTEXT24	OTEXT25	OTEXT26	OTEXT27	OTEXT28	OTEXT29	OTEXT30	OTEX31

- **Intermediate data register for common-key cryptography: RDATA0~7**

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

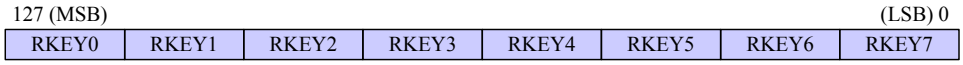
1. Designating the round at which the register holds the intermediate data.
   The round selection register RSEL[DRSEL] specifies the round. The data aligns in RDATA0 toward RDATA7 starting with the upper 16 bits.
2. When a fault error occurs.
   A fault error asserts the Err[0] signal in the AES_FA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



- **Intermediate key register for common-key cryptography: RKEY0~7**

A set of these register fractions reads an intermediate key value at the designated round for AES6. The register is valid when the selection registers IPSEL/OUTSEL designate AES6 and the KEN bit of the round selection 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 selection 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[0] signal in the AES_FA module of AES6. It also raises the KERR bit of the round selection register RSEL and captures the intermediate key.



- **Exponent register for public-key cryptography: EXP0~31**

The register takes the 512-bit exponent for RSA. EXP0 is associated with the upper 16 bits, EXP1 has the next 16 bits, and the rest follows accordingly. ECC does not use this register.

- **Modulus register for public-key cryptography: MOD0~31**

The register takes the 512-bit modulus for RSA. MOD0 is associated with the upper 16 bits, MOD1 has the next 16 bits, and the rest part follows accordingly. ECC does not use this register.

- **Preprocessed data register for public-key cryptography: PREDAT0~15**

The register takes the 256-bit preprocessed data for RSA in CTR mode. PREDAT0 is associated with the upper 16 bits, PREDAT1 has the next 16 bits, and the rest part follows accordingly. ECC does not use this register.

- **Input data register for public-key cryptography: IDATA0~31**

The register takes the 512-bit input data for RSA. IDATA0 is associated with the upper 16 bits, IDATA1 has the next 16 bits, and the rest part follows accordingly.

The register takes the following data for ECC:

IDATA0~3 :	64-bit private key data
IDATA4~7 :	64-bit random number data for side-channel countermeasure This is for future use. The current ECC core does not use it.
IDATA8~11 :	64-bit x component of the Affine coordinate for the input point.
IDATA12~15 :	64-bit z component of the projective coordinate for the input point
IDATA16~19 :	64-bit elliptic curve parameter b

IDATA20~31 : Unused.

For each group of register fractions, the smaller number suffix associates with the upper bits.

- **Output data register for public-key cryptography: ODATA0~31**

For RSA, the register exports the 512-bit processing result. ODATA0 is associated with the upper 16 bits, ODATA1 has the next 16 bits, and the rest follows in sequence.

For ECC, ODATA0~3 export the 64-bit processing result. ODATA0 has the most significant bits, and ODATA1, ODATA2, and ODATA3 follow in order. ODATA4~31 are not in use.

- **Version register: VER**

This register reads the cryptographic LSI's version. It is read only. The 130nm version has the fixed number 0x0450A, while the 90nm version has the fixed number 0x34F9.

3.4 Clock Tree

Figure 3.5 shows the clock system of the cryptographic LSI. The LSI provides only the measurement target cores with the clock signal in accordance with the interface register settings. To ease a 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.

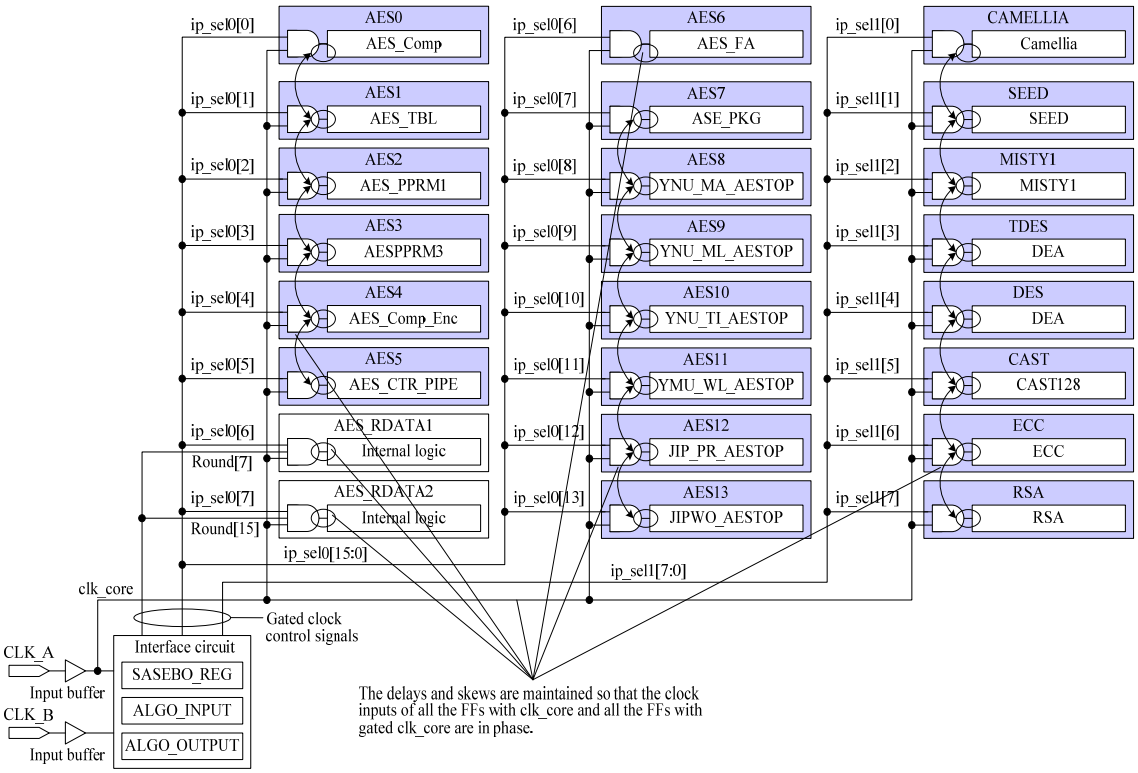


Figure 3.5 Clock System

3.5 Reset

Figure 3.6 provides an overview of the reset system of the LSI. The reset sequence is explained below. 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.

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 register 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 No.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, the output selection register subsequently must be properly set.

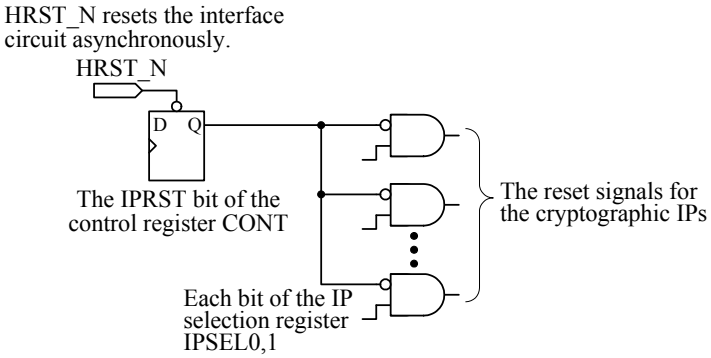


Figure 3.6 Reset System

3.6 Supplementary Functions

- **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, the internal synchronization circuit was designed assuming that the phases of CLK_A and CLK_B are 180-degree out of phase with each other.

- **Key length limit**

To meet the export control regulations, the key length for the common-key algorithms is limited to 56 bits, while lengths for RSA and ECC are limited to 512 bits and 64 bits, respectively.

For DES, set a 56-bit key excluding parity bits to the lower 8 bits of KEY4 and KEY5~7. Figure 3.7 shows the key data bit assignment for the DES core.

For 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)

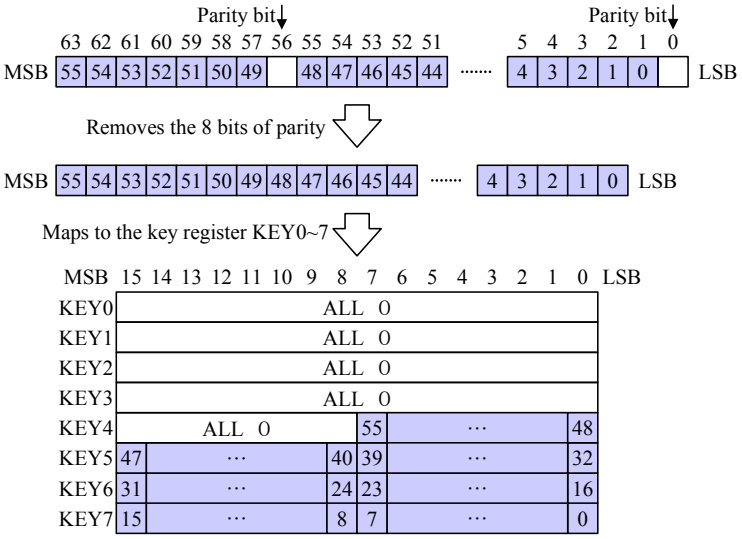


Figure 3.7 Key Data Bit Assignment for the DES Core

For common-key cryptographic algorithms, the 128-bit key is divided into the fixed upper 72 bits and the user-definable lower 56 bits as shown in Figure 3.8.

[127:56] 0x000102030405060708 (Fixed value)

[55:0] A 56-bit value consisting of the lower 8 bits of KEY4 and KEY5~7.

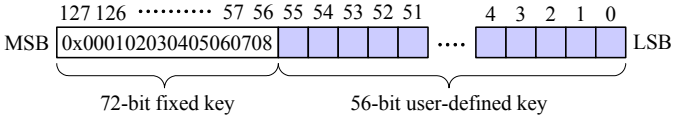


Figure 3.8 Key Data Bit Assignment for the Common-key Algorithm Cores

- **Delayed operation**

To obtain power traces precisely during cryptographic operations in the LSI, the timings of key setting, data input/output, and cryptographic operation differ from one another. Turn the RUN bit of the control register CONT on to initiate the operation, and after 8 clock cycles, the interface circuit asserts the operation initiation request signal START_N. After another 8 cycles, the cryptographic algorithm core starts, asserting the EXEC signal to indicate it is in operation. At the termination of the algorithm core operation, the circuit deasserts EXEC. 8 cycles later, the circuit asserts the END_N signal to indicate that the operation has terminated. After another 8 cycles, it turns the control register CONT[RUN] to '0'. Figure 3.9 depicts the detailed operation timings. Note that the clock cycle counts are CLK_A equivalent.

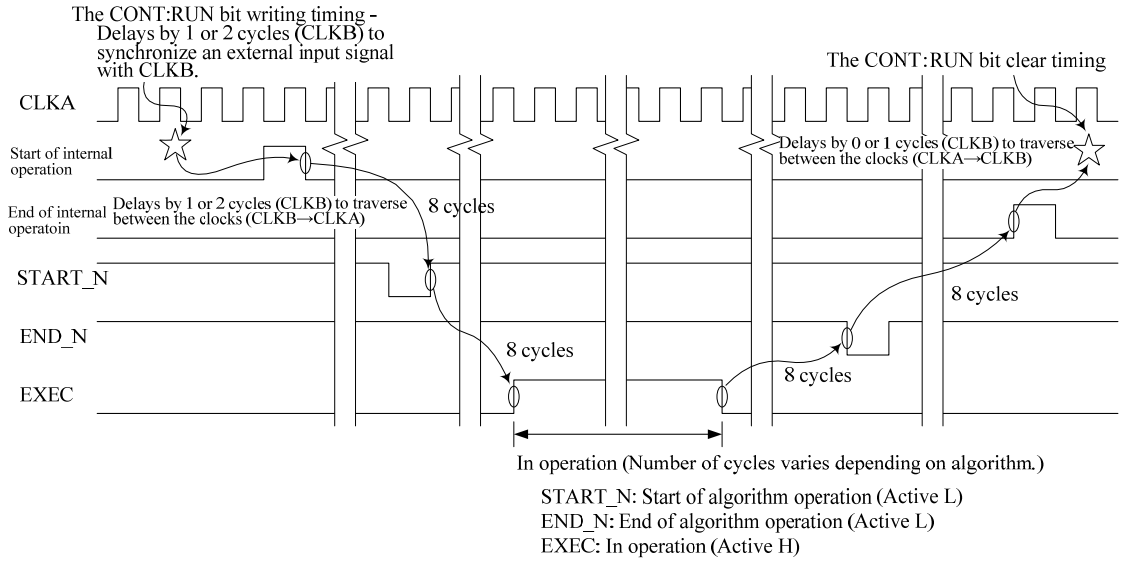


Figure 3.9 Timing Chart for Delayed Operation

- **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. To take advantage of this feature, select two or more cryptographic IPs on the IP selection register IPSEL, while simultaneously selecting the target IP on the output selection register OUTSEL.

- **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. Set the mode register MODE[OUTINH] to "10", and all the evaluation signals stay '0'.
2. Set the mode register MODE[OUTINH] to "11", and all the evaluation signals but START_N stay '0'.

- **Free-run mode**

The AES0 core has the free-run mode in which encryption or decryption repeats every 0.3 seconds. Turning the control register CONT[RUN] to '1' activates the free-run mode. 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. Set the mode register MODE[FRUN] to "10", and 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.
2. Set the mode register MODE[FRUN] to "11", and 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.

- **Handling the input text register ITEXT**

For the common-key cryptographic cores, note that each IP has a different size of input data and a different mapping to the input text register as follows:

1. AES5 (CTR mode supported, 4-stage pipelined)
   - ITEXT0~7 128-bit input (First block)
   - ITEXT8~15 128-bit input (Second block)
   - ITEXT16~23 128-bit input (Third block)
   - ITEXT24~31 128-bit input (Fourth block)
2. Other 128-bit block cryptographic cores
   - ITEXT0~7 128-bit input
   - ITEXT8~31 Unused
3. 64-bit block cryptographic cores
   - ITEXT0~3 64-bit input
   - ITEXT4~31 Unused

- **Handling the output text register OTEXT**

For the common-key cryptographic cores, note that each IP has a different size of output data and a different mapping of output text register as follows:

1. AES5 (CTR mode supported, 4-stage pipelined)
   - OTEXT0~7 128-bit input (First block)
   - OTEXT8~15 128-bit input (Second block)
   - OTEXT16~23 128-bit input (Third block)
   - OTEXT24~31 128-bit input (Fourth block)
2. Other 128-bit block cryptographic cores
   - OTEXT0~7 128-bit input
   - OTEXT8~31 Don't care
3. 64-bit block cryptographic cores
   - OTEXT0~3 64-bit input
   - OTEXT4~7 0x0000000000000000
   - OTEXT7~31 Don't care

- **Handling the random number register RAND for DPA countermeasure**

While the AES8(Masked AND operation) and AES10(Threshold implementation) cores take the random number seed from the random number register RAND0~7, the AES9(MDPL) core takes its 32-bit seed from the random number register RAND0~1.

- **CTR operation of AES5 (CTR mode supported, pipelined)**

Unlike the other cores, the AES5 core performs data input and output of 4 128-bit blocks continuously. Set the initial value of the counter to the initial value register (IV), and the core generates the random numbers for the 4 blocks for the CTR mode. The output text register OTEXT will not export the output text until the input text register ITEXT receives a 4-block plaintext or ciphertext. Immediately after the output text register OTEXT exports the resulting 4-block ciphertext or plaintext, the next 4-block random number is generated.

- **Exporting intermediate data**

The AES6 (Fault injection attack countermeasure implemented) core is able to export the intermediate data value. To enable this function, select AES6 on both the IP selection register IPSEL and the output selection register OUTSEL, and set the round selection register RSEL[DEN] to '1', and then the intermediate data register RDATA0~7 exports the intermediate data value. Use the round selection register RSEL[DRSEL] to specify the exporting round.

- **Exporting intermediate key**

The AES6 core is able to export the intermediate key value. To enable this function, select AES6 on both the IP selection register IPSEL and the output selection register OUTSEL, and set the round selection register RSEL[KEN] to '1', and then the intermediate key register RKEY0~7 exports the intermediate key value. Use the round selection register RSEL[KRSEL] to specify the exporting round.

- **Exporting for fault injection attack (FA) experiments**

In the event of a fault error while the AES6 core is in operation, either of the round selection register RSEL[KERR] or RSEL[DERR] turns to '1'. At the time of the fault, the intermediate register RDATA0~7 and intermediate key register RKEY0~7 export the intermediate data value and key value, respectively.

4. PHYSICAL LAYOUT

4.1 130-nm Version

This section presents the layout information resulting from the logic synthesis of the cryptographic LSI with a 130-nm standard cell library. Table 4.1 shows the outline of the 130-nm LSI. The logic uses 24.95 % of the gates on its 5×5 mm² die. Table 4.2, Table 4.3, and Table 4.4 list the software tools, libraries, and constrains used for logic synthesis, respectively. While the target frequency is 24 MHz (41-ns cycle), to ease timing adjustment during the layout phase, the design was synthesized with a 31-MHz constraint which has a margin of 30 %. Significant delays were given in the inputs and outputs to ensure extra setup margins.

Table 4.1 Outline of the 130nm Cryptographic LSI

Item	Description
Technology	130-nm Logic General Purpose 1P8M 1.2V- 3.3V CU FSG
Wafer process	TSMC CLN130G 130-nm CMOS, Al 7-layer metal
Core power voltage	1.2 ± 0.12 V
I/O power voltage	3.3 ± 0.16 V
Operating frequency	24 MHz (41 ns)
Data area	5 × 5 mm^2
Cell usage (count)	4,129,178/16,550,023 (Cell area used/available)
Cell usage (%)	24.9 5%
Number of PAD	160
Custom cell	SRAM

Table 4.2 EDA Tools Used to Design the 130nm Cryptographic LSI

Purpose	Software	Vendor	Version
Logic synthesis	Design Compiler	Synopsys	Z-2007.03-SP5
Place and route	SOC Encounter	Cadence	v06.20-s285_1
RC extraction	Star-RCXT	Synopsys	Z-2006.12-SP1
Cross-talk extraction	CeltIC	Cadence	v06.20-s075_1
STA	PrimeTimeSI	Synopsys	Z-2007.06.-SP3
Layout verification	Calibre	Mentor	v2008.3-25.16
Power verification	AstroRail	Synopsys	Z-2007.03-SP8
Equivalence checking	Formality	Synopsys	Z-2007.06.SP-3

Table 4.3 Libraries Used to Design the 130nm Cryptographic LSI

Type	Library	Version
Standard cell	SAGE-X Standard Cells (TSMC CL013G)FB	2007q1v2
	SAGE-X Standard Cells (TSMC CL013G) FX-CeltIC	2005q3v1
Digital I/O	EZBond, I/O, TPZ013G3, 1.2V/3.3V	210c
RAM	2P-RF ADV(TSMC CL013G) FB	2004q2v1
	SP-RF ADV(TSMC CL013G) FB	2003q4v1

Table 4.4 Logic Synthesis Constraints for the 130-nm Cryptographic LSI

Constraint name	Constraint condition
Operating frequency	31 MHz (32 ns) 24 MHz + 30 %margin
Input delay	2 ns
Output delay	2 ns
External load capacitance	20 pf
Virtual wire delay	tsmc130_w110

Figure 4.1 shows the top view of the 130-nm cryptographic LSI excluding the power lines. Figure 4.2 depicts the register array allocation. Figure 4.3 shows the cryptographic module allocation of the LSI. The module sizes are shown in Table 4.5. Table 4.6 lists the LSI's speeds for each operating environment of Worst (125 °C, 1.08 V), Typical (25 °C, 1.20 V), and Best (-40 °C, 1.32 V). The timing analysis was performed using the Typical manufacturing process parameters. Table 4.7 shows each module's performance. Among them, the AES circuit with the side-channel attack countermeasure MDPL(Masked Dual-rail Precharge Logic) is the slowest and largest, which runs at 25.78MHz for the Worst case (41.24 MHz for Typical, 62.68 MHz for Best) and results in a circuit size of 124,319 gates. Although this module barely meets the target frequency of 24MHz, the entire LSI falls to 20.410 MHz for the Worst case (24.889 MHz for Typical, 29.410 MHz for Best) and does not meet the frequency constraint as shown in Table 4.6. Nevertheless, there is little impact of this degradation factor on an experiment, because the SASEBO-R, designed to mount this LSI, is able to change its default clock frequency of 24 MHz; and its core voltage is variable as well.

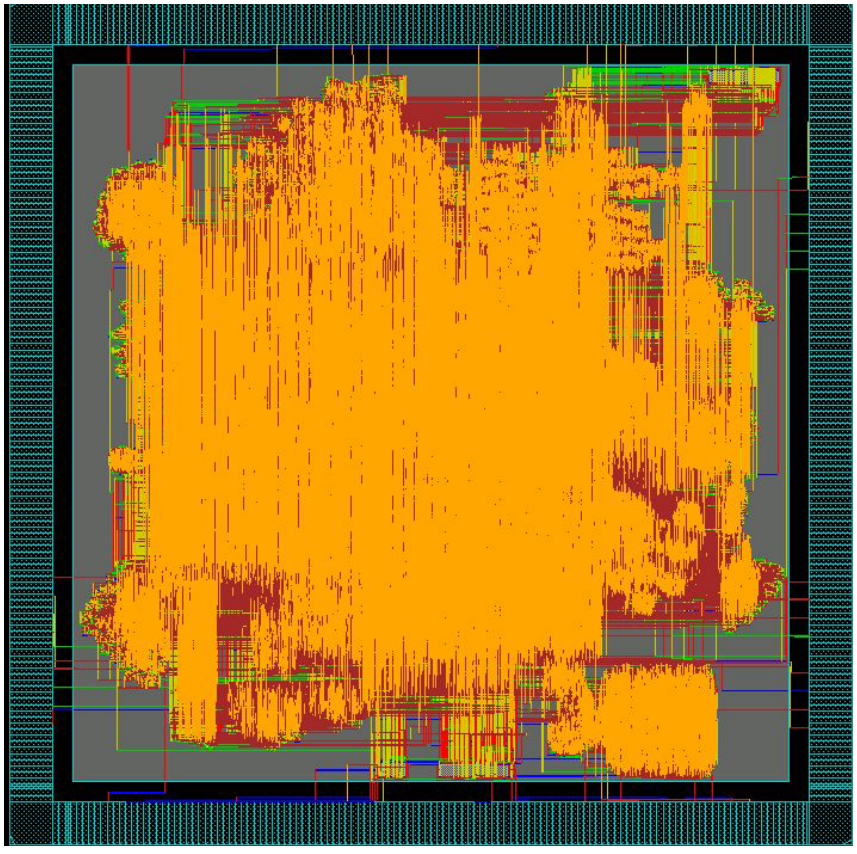


Figure 4.1 Top View of the 130-nm Cryptographic LSI

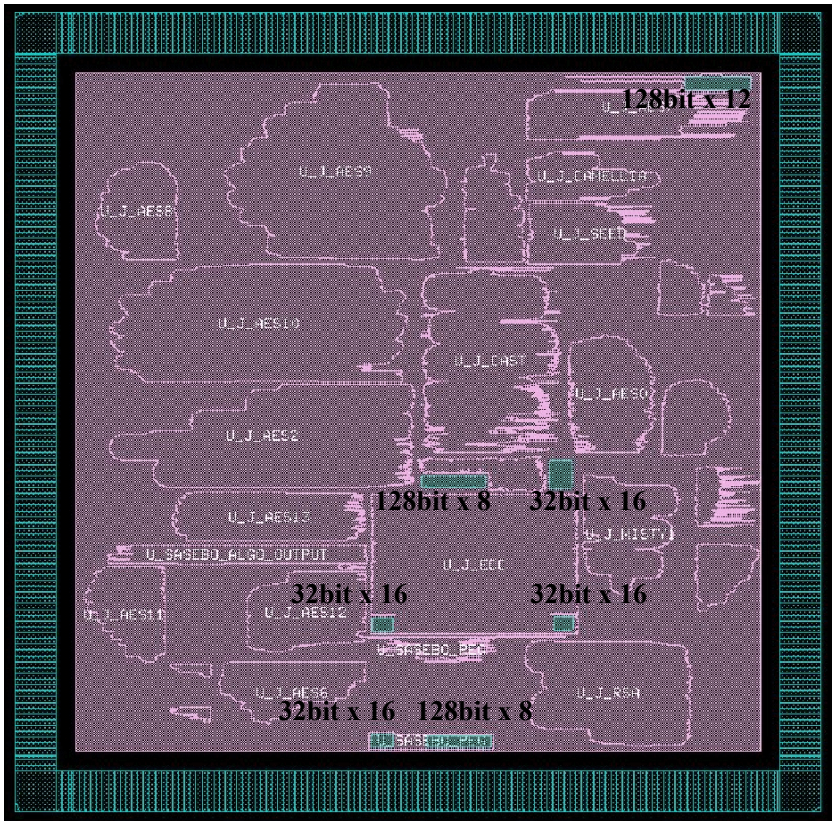


Figure 4.2 Register Array Allocation of the 130-nm Cryptographic LSI

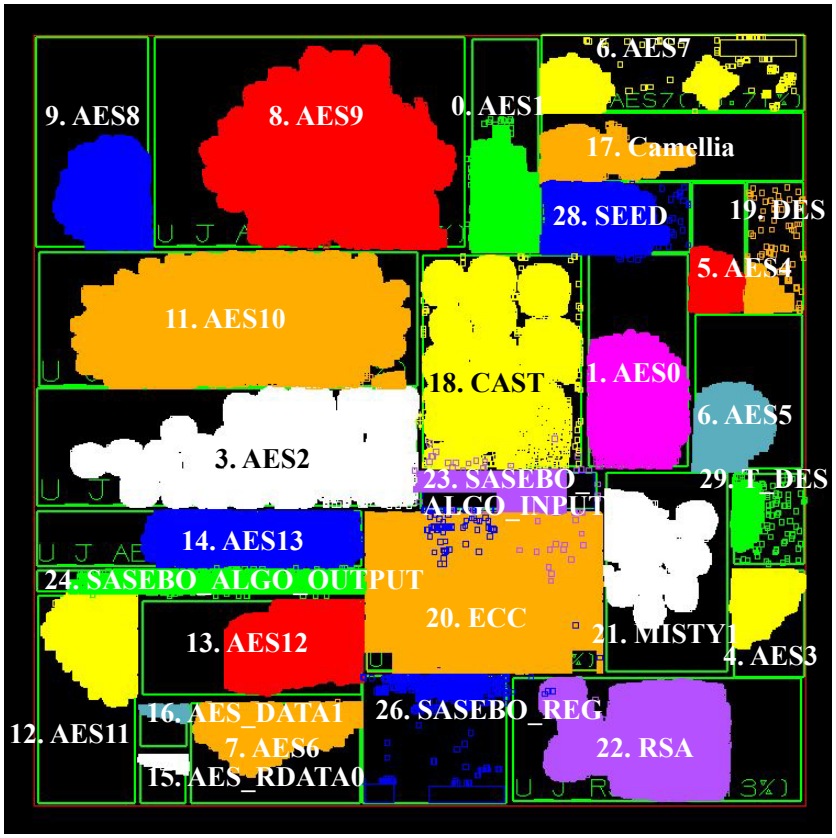


Figure 4.3 Cryptographic Module Allocation of the 130-nm Cryptographic LSI

Table 4.5 Module Areas of the 130-nm Cryptographic LSI

Module name	Area (μm^2)	Area (%)	Gate count
1. AES0 (Composite field S-box)	129,763	2.7	25,483
2. AES1 (Table S-box)	105,097	2.2	20,639
3. AES2 (1-stage PPRM S-box)	314,702	6.5	61,801
4. AES3 (3-stage PPRM S-box)	84,230	1.7	16,541
5. AES4 (Composite field S-box)	61,408	1.3	12,059
6. AES5 (CTR mode)	112,794	2.3	22,150
7. AES6 (FA countermeasure)	105,125	2.2	20,644
8. AES7 (Round key preprocess)	93,655	1.9	18,392
9. AES8 (MAO)	179,253	3.7	35,202
10. AES9 (MDPL)	633,056	13.1	124,319
11. AES10 (Threshold)	555,510	11.5	109,090
12. AES11 (WDDL)	152,225	3.1	29,894
13. AES12 (Pseudo RSL)	169,789	3.5	33,343
14. AES13 (Pseudo RSL)	99,295	2.1	19,499
15. AES_RDATA2	7,373	0.2	1,448
16. AES_RDATA1	7,387	0.2	1,451
17. Camellia	73,407	1.5	14,416
18. CAST	148,921	3.1	29,245
19. DES	16,176	0.3	3,177
20. ECC	339,046	7.0	66,581
21. MISTY1	85,733	1.8	16,836
22. RSA	359,011	7.4	70,502
23. SASEBO_ALGO_INPUT	63,135	1.3	12,398
24. SASEBO_ALGO_OUTPUT	33,143	0.7	6,509
25. SASEBO_INPUT	2,400	0.0	471
26. SASEBO_REG	127,903	2.6	25,117
27. SASEBO_VALUE	596	0.0	117
28. SEED	115,237	2.4	22,630
29. T_DES	27,127	0.6	5,327
Total cell area	4,830,232	100.0	948,555

1 gate = 2-input NAND (3.69μm×1.38μm)

Table 4.6 The Operation Speeds of the 130-nm LSI by Static Timing Analysis

Property	Worst (125°C 1.08V)	Typical (25°C 1.2V)	Best (−40°C 1.32V)
Maximum frequency	20.410 MHz	24.889 MHz	29.410 MHz
Critical path	48.995 ns	40.178 ns	33.993 ns
Hold time	0.482 ns	0.340 ns	0.208 ns

Table 4.7 The Operation Speeds of the Cryptographic Modules of the 130-nm LSI (for Typical Process)

Module name	Worst (125°C 1.08V)			Typical (25°C 1.2V)			Best (−40°C 1.32V)		
	Freq. (MHz)	Critic. (ns)	Hold (ns)	Freq (MHz)	Critic. (ns)	Hold (ns)	Freq (MHz)	Critic. (ns)	Hold (ns)
1. AES0 (Composite field S-box)	66.30	15.085	0.537	107.54	9.299	0.328	163.51	6.116	0.104
2. AES1 (Table S-box)	108.47	9.219	0.528	125.56	5.696	0.322	266.45	3.753	0.210
3. AES2 (1-stage PPRM S-box)	41.86	23.88	0.503	68.18	14.666	0.295	105.82	9.450	0.193
4. AES3 (3-stage PPRM S-box)	93.21	10.729	0.514	152.28	6.567	0.302	236.41	4.230	0.197
5. AES4 (Composite field S-box)	93.79	10.662	0.547	153.44	6.517	0.324	239.64	4.173	0.209
6. AES5 (CTR mode)	29.03	34.452	0.495	47.74	20.948	0.282	73.25	13.652	0.168
7. AES6 (FA countermeasure)	46.41	21.546	0.516	74.35	13.449	0.324	111.91	8.936	0.209
8. AES7 (Round key preprocess)	92.99	10.754	0.423	152.93	6.539	0.252	238.83	4.187	0.145
9. AES8 (MAO)	61.36	16.298	0.498	102.08	9.796	0.297	161.21	6.203	0.193
10. AES9 (MDPL)	25.78	38.786	0.431	41.24	24.249	0.241	62.68	15.955	0.147
11. AES10 (Threshold)	52.78	18.953	0.481	87.61	11.414	0.297	137.51	7.272	0.187
12. AES11 (WDDL)	54.00	18.517	0.393	88.33	22.323	0.216	135.01	7.407	0.128
13. AES12 (Pseudo RSL)	30.36	32.940	0.319	35.51	28.164	0.197	39.14	15.499	0.098
14. AES13 (Pseudo RSL)	61.33	16.304	0.511	100.41	9.959	0.317	156.23	6.401	0.207
15. AES_RDATA2	311.82	3.207	0.714	498.26	2.007	0.444	733.14	1.364	0.292
16. AES_RDATA1	283.13	3.432	0.701	451.88	2.213	0.435	663.13	1.508	0.289
17. Camellia	69.78	14.330	0.572	112.20	8.913	0.355	169.81	5.889	0.225
18. CAST	33.36	29.889	0.483	54.14	18.471	0.305	81.91	12.209	0.203
19. DES	143.29	6.979	0.510	228.99	4.367	0.320	342.35	2.921	0.211
20. ECC	63.16	15.933	0.435	101.62	9.841	0.268	153.63	6.509	0.163
21. MISTY1	29.46	33.943	0.485	47.68	20.971	0.301	72.58	13.778	0.205
22. RSA	30.60	32.683	0.475	51.32	19.486	0.286	80.22	12.466	0.165
23. SASEBO_ALGO_INPUT	30.51	32.778	0.362	49.83	20.068	0.233	75.67	13.125	0.150
24. SASEBO_ALGO_OUTPUT	340.60	2.936	0.513	541.42	1.847	0.315	817.66	1.223	0.202
25. SASEBO_INPUT	620.35	1.612	0.469	1007.0	0.993	0.273	1477.1	0.677	0.173
26. SASEBO_REG	53.31	18.759	0.197	84.01	11.904	0.197	123.03	8.128	0.121
27. SASEBO_VALUE	–	–	–	–	–	–	–	–	–
28. SEED	29.09	34.38	0.508	47.54	21.033	0.313	73.13	13.674	0.204
29. T_DES	103.80	9.634	0.508	165.92	6.027	0.319	248.20	4.029	0.205

The LSI adopts 7-layer metal wiring. Figure 4.4 represents the signal wirings for each layer excluding power lines. Figure 4.5 through Figure 4.8 illustrate the power supply to all the cells over the chip. The VDD/VSS power rings form along the circumference of the chip with the 6th and 7th layers. The power disperses over the chip from the rings on the meshes on the 4th through 7th layers. Stripes on the 4th layer relay the power to each cell through the Stack Vias. The I/O buffers connect with the VDD/VSS power through the 2nd and 3rd layers.

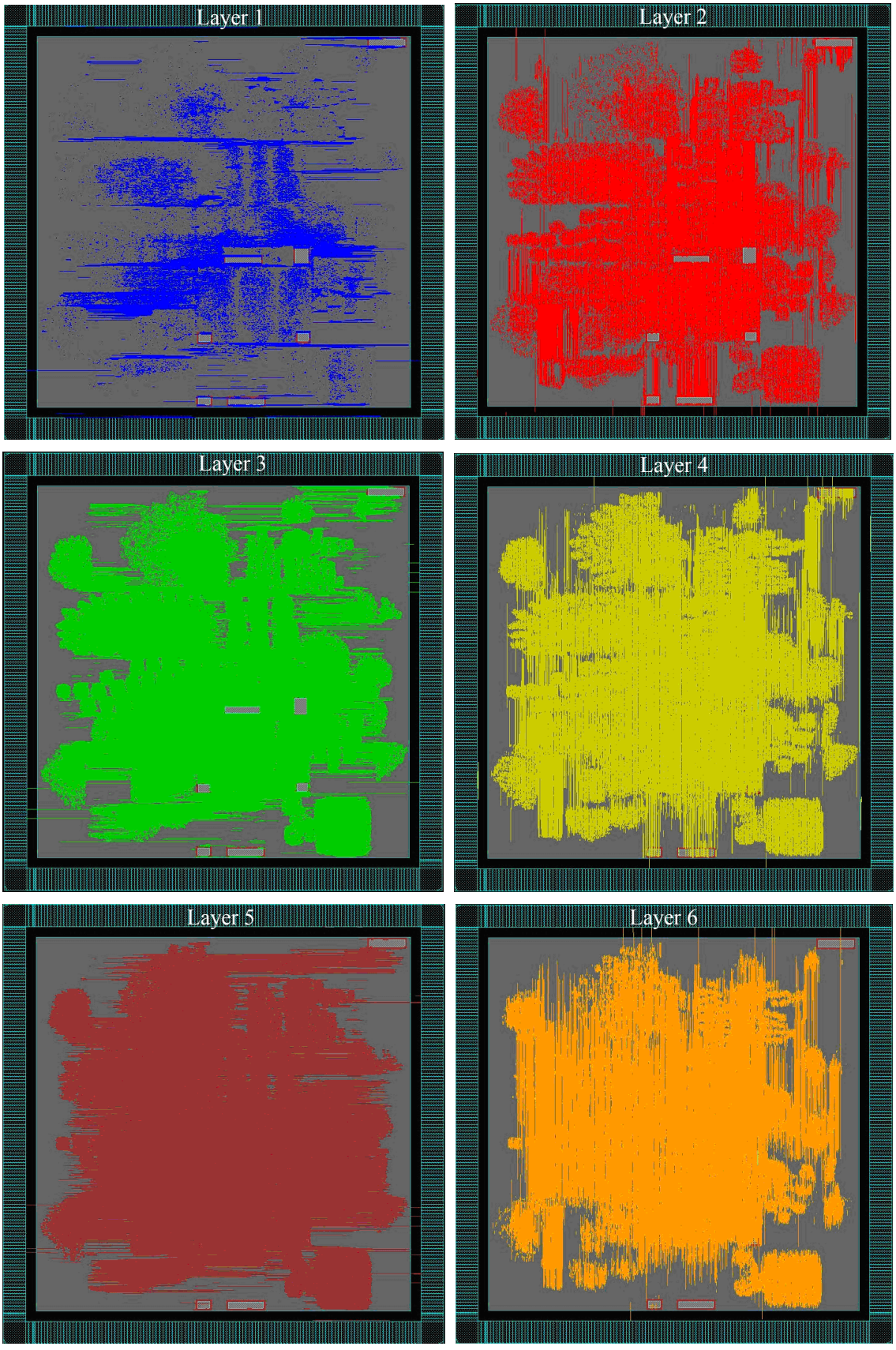


Figure 4.4　Signal Wiring Patterns of the Layers of 130-nm LSI (1/2)

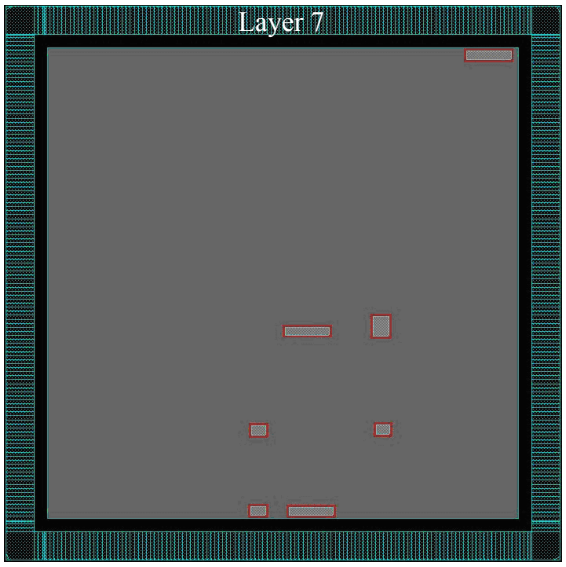


Figure 4.4 Signal Wiring Patterns of the Layers of 130-nm LSI (2/2)

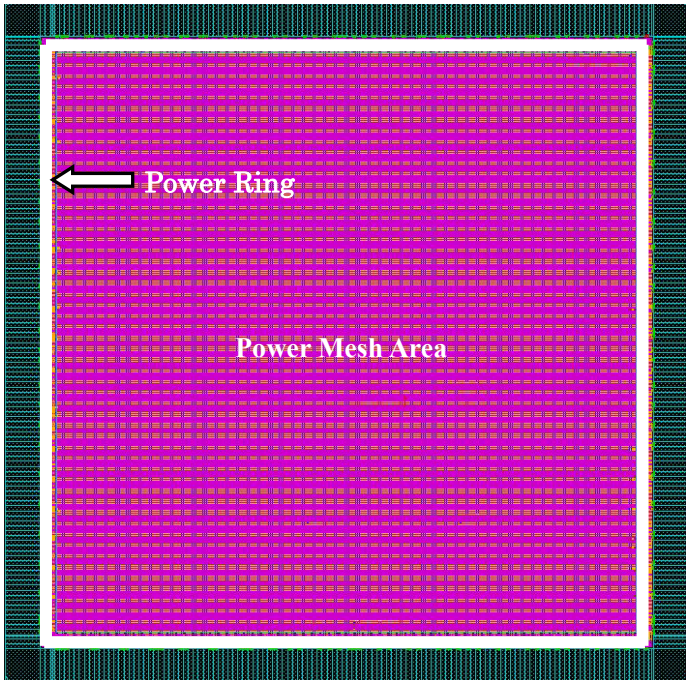


Figure 4.5 Power Wiring of the 130-nm LSI

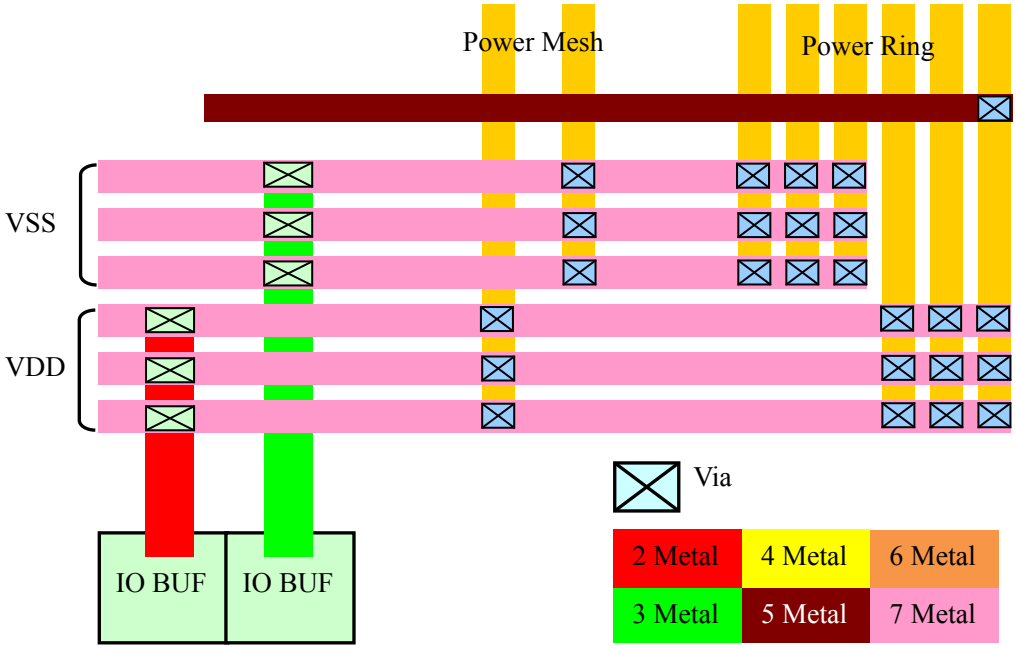


Figure 4.6 Power Rings of the 130-nm LSI

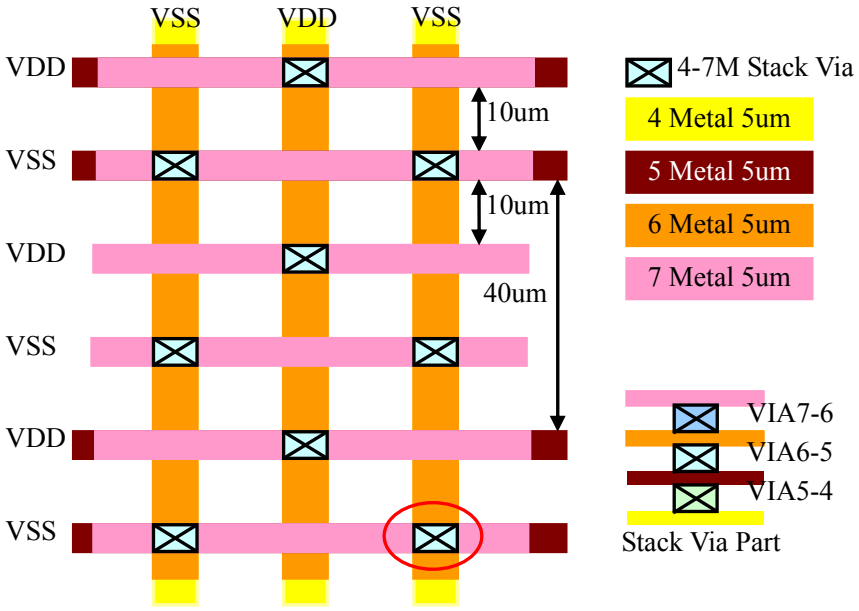


Figure 4.7 Power Mesh Structure of the 130-nm LSI

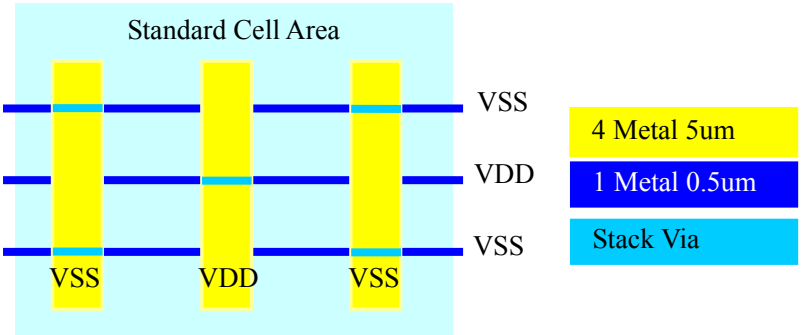


Figure 4.8 Power Supply to a Cell of the 130-nm LSI

Figure 4.9 shows the power line wiring patterns. Table 4.8 shows the power consumption and voltage drops with the assumption that the 30 % of all the cells are active. Figure 4.10 renders the voltage drops on the core power planes on the VDD and VSS sides in colors. Because only a single cryptographic core out of the 22 cores operates at a time in practice and the drop ratio of 0.4253 % is sufficiently small, a voltage drop will not cause a problem in normal operation.

Table 4.8 VDD/VSS Voltage Drops of the 130-nm LSI

	VDD	VSS
Frequency	24 MHz	
Transition probability	30 %	
Power consumption	109.429 mW	
Worst drop	5.104 mV	4.333 mV
Drop ratio	0.4253 %	0.3611 %

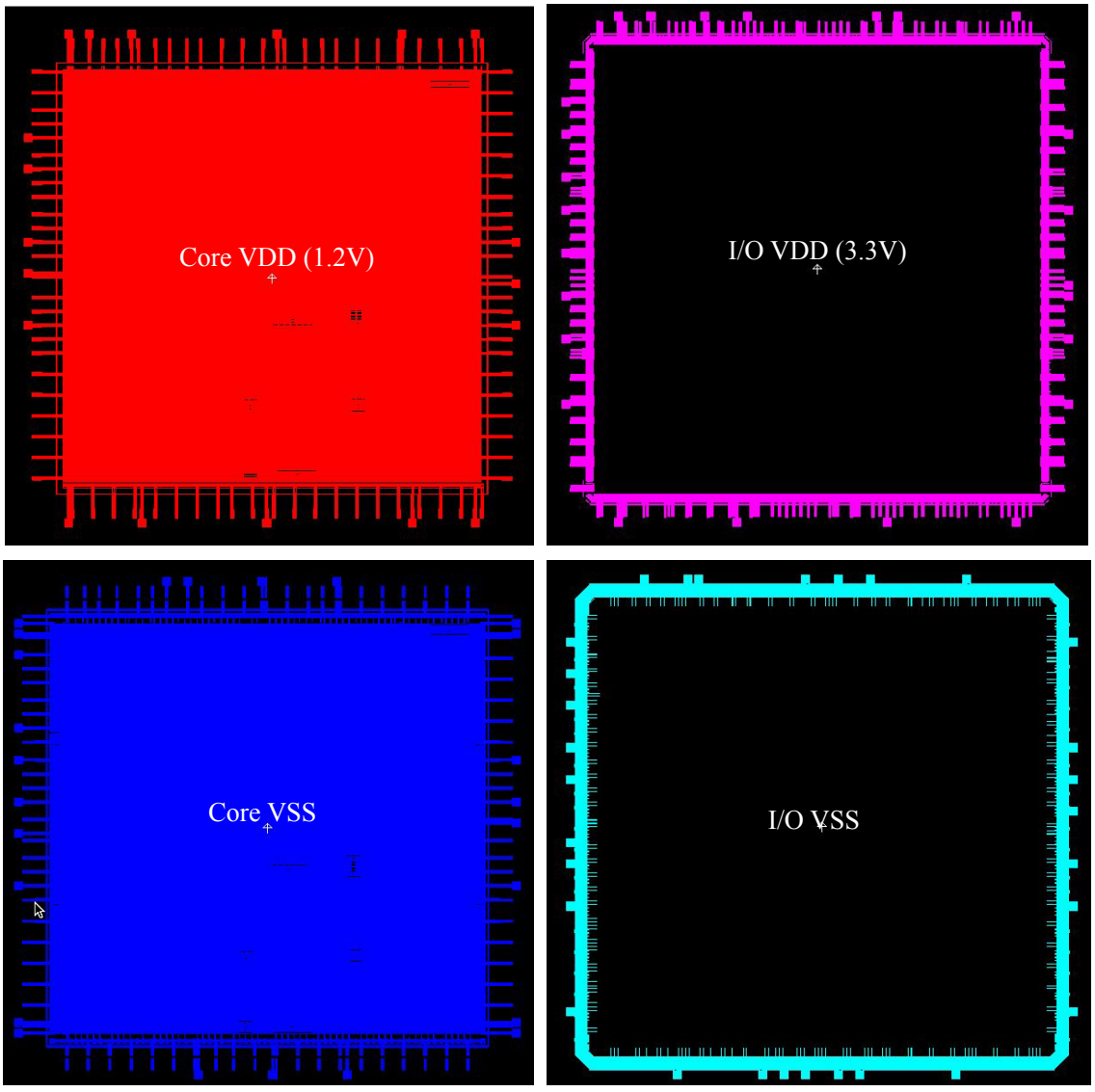
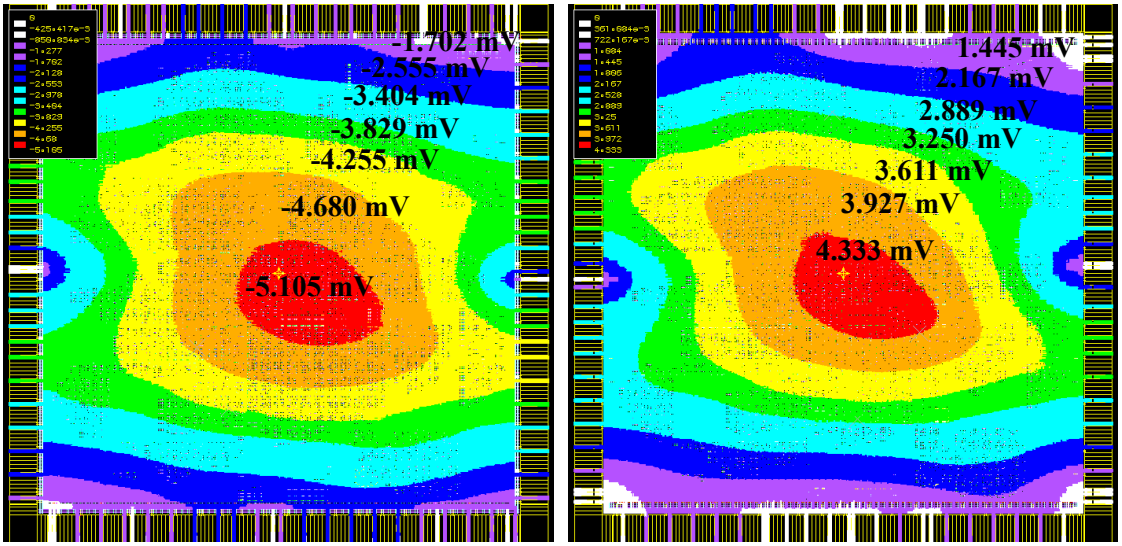


Figure 4.9 Power Line Wiring Patterns of the 130-nm LSI



(a) IR-DROP(VDD) (b) IR-DROP(VSS)

Figure 4.10 VDD/VSS Voltage Drops of the 130-nm LSI

4.2 90-nm Version

This section presents the layout information resulting from the logic synthesis of the cryptographic LSI with a 90-nm standard cell library. Table 4.9 shows the outline of the 90-nm LSI. The logic uses 19.43 % of the gates on its 4×4 mm² die. Table 4.10, Table 4.11, and Table 4.12 list the software tools, libraries, and constrains used for logic synthesis, respectively. While the target frequency is 24 MHz (41-ns cycle), to ease timing adjustment during the layout phase, the design was synthesized with a 31-MHz constraint which has a margin of 30 %. Significant delays were given in the inputs and outputs to ensure extra setup margins.

Table 4.9 Outline of the 90-nm Cryptographic LSI

Item	Description
Technology	90-nm Logic General Purpose 1P9M 1V-3.3V All Cu Low-k
Wafer process	TSMC CLN90G 90-nm CMOS, Al 7-layer metal
Core power voltage	1.0 ± 0.10 V
I/O power voltage	3.3 ± 0.16 V
Operating frequency	24 MHz (41 ns)
Data area	4 × 4 mm^2
Cell usage (count)	2,078,003/10,692,39 (Cell area used/available)
Cell usage (%)	19.43 %
Number of PAD	160
Custom cell	SRAM

Table 4.10 EDA Tools Used to Design the 90-nm Cryptographic LSI

Purpose	Software	Vendor	Version
Logic synthesis	Design Compiler	Synopsys	Z-2007.03-SP5
Place and route	SOC Encounter	Cadence	V06.20-s285_1
RC extraction	Star-RCXT	Synopsys	Z-2006.12-SP1
Cross-talk extraction	CeltIC	Cadence	V06.20-s075_1
STA	PrimeTimeSI	Synopsys	Z-2007.06.-SP3
Layout verification	Calibre	Mentor	V2008.3-25.16
Power verification	AstroRail	Synopsys	Z-2007.03-SP8
Equivalence checking	Formality	Synopsys	Z-2007.06.SP-3

Table 4.11 Libraries Used to Design the 90-nm Cryptographic LSI

Type	Library	Version
Standard cell	SAGE-X Standard Cells (TSMC CLN90G) FB	A0173
	SAGE-X Standard Cells (TSMC CLN90G) FX-CeltIC	2005q3v2
Digital I/O	In-line, I/O, TPDN90G3, 1.0V/3.3V, (TSMC CLN90G)	130a
RAM	2P-RF ADV (TSMC CLN90G) FB	2008Q3V1
	SP-RF ADV (TSMC CLN90G) FB	2007Q2V1

Table 4.12 Logic Synthesis Constraints for the 90-nm Cryptographic LSI

Constraint name	Constraint condition
Operating frequency	31 MHz (32 ns) 24 MHz + 30 %margin
Input delay	2 ns
Output delay	2 ns
External load capacitance	20 pf
Virtual wire delay	tsmc90_w110

Figure 4.11 shows the top view of the 90-nm cryptographic LSI excluding the power lines. Figure 4.12 depicts the register array allocation. Figure 4.13 shows the cryptographic module allocation of the LSI. The module sizes are shown in Table 4.13. Table 4.14 lists the LSI's speeds for each operating environment of Worst (125 °C, 0.9 V), Typical (25 °C, 1.0 V), and Best (-40 °C, 1.1 V). The timing analysis was performed using a few different process parameter combinations of parasitic capacitances and resistances. Table 4.14 shows each module's performance. Among them, the AES circuit with the side-channel attack countermeasure Pseudo RSL2 (Random Switching Logic) is the slowest, which runs at 20.75MHz for the Worst case (35.56 MHz for Typical, 38.99 MHz for Best). As shown in Table 4.16, the entire LSI runs at 30.468 MHz for the Worst case (35.319 MHz for Typical, 38.799 MHz for Best), which meets the target clock frequency of 24 MHz.

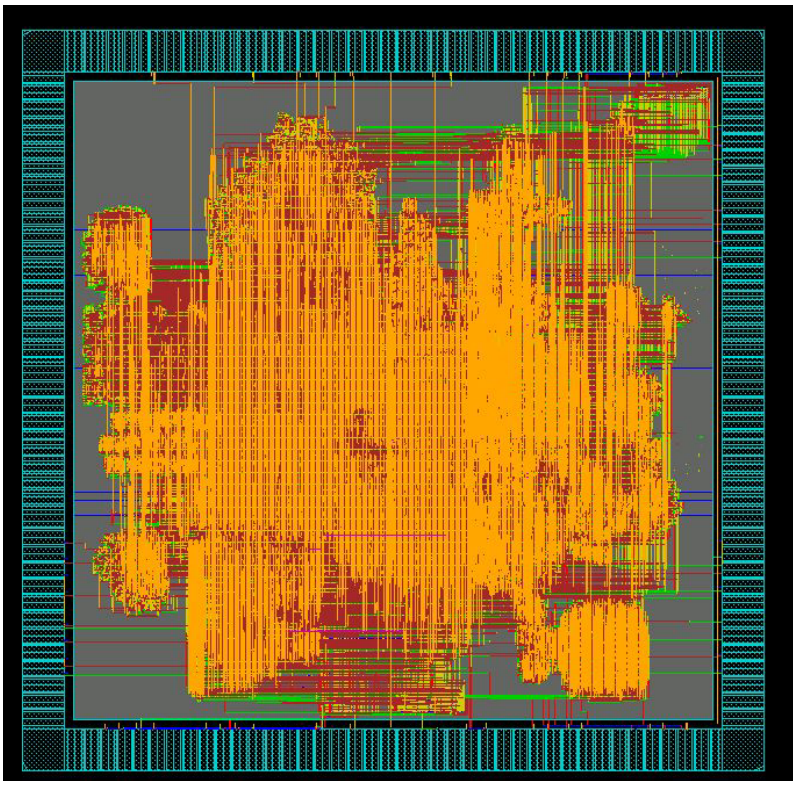


Figure 4.11 Top View of the 90-nm Cryptographic LSI

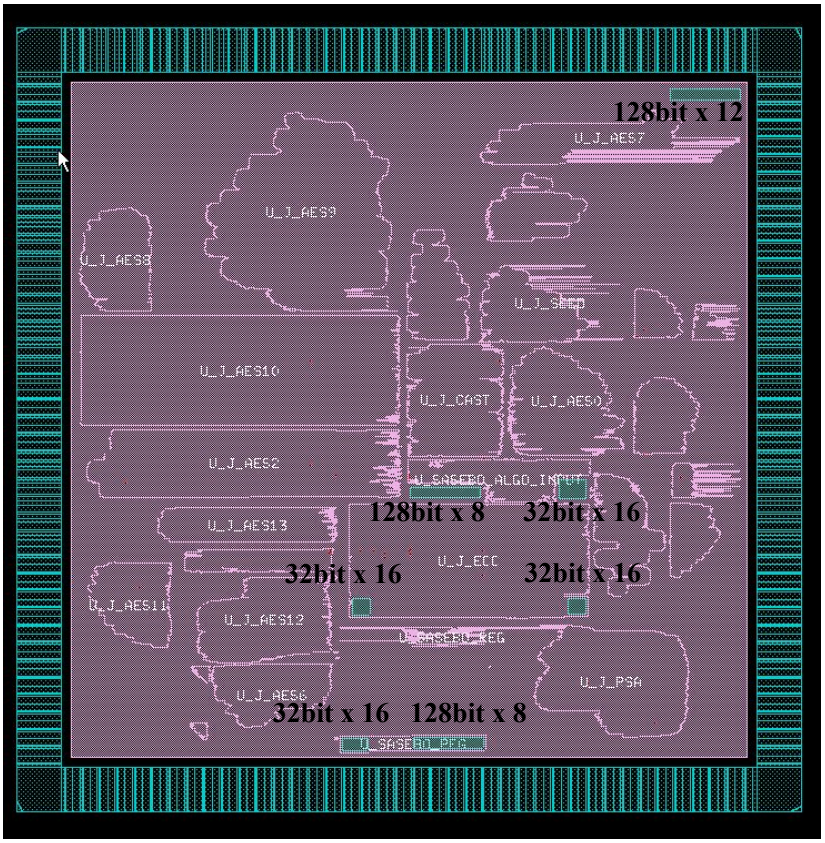


Figure 4.12 Register Array Allocation of the 90-nm Cryptographic LSI

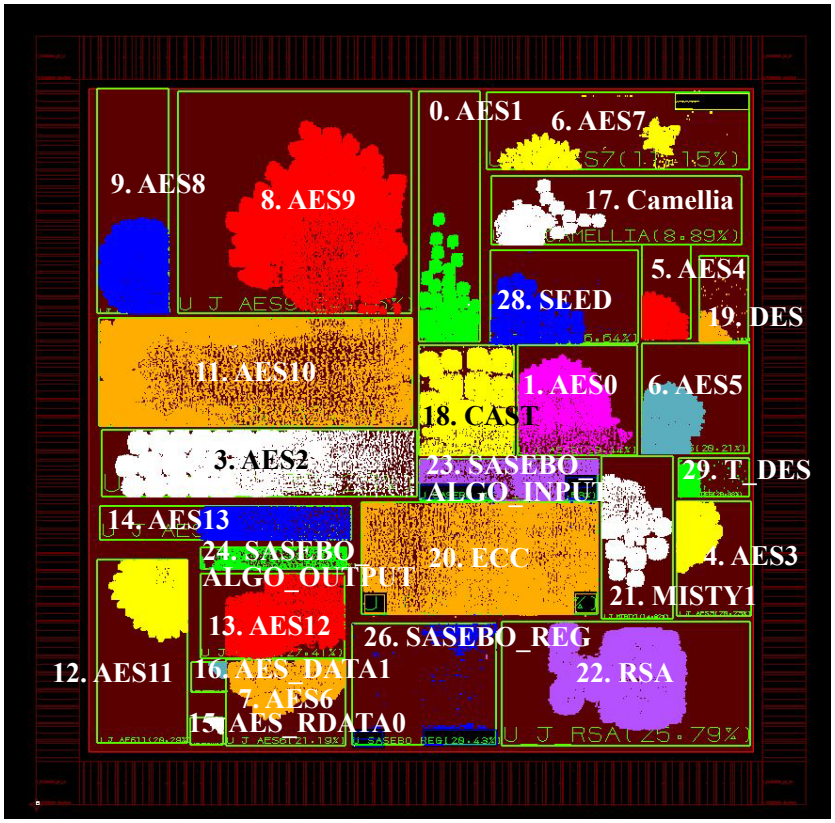


Figure 4.13 Cryptographic Module Allocation of the 90-nm Cryptographic LSI

Table 4.13 Module Areas of the 90-nm Cryptographic LSI

Module name	Area (μm^2)	Area (%)	Gate count
1. AES0 (Composite field S-box)	64,111	2.5	22,715
2. AES1 (Table S-box)	52,958	2.1	18,763
3. AES2 (1-stage PPRM S-box)	152,991	5.9	54,206
4. AES3 (3-stage PPRM S-box)	41,358	1.6	14,653
5. AES4 (Composite field S-box)	29,921	2.0	10,601
6. AES5 (CTR mode)	58,430	2.2	20,702
7. AES6 (FA countermeasure)	50,721	2.0	17,971
8. AES7 (Round key preprocess)	54,855	2.1	19,436
9. AES8 (MAO)	87,345	3.3	30,947
10. AES9 (MDPL)	306,231	11.6	108,500
11. AES10 (Threshold)	269,448	10.5	95,468
12. AES11 (WDDL)	83,206	3.2	29,480
13. AES12 (Pseudo RSL)	80,511	3.1	28,526
14. AES13 (Pseudo RSL)	46,131	1.8	16,344
15. AES_RDATA2	3,014	0.1	1,068
16. AES_RDATA1	3,018	0.1	1,069
17. Camellia	35,633	1.4	12,625
18. CAST	71,445	2.8	25,313
19. DES	8,058	0.3	2,855
20. ECC	168,459	6.6	59,686
21. MISTY1	41,051	1.6	14,545
22. RSA	191,661	7.4	67,907
23. SASEBO_ALGO_INPUT	22,195	1.1	7,864
24. SASEBO_ALGO_OUTPUT	13,572	0.6	4,809
25. SASEBO_INPUT	1,082	0.0	383
26. SASEBO_REG	86,867	3.3	30,778
27. SASEBO_VALUE	315	0.0	112
28. SEED	55,425	2.1	19,638
29. T_DES	13,377	0.5	4,740
Total cell area	2,557,262	100.00	906,059

1 gate = 2-input NAND (2.52μm×1.12μm)

Table 4.14 The Operation Speeds of the 90-nm LSI by Static Timing Analysis

Process condition	Property	Worst (125°C 0.9V)	Typical (25°C 1.0V)	Best (−40°C 1.1V)
Parasitic capacitance : Best	Max. frequency	30.902 MHz	35.690 MHz	39.095 MHz
	Critical path	32.360 ns	28.019 ns	25.579 ns
	Hold time	0.117 ns	0.066 ns	0.037 ns
Parasitic resistance capacitance : Best	Max. frequency	30.469 MHz	35.314 MHz	38.790 MHz
	Critical path	32.820 ns	28.317 ns	25.780 ns
	Hold time	0.122 ns	0.077 ns	0.042 ns
Parasitic resistance capacitance : Typical	Max. frequency	30.747 MHz	35.556 MHz	38.989 MHz
	Critical path	32.524 ns	28.125 ns	25.648 ns
	Hold time	0.122 ns	0.079 ns	0.043 ns
Parasitic capacitance : Worst	Max. frequency	30.468 MHz	35.319 MHz	38.799 MHz
	Critical path	32.821 ns	28.313 ns	25.734 ns
	Hold time	0.137 ns	0.087 ns	0.048 ns
Parasitic resistance capacitance : Worst	Max. frequency	30.917 MHz	35.696 MHz	39.093 MHz
	Critical path	32.344 ns	28.014 ns	25.580 ns
	Hold time	0.136 ns	0.078 ns	0.046 ns

Table 4.15 The Operation Speeds of the Cryptographic Modules of the 90-nm LSI (for Typical process)

Module name	Worst (125°C 1.08V)			Typical (25°C 1.2V)			Best (−40°C 1.32V)		
	Freq. (MHz)	Critic. (ns)	Hold (ns)	Freq (MHz)	Critic. (ns)	Hold (ns)	Freq (MHz)	Critic. (ns)	Hold (ns)
1. AES0 (Composite field S-box)	144.30	8.749	0.288	181.42	5.512	0.164	312.30	3.202	0.104
2. AES1 (Table S-box)	153.44	6.517	0.276	249.56	4.007	0.165	402.41	2.485	0.107
3. AES2 (1-stage PPRM S-box)	55.50	18.017	0.275	88.30	11.325	0.159	165.07	6.058	0.104
4. AES3 (3-stage PPRM S-box)	157.13	6.364	0.279	257.20	3.888	0.159	434.22	2.303	0.104
5. AES4 (Composite field S-box)	154.99	6.452	0.287	250.31	3.995	0.169	426.62	2.344	0.105
6. AES5 (CTR mode)	59.69	16.753	0.237	99.28	10.073	0.139	152.51	6.557	0.088
7. AES6 (FA countermeasure)	81.73	12.236	0.267	131.56	7.601	0.148	210.13	4.759	0.094
8. AES7 (Round key preprocess)	150.24	6.656	0.261	242.31	4.127	0.154	418.76	2.388	0.093
9. AES8 (MAO)	98.75	10.127	0.200	160.49	6.231	0.118	290.44	3.443	0.070
10. AES9 (MDPL)	40.88	24.461	0.199	71.27	14.032	0.119	114.26	8.752	0.075
11. AES10 (Threshold)	81.89	12.211	0.225	129.30	7.734	0.141	226.55	4.414	0.084
12. AES11 (WDDL)	92.40	10.823	0.192	155.62	6.426	0.119	262.05	3.816	0.077
13. AES12 (Pseudo RSL)	30.75	32.524	0.200	35.56	28.125	0.120	38.99	25.648	0.066
14. AES13 (Pseudo RSL)	92.34	10.829	0.216	147.84	6.764	0.127	251.07	3.983	0.066
15. AES_RDATA2	568.18	1.760	0.345	929.37	1.076	0.212	1420.5	0.704	0.129
16. AES_RDATA1	583.43	1.714	0.327	956.94	1.045	0.198	1483.7	0.674	0.125
17. Camellia	106.55	9.385	0.291	171.12	5.844	0.181	272.41	3.671	0.108
18. CAST	48.82	20.484	0.251	80.46	12.428	0.144	130.26	7.677	0.092
19. DES	256.67	3.896	0.298	419.11	2.386	0.183	681.66	1.467	0.099
20. ECC	107.70	9.285	0.150	170.77	5.856	0.089	277.24	3.607	0.054
21. MISTY1	42.05	23.784	0.292	68.58	14.581	0.166	108.61	9.207	0.111
22. RSA	62.02	16.123	0.241	104.54	9.566	0.136	167.84	5.958	0.086
23. SASEBO_ALGO_INPUT	50.36	19.856	0.317	83.34	11.999	0.181	127.89	7.819	0.115
24. SASEBO_ALGO_OUTPUT	659.63	1.516	0.284	1061.6	0.942	0.175	1655.6	0.604	0.107
25. SASEBO_INPUT	1485.9	0.673	0.211	2433.1	0.411	0.124	3703.7	0.270	0.075
26. SASEBO_REG	98.58	10.144	0.204	155.09	6.448	0.121	233.15	4.289	0.074
27. SASEBO_VALUE	–	–	–	–	–	–	–	–	–
28. SEED	36.33	27.529	0.255	60.85	16.435	0.150	97.59	10.247	0.096
29. T_DES	199.40	5.015	0.292	328.95	3.050	0.175	546.15	1.831	0.102

The 90-nm LSI adopts 7-layer metal wiring as the 130-nm LSI does. Figure 4.14 shows the signal wirings for each layer excluding power lines. The power supply to all the cells over the chip, including the widths and spaces of wires, is identical to that of the 130-nm LSI shown in Figure 4.6 through Figure 4.8.

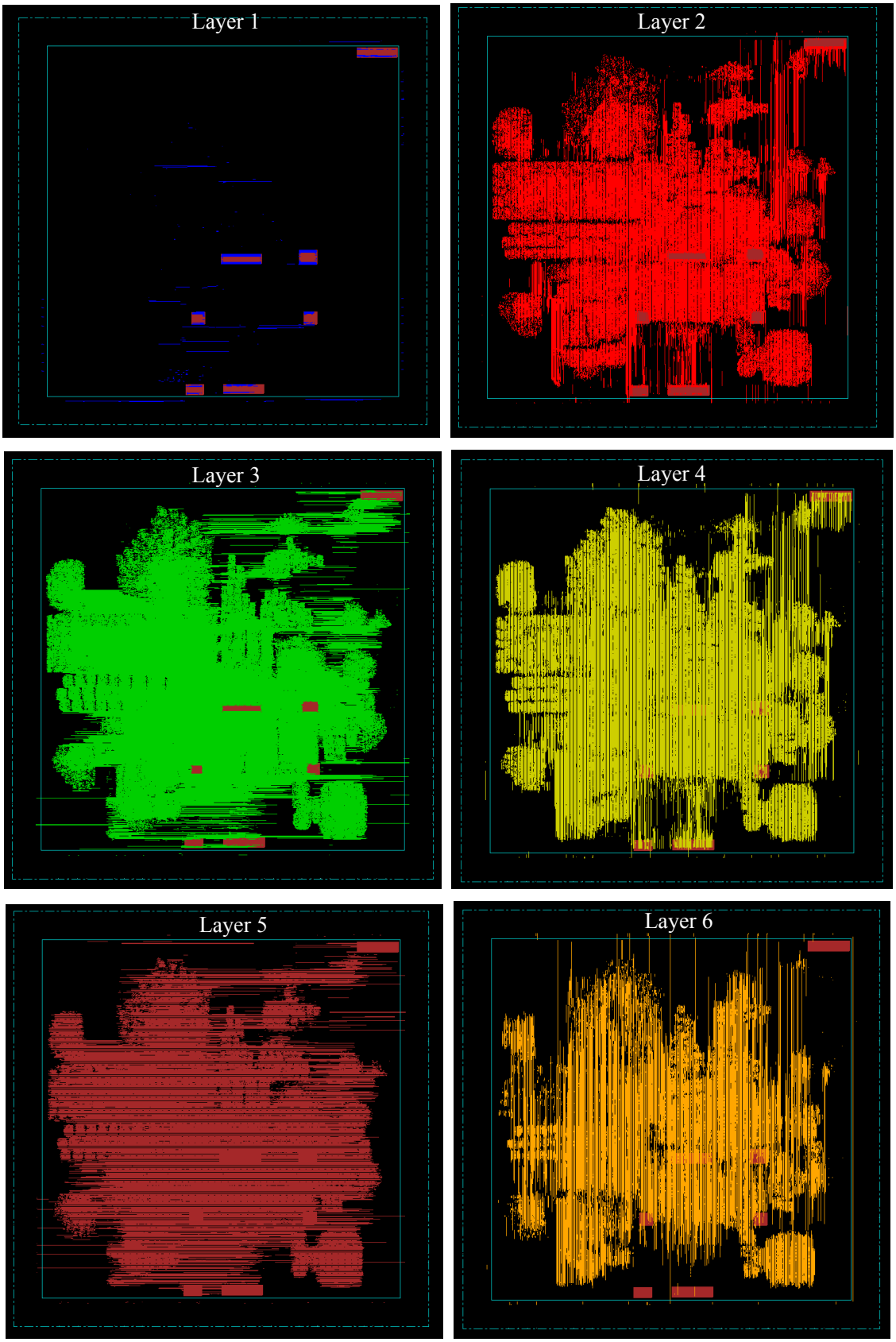


Figure 4.14　Signal Wiring Patterns of the Layers of 90-nm LSI (1/2)

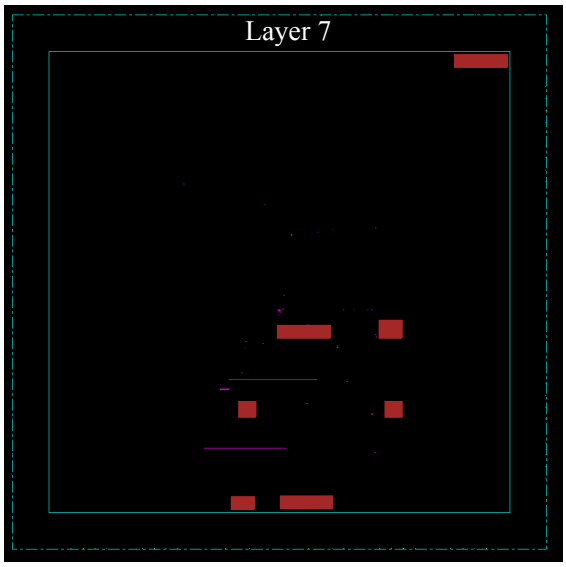


Figure 4.14　Signal Wiring Patterns of the Layers of 90-nm LSI (2/2)

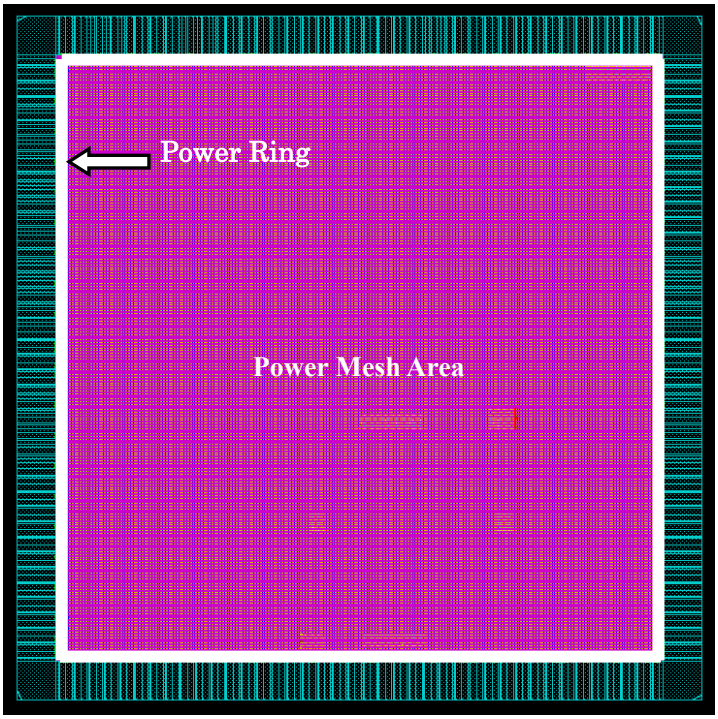


Figure 4.15　Power Wiring of the 90-nm LSI

Figure 4.16 shows the power line wiring patterns. Table 4.16 shows the power consumption and voltage drops with the assumption that the 30 % of all the cells are active. Figure 4.17 renders the voltage drops on the core power planes on the VDD and VSS sides in colors. Because only a single cryptographic core out of the 22 cores operates at a time in practice and the drop ratio of 0.3224 % is sufficiently small, a voltage drop will not cause a problem in normal operation.

Table 4.16　VDD/VSS Voltage Drops of the 90-nm LSI

	VDD	VSS
Frequency	24 MHz	
Transition probability	30 %	
Power consumption	49.9154 mW	
Worst drop	2.763 mV	3.224 mV
Drop ratio	0.2763 %	0.3224 %

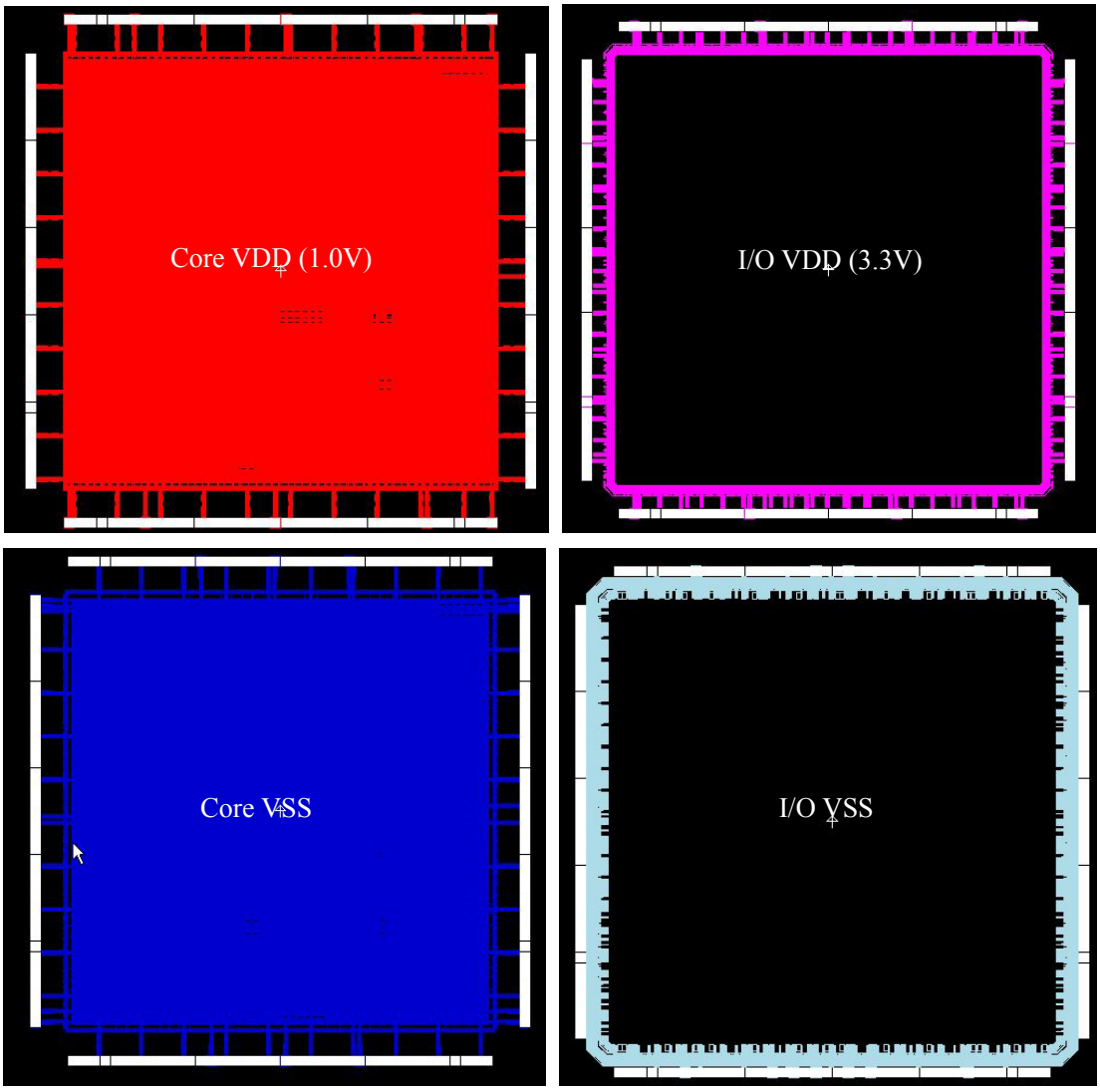
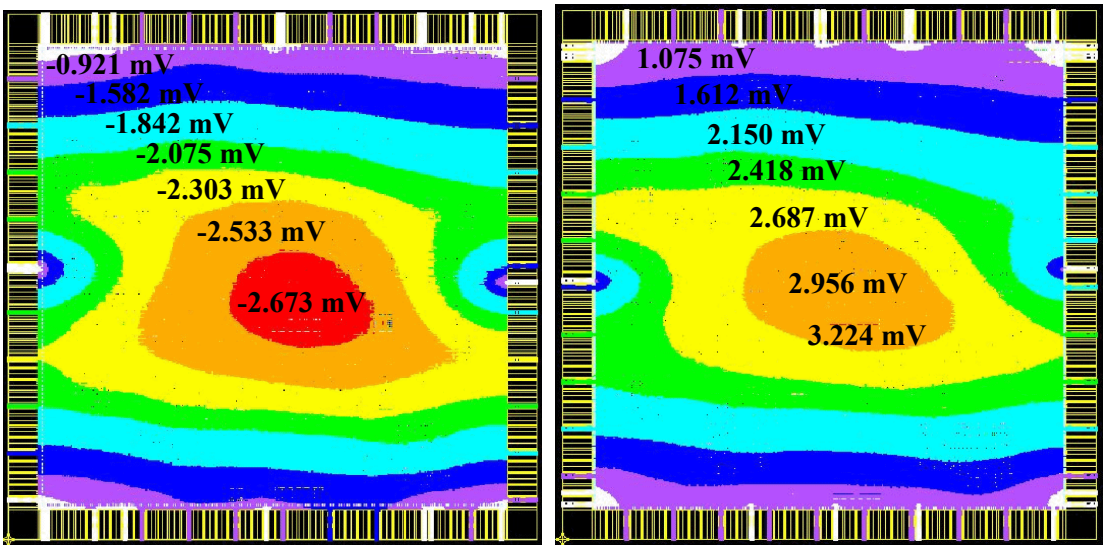


Figure 4.16　Power Line Wiring Patterns of the 90-nm LSI



(a) IR-DROP(VDD) (b) IR-DROP(VSS)

Figure 4.17 VDD/VSS Voltage Drops of the 90-nm LSI

5. CRYPTOGRAPHIC HARDWARE IPs

5.1 AES0 (Composite Field S-box)

Table 5.1 and Table 5.2 show the overview specifications of the AES cryptographic macro AES0 and the I/O ports of the macro, respectively. 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 5.1 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 5.2 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 5.1 and Figure 5.2, 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).

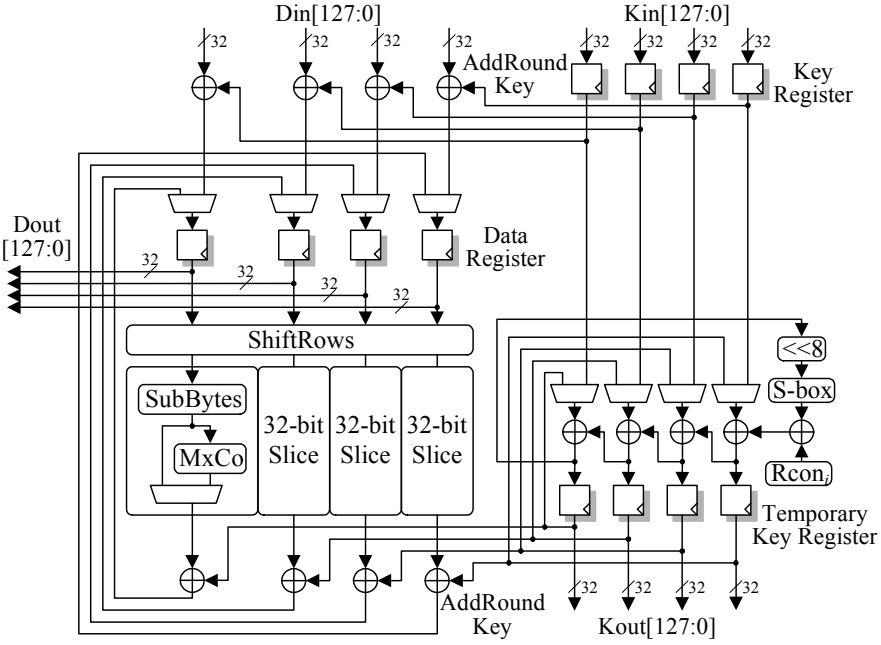


Figure 5.1 Encryption Datapath of AES0

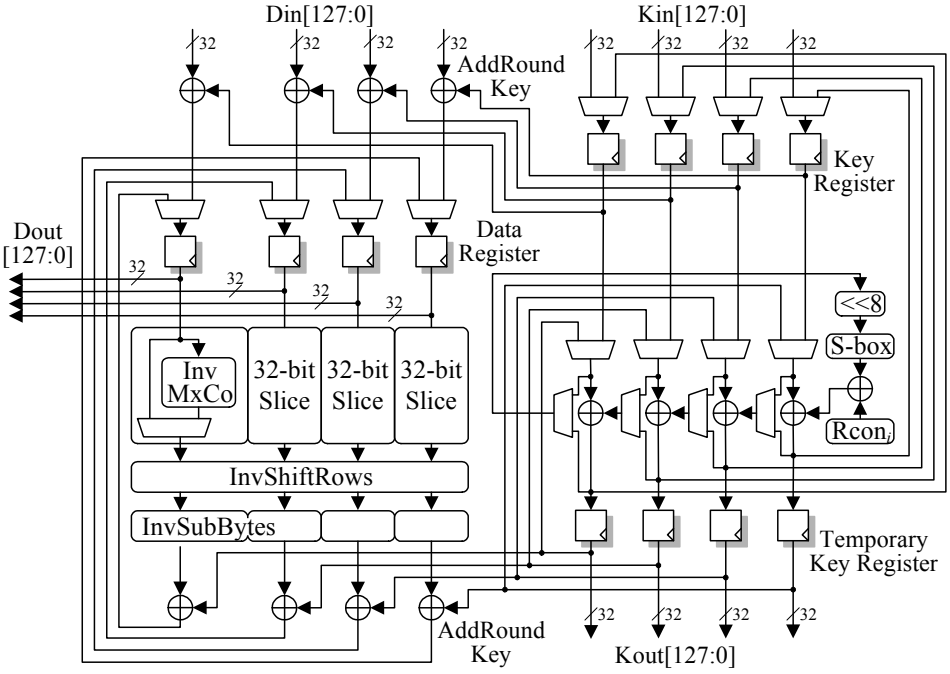


Figure 5.2 Decryption Datapath of AES0

Figure 5.3 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.

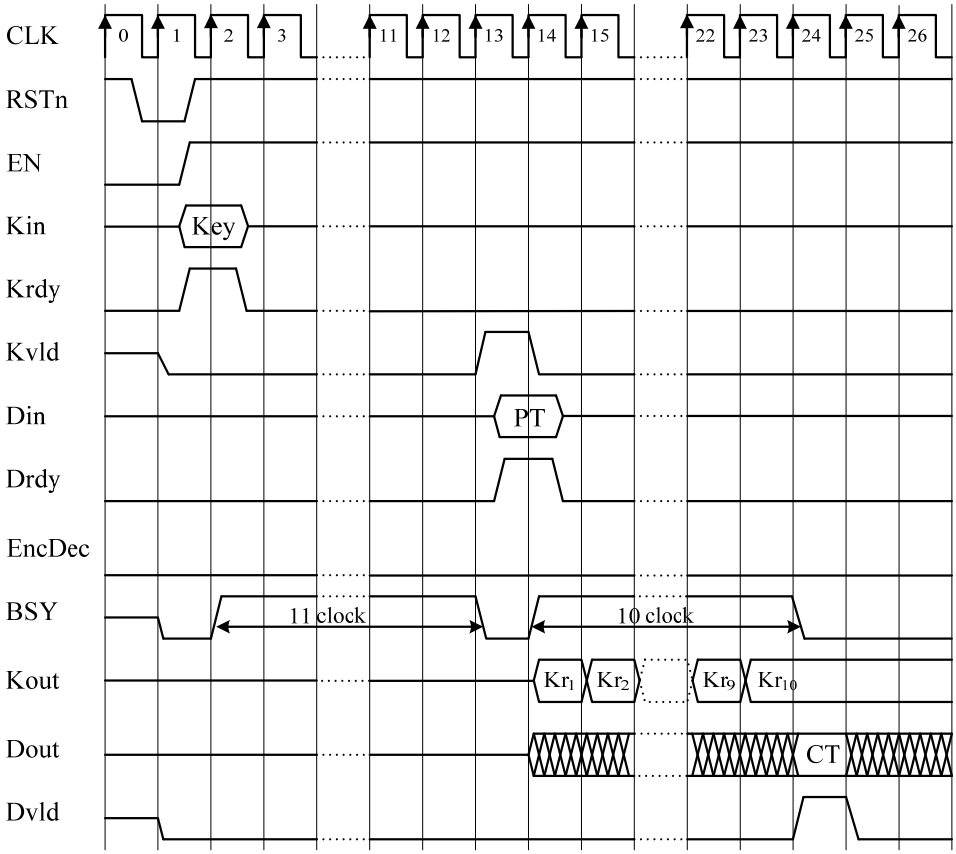


Figure 5.3 Timing Chart for Encryption on AES0

Figure 5.4 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.

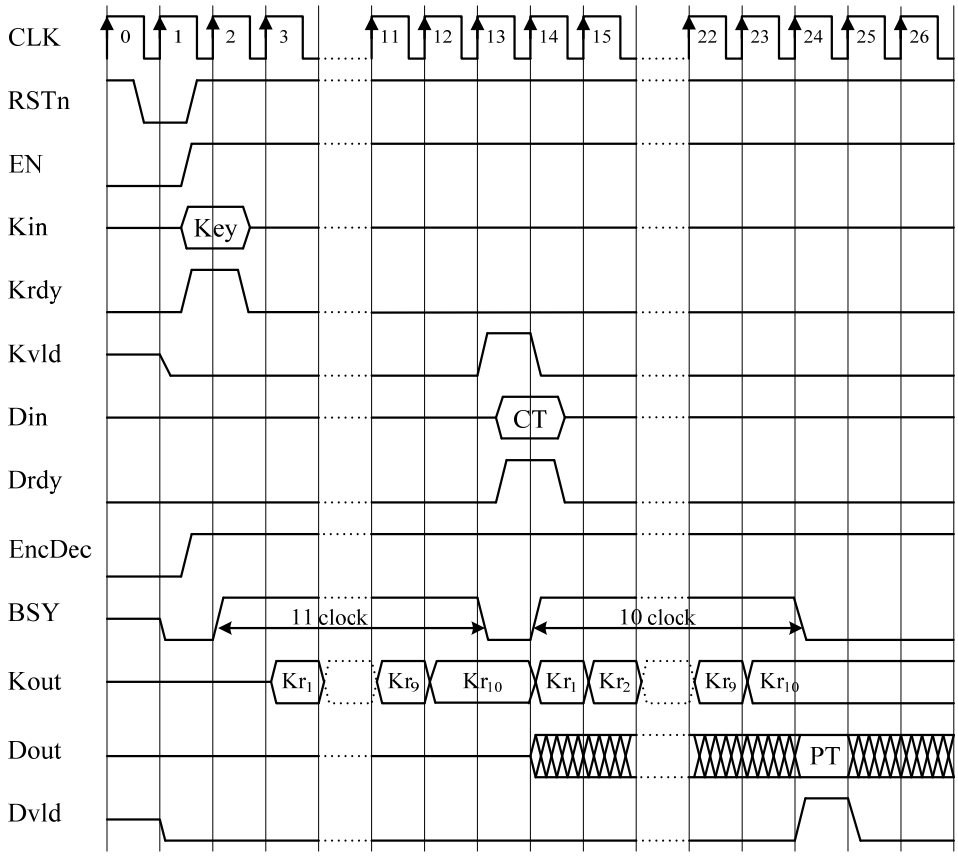


Figure 5.4 Timing Chart for Decryption on AES0

5.2 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) logic[4]. AES4 implements the multiplicative inverse circuit with a composite field[3]. 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 5.3 and Table 5.4, respectively. Although they have the same datapath architecture as that of the AES0 encryption circuit shown in Figure 5.1, their timing shown in Figure 5.5 differs from AES0 since they do not need the round key initialization at the time of secret key entry on the decryption circuit.

Table 5.3 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 5.4 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 5.5 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.

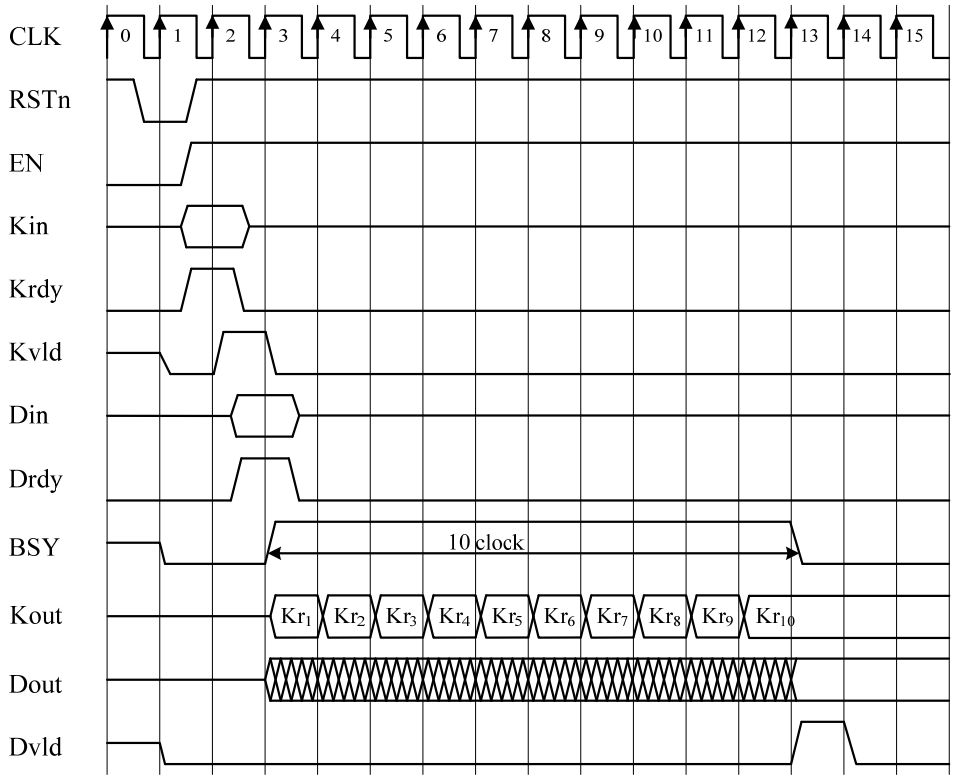


Figure 5.5 Timing Chart for AES1, AES2, AES3, and AES4

5.3 AES5 (CTR Mode)

The AES cryptographic macro AES5 supports the CTR mode of operation[5]. It has a 4-stage pipeline to achieve fast operation. Table 5.5 and Table 5.6 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 5.5 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 5.6 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 5.6 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 bits / 46 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.

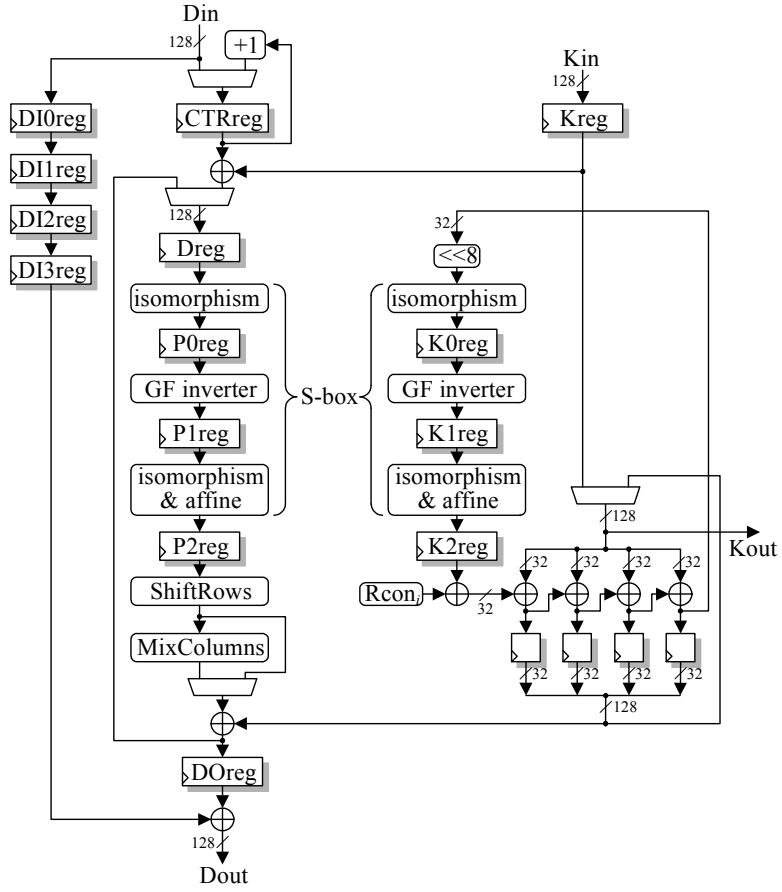


Figure 5.6 Datapath of AES5

Figure 5.7 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 been 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.

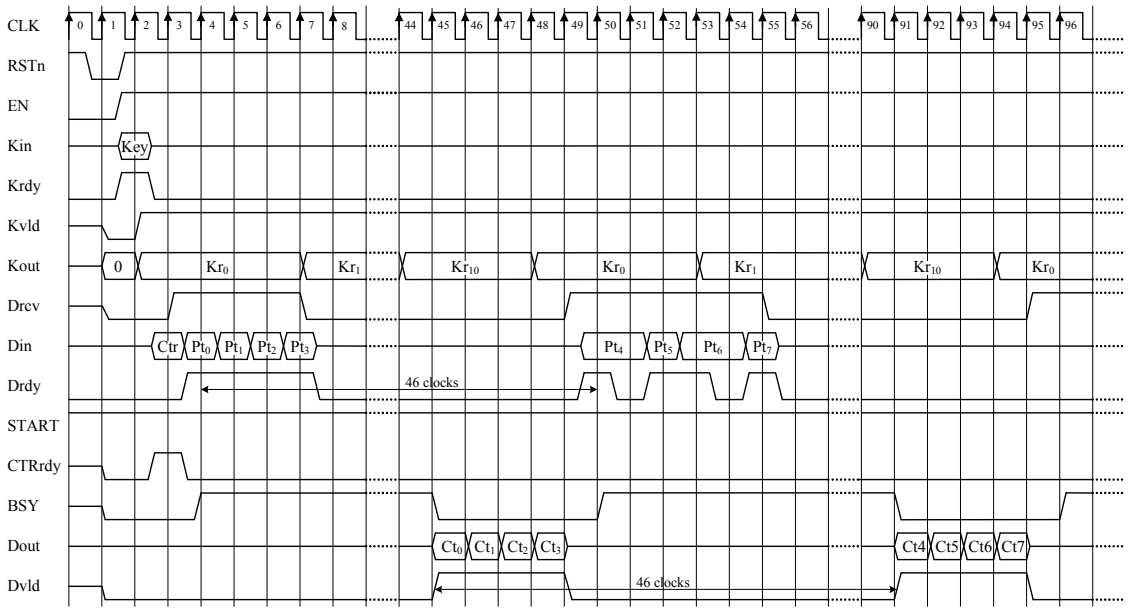


Figure 5.7-1 Timing Chart for AES5 Encryption/Decryption

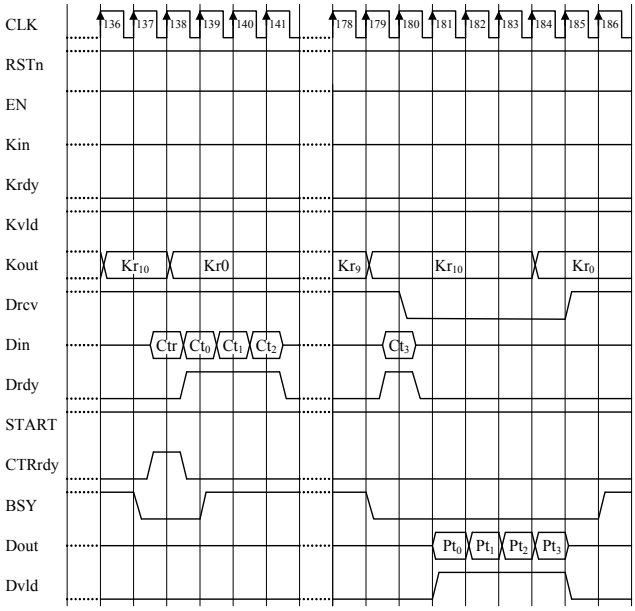


Figure 5.7-2 Timing Chart for AES5 Encryption/Decryption

Figure 5.8 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 5.7, 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 5.7 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.

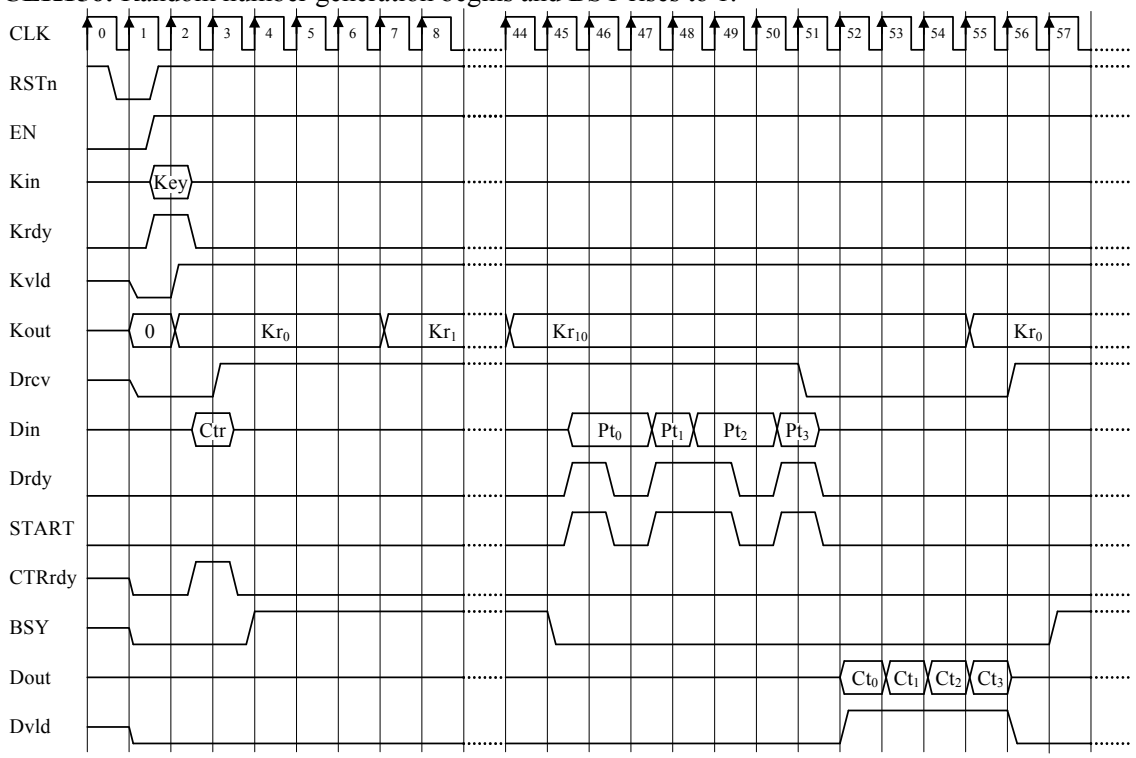


Figure 5.8-1 Timing Chart for AES5 with START Signal Control

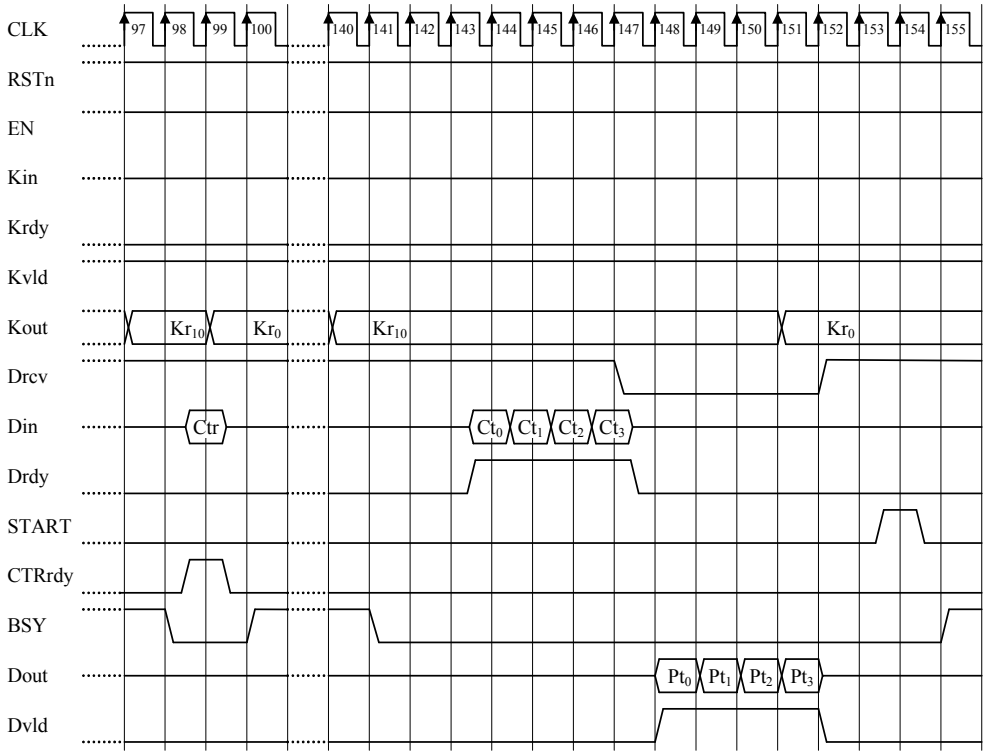


Figure 5.8-2 Timing Chart for AES5 with START Signal Control

5.4 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 5.7 and Table 5.8, 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 5.7 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 5.8 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 5.9 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.

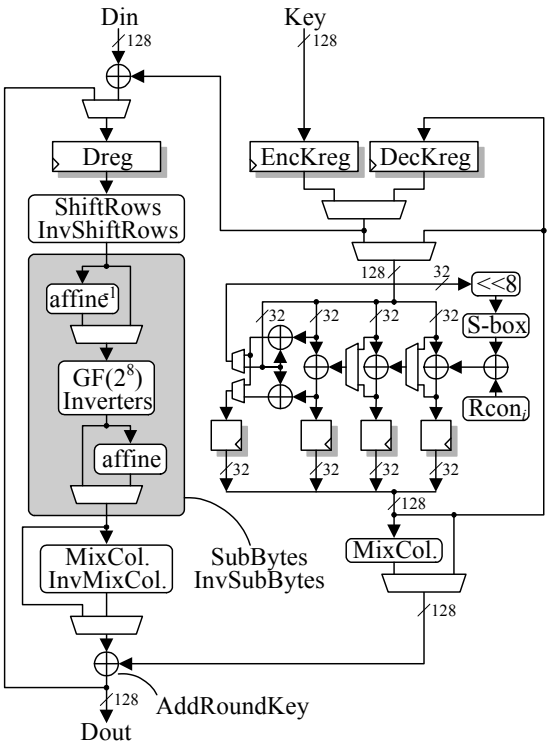


Figure 5.9 Typical Datapath Architecture of AES with the Core Components Shared by Encryption and Decryption

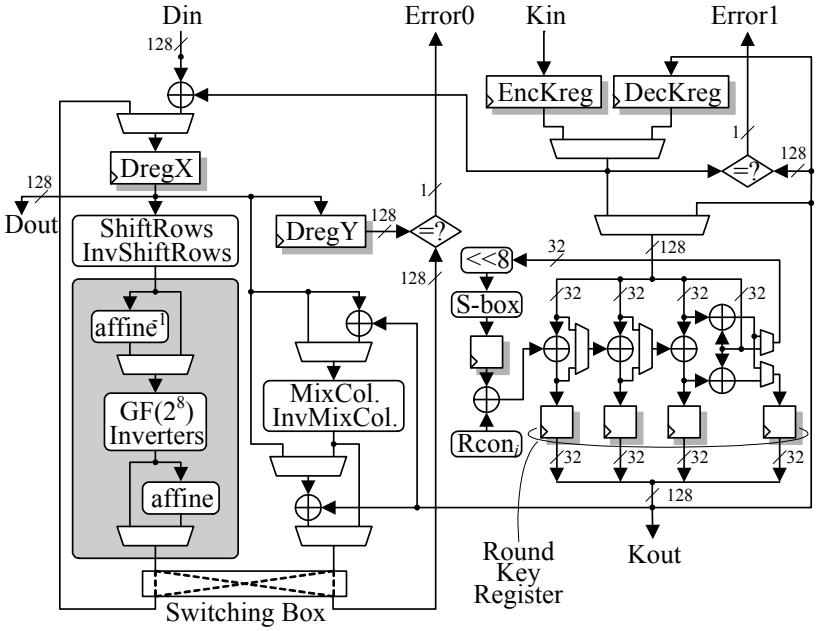


Figure 5.10 Encryption and Decryption Circuit of AES6

Figure 5.10 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 5.9, 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.

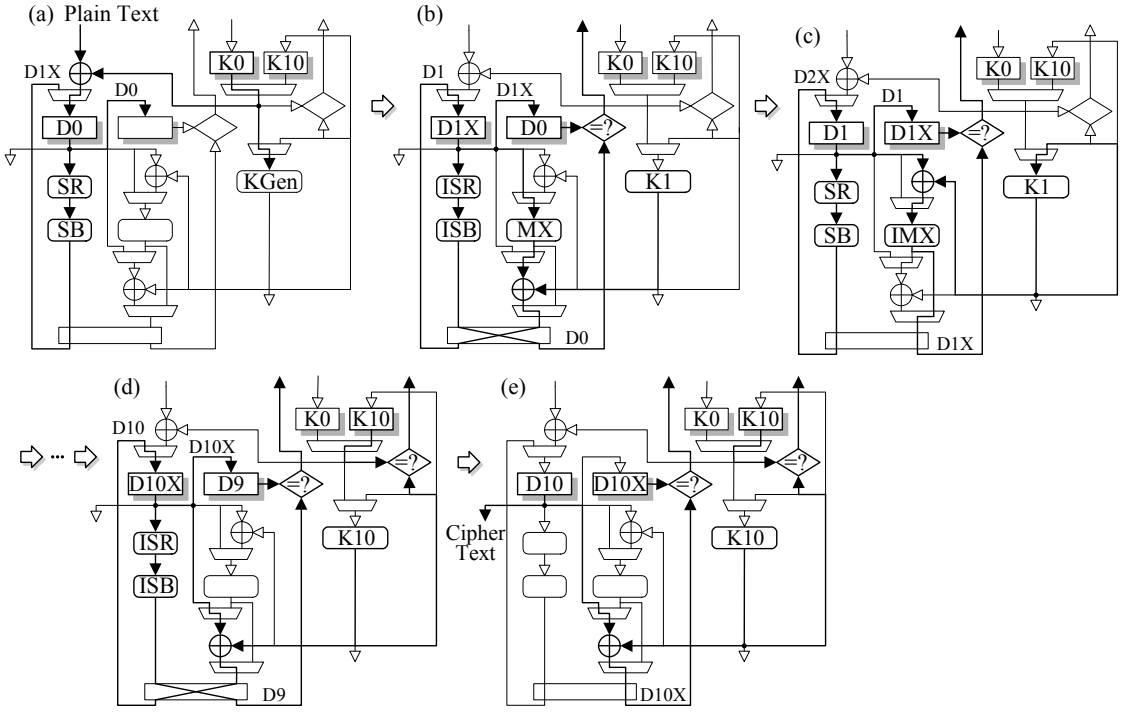


Figure 5.11 Encryption Operation Examples of AES6

Figure 5.11 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 though 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 × 2 cycles and 1 cycle for (e).

5.5 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 5.9 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 5.10 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 5.12 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 5.1. The entry of a secret key at Kin 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.

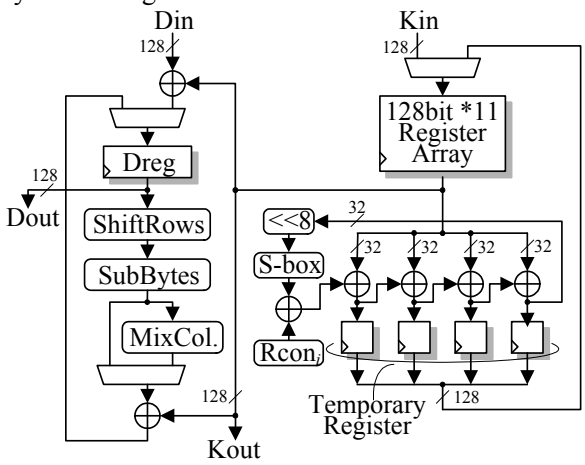


Figure 5.12 Datapath Architecture of AES7

The encryption timing for AES7 with the minimum possible cycles is shown in Figure 5.13. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.
CLK2: Krdy=1 transfers the 128-bit secret key Key presented on Kin 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 Pt0 presented on Din is XORed with the first round key Kr0 (This is the secret key Key input through Kin.) output on the key registers. Drdy=1 loads the result of XOR into the data register Dreg. The 128-bit port Kout outputs Kr0.
CLK16: 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 the second round key Kr1. Likewise, 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.
CLK17~26: Encryption takes 10 clocks and completes at CLK25. Dout presents the ciphertext, BSY falls to 0, and Dvld turns to 1 at CLK26. The next plaintext Pt1 can be input at CLK26.

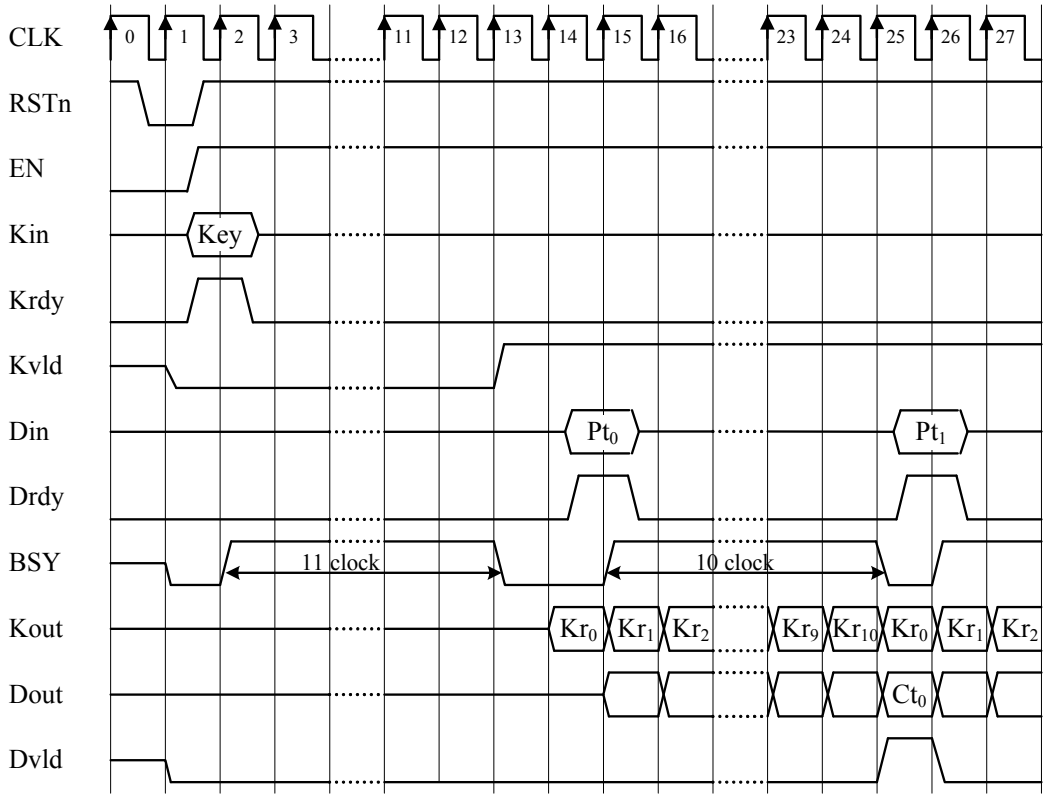


Figure 5.13 Timing Chart for Encryption on AES7

5.6 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 5.14 illustrates the basic structure of a Masked-AND gate. The original input data <a, b> are XOR-masked with random numbers <m_a, m_b> that are independent with each other, resulting in the gate inputs <$\tilde{a}$, $\tilde{b}$ >. The gate outputs $(a \bullet b) \oplus m$, 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 <a, b> or output $a \bullet b$. However, it has been reported that the gate risks secret information leaking through power consumption due to glitches caused by signal delay variations.

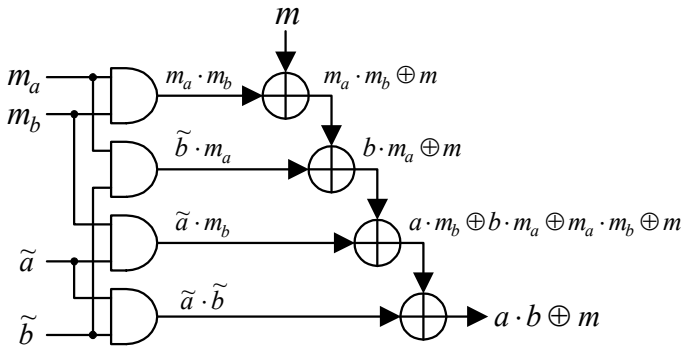


Figure 5.14 Masked-AND Gate

5.7 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 WDDL[9] with random number masking. Figure 5.15 shows the main components and the basic construction of the MDPL. Figure 5.15 (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 MDLP-AND gate is shown in Table 5.11.

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

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 $\overline{m}$) as shown in Figure 5.15 (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.

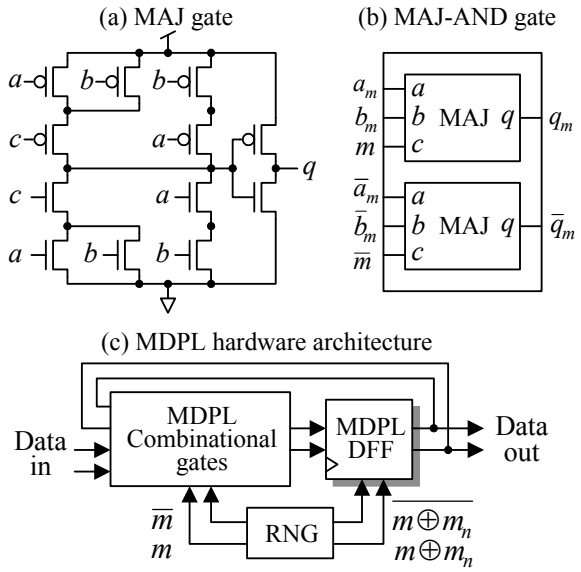


Figure 5.15 Masked Dual-rail Precharge Logic

Table 5.11 Truth Table for the MDPL-AND Gate

a	b	m	a_m	b_m	q_m	$\overline{m}$	$\overline{a}_m$	$\overline{b}_m$	$\overline{q}_m$
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

5.8 AES10 (Threshold Implementation)

AES10 employs the Threshold implementations[7], 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 addition and summation over GF(2^m), respectively. The input variables are represented as $x = \bigoplus_{i=1}^{n} x_i$ and $y = \bigoplus_{i=1}^{n} y_i$, 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.

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

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

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

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

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

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.

5.9 AES11 (WDDL)

AES11 implements Wave Dynamic Differential Logic (WDDL)[9], the DPA countermeasure proposed by Tiri et al. Figure 5.16 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.

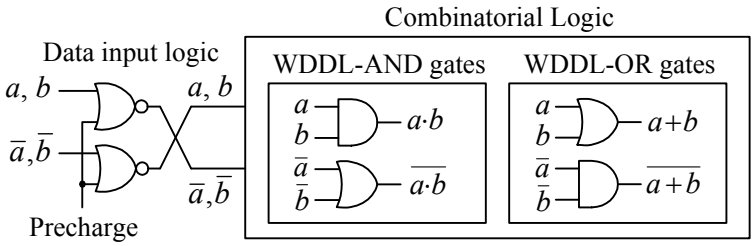


Figure 5.16 Wave Dynamic Differential Logic

5.10 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 5.17 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:

Input: $\overline{en}$, $\begin{cases} x = a \oplus r_x \\ y = b \oplus r_y \end{cases}$, $\begin{cases} r_z \\ r_{xz} = r_x \oplus r_z \\ r_{yz} = r_y \oplus r_z \end{cases}$ 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: $\text{RSL-NAND}(x_z, y_z, r_z, \overline{en})$ (Applies the input data to the RSL-NAND gate)

Process 4 $\overline{en} = 0$ (Enables the output after stabling 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 5.18 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.

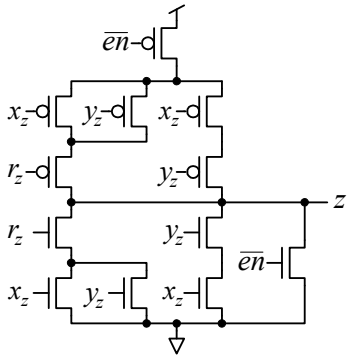


Figure 5.17 RSL-NAND Gate

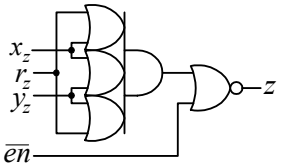


Figure 5.18 Pseudo RSL-NAND Gate

5.11 Camellia

The overview specifications and I/O ports of the cryptographic circuit macro for Camellia[11] are shown in Table 5.12 and Table 5.13, 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 5.12 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 5.13 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 5.19 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 Kl, 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 Kl registers' data on-the-fly.

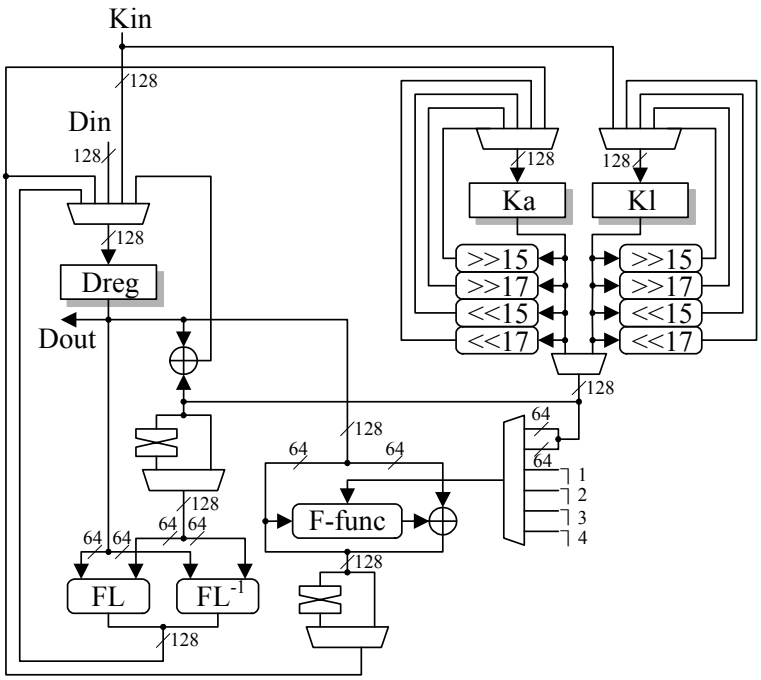


Figure 5.19 Datapath Architecture of Camellia

Figure 5.20 and Figure 5.21 illustrate the timings for key scheduling and encryption for Camellia each with the minimum possible cycles, respectively.

CLK1: RSTn=0 resets the control circuit.
CLK2: Krdy=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 Krdy 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, Drdy=1 loads the plaintext presented on the 128-bit input port Din into the data register Dreg.
CLK10: 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 the round keys Kw1 and Kw2. Likewise, 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.
CLK32: Encryption completes in 23 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.
CLK33: Drdy=1 initiates the next operation. At this clock, with EncDec=1 for decryption, the 128-bit port Din latches the ciphertext.
CLK34: 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.
CLK57: Decryption finishes in 23 clock cycles. Dout outputs the plaintext and BSY turns to 0. Dvld turns 1 for a single clock cycle.

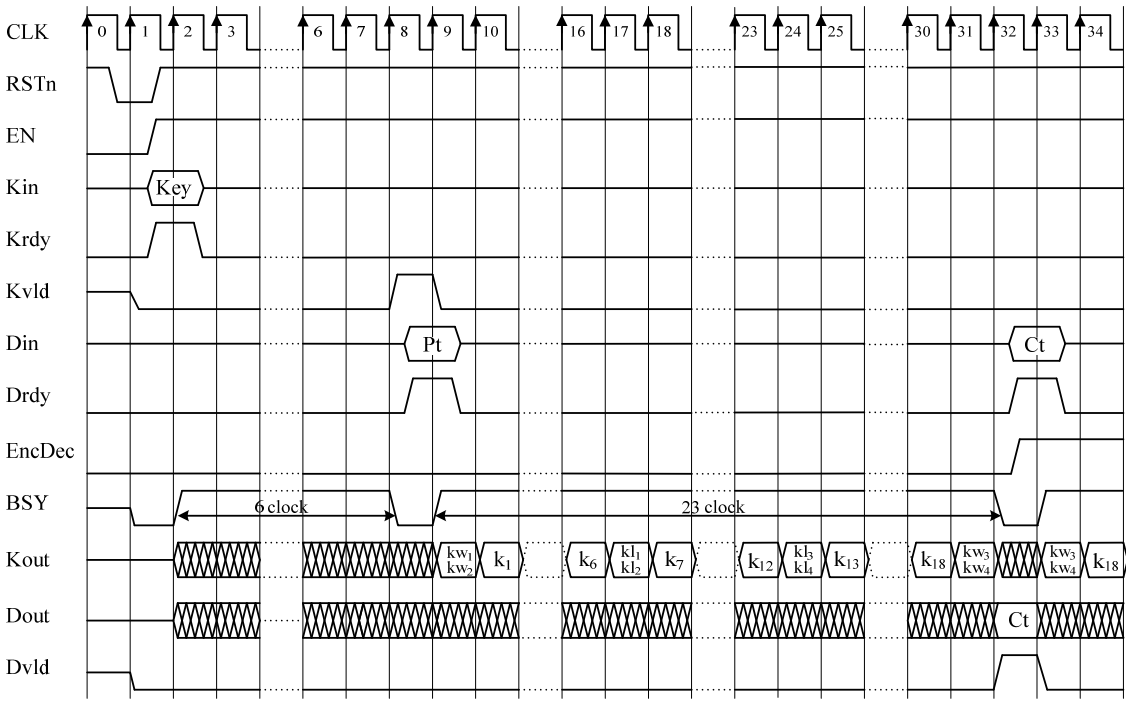


Figure 5.20 Timing Chart for Key Scheduling and Encryption of Camellia

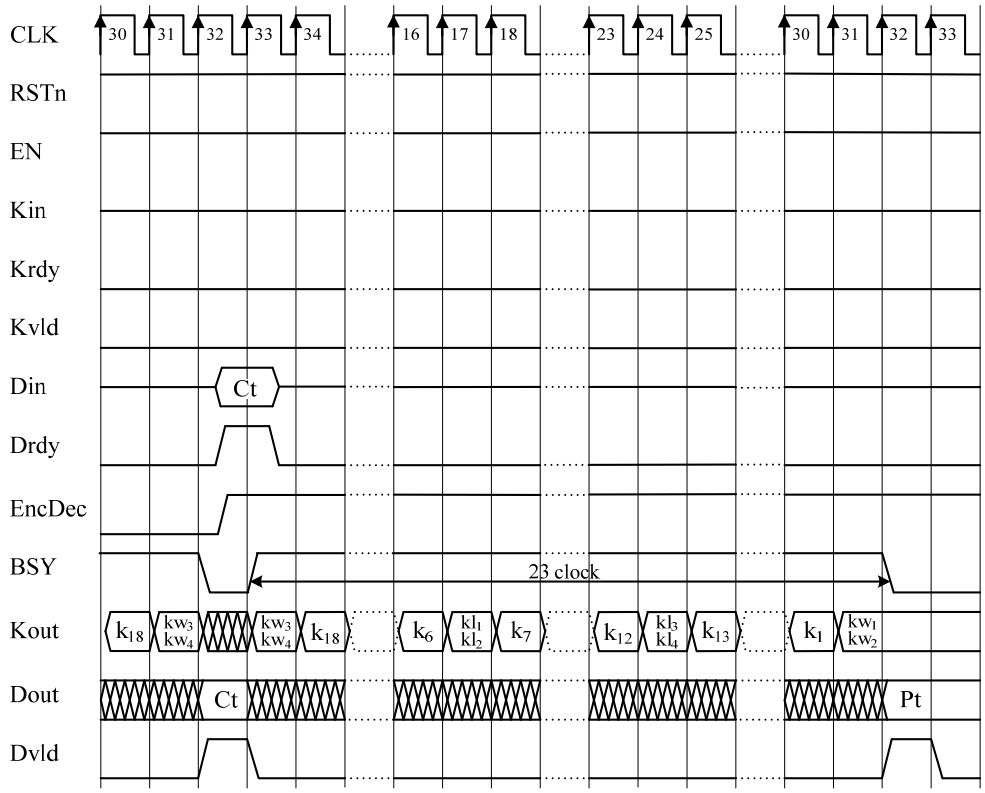


Figure 5.21 Decryption Timing Chart for Camellia

5.12 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 5.14 and Table 5.15, respectively.

Table 5.14 Overview Specifications of CAST-128

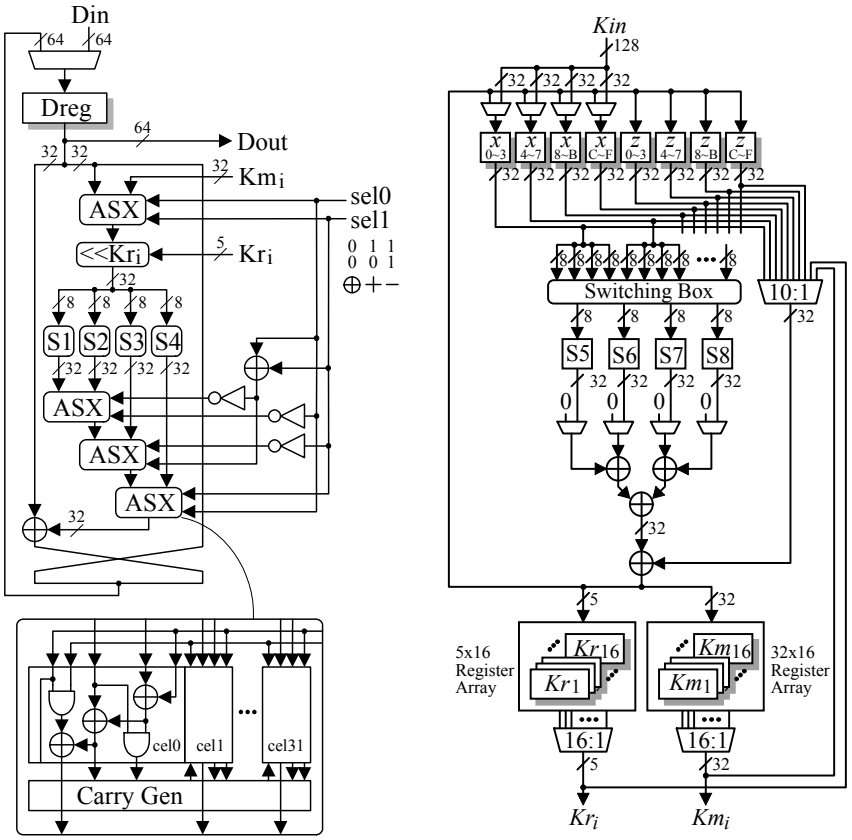
Algorithm	CAST-128
Data block length	64 bits
Key length	128 bits
Function	Encryption/Decryption
Mode of operation	Electronic Code Book (ECB)
Source file	CAST128.v
Description language	Verilog-HDL
Top module	CAST
S-box	Table implementation
Throughput	64 bits / 17 clocks
Round key generation	Pre-calculation

Table 5.15 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 5.22 represents the datapath architecture of the CAST-128[13] 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.



(a) Encryption Part (b) Key Scheduler

Figure 5.22 Datapath Architecture of CAST-128

Figure 5.23 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: Krdy=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 Krdy 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, Drdy=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: Drdy=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.

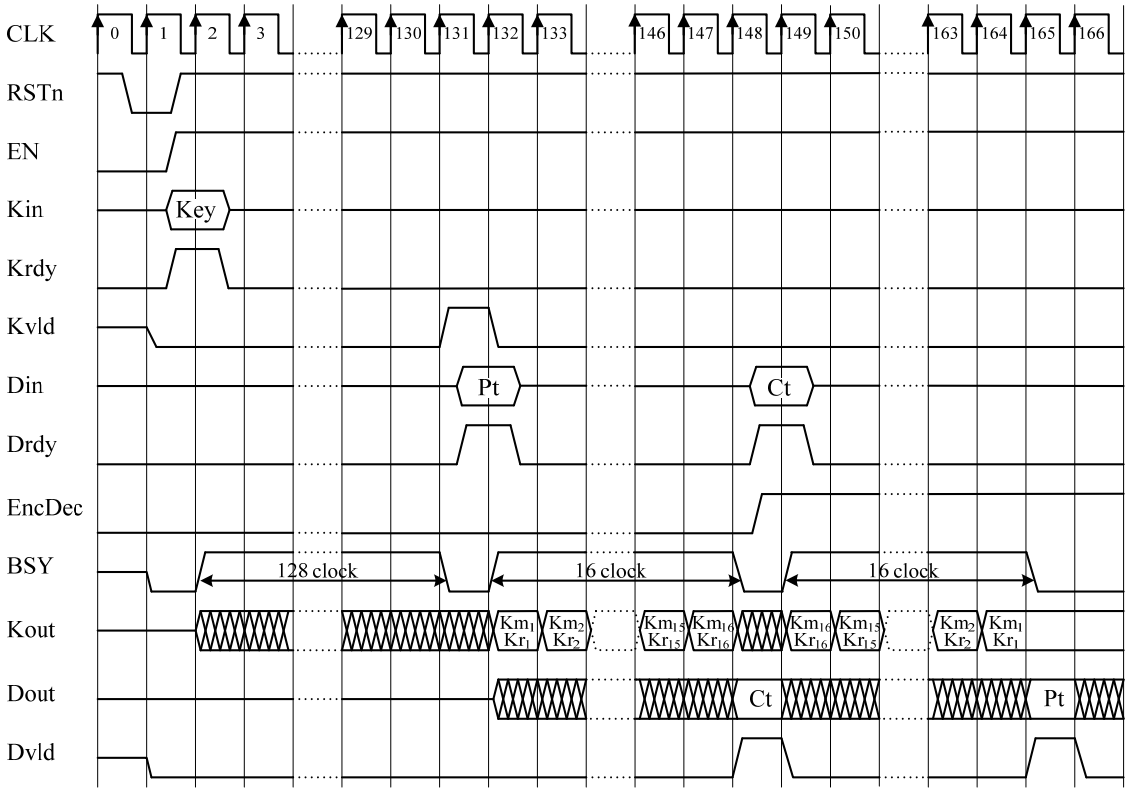


Figure 5.23 Timing Chart for CAST-128

5.13 DES

The overview specifications and I/O ports of the cryptographic circuit macro for DES[14] are shown in Table 5.16 and Table 5.17, respectively. DES is a block cipher with the Feistel structure, and is thus suitable for small-footprint implementations.

Table 5.16 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 5.17 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.
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 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 5.24 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.

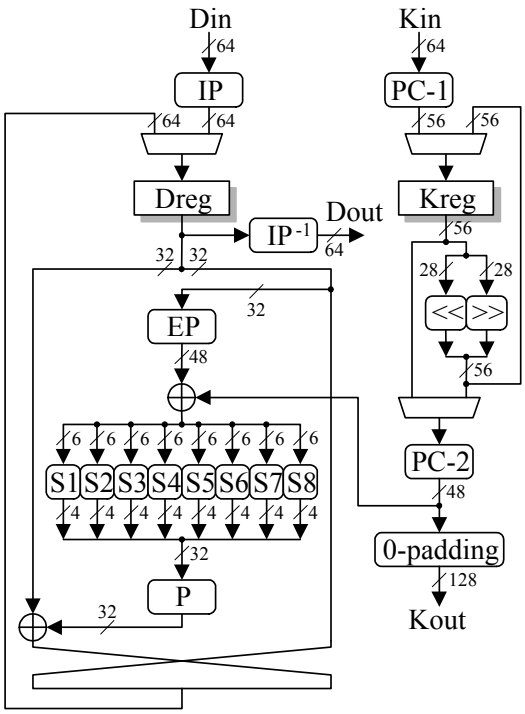


Figure 5.24 Datapath Architecture of DES

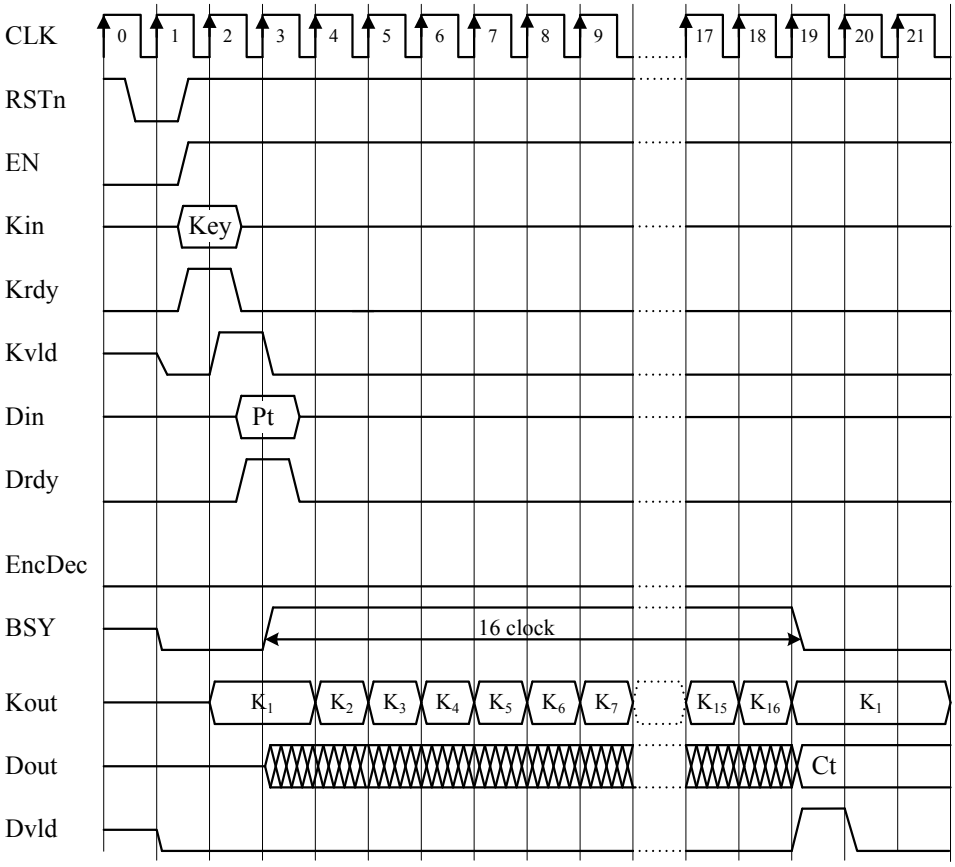


Figure 5.25 Timing Chart for DES

Figure 5.25 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.

5.14 ECC

The overview specifications and I/O ports of the cryptographic circuit macro for ECC are shown in Table 5.18 and Table 5.19, respectively. The memory map of internal variables of 16 words × 64 bits is shown in Table 5.20. 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(2^{61}) defined with the irreducible polynomial:

$$f(x) = x^{61} + x^5 + x^2 + x + 1.$$

To take the 61-bit data of the x-coordinate of the initial point through the 32-bit input port, the macro adds 3-0 bits to the upper side of the 61-bit x-coordinate, splits the resulting 64-bit data into 2 32-bits, and inputs them in two cycles. For inputting the 64-bit scalar value for elliptic multiplication, which is the key, the macro assumes the 65th bit on the MSB side is 1. In other words, in the Algorithm 1: Montgomery Powering Ladder method described below, it is dealt with as n-2 = 65-2 = 63. The key will be stored in the dedicated 64-bit register, instead of in the memory shown in Table 5.20.

The 64 bits (the lower 61 bits are valid) at memory address 3 stores the x-coordinate of the initial point in affine coordinates for scalar multiplications. Address 4 loads the parameter b of the elliptic curve E. Address 5, 6, and 7 house intermediate values used during additions and doublings of points. Address 5 also stores a random number when MODE=1. The variables (X_1, Z_1) denoting the initial point in projective coordinates stored in Address D and 9. The variables (X_2, Z_2) representing the doubled point of (X_1, Z_1) go to Address E and A. The inputs in projective coordinates are prepared to support the side-channel attack countermeasure using a random value (polynomial remainder with mod $f(x)$ except for 0) in the z-coordinate which will be implemented in the future. Because the countermeasure is not implemented in the present version, any fixed value for Z_1 may be specified such as Z_1=1 instead of a random number.

Note that, in case that the intermediate result becomes the point of infinity, correct operations may not be performed because the ECC macro does not check for the point of infinity.

Table 5.18 Overview Specifications of ECC

Algorithm	Elliptic scalar multiplications with the Montgomery Power Ladder algorithm over GF(2^{61})
Data block length	61 bits (with 3-0 bits concatenated to the upper side)
Key length	64 bits (scalar)
Function	Elliptic scalar multiplications over GF(2^{61})
Source file	uec_input_to_ECC_OS.v
Description language	Verilog-HDL
Top module	uec_ECC_OS

Table 5.19 I/O Ports of ECC

Port name	Direction	Bit width	Description
Kin	In	32	Key input. The 64-bit key (scalar) and 32-bit random number (unused for the macro of the present version) are taken in 3 clock cycles.
Din	In	32	The 32-bit port Din takes 6 clock cycles to import the following 3 64-bit data each with the upper 3 bits padded with 0s: the initial point's x-coordinate in affine coordinates, Z in projective coordinates, and the curve parameter b, dividing each into 2 cycles.
Dout	Out	32	The 32-bit port Dout outputs the 64-bit data, with the upper 3 bits padded with 0s, of the x-coordinate of the result of elliptic scalar multiplication in 2 cycles.
Krdy	In	1	At the clock when Krdy turns to 1 and the next clock, the internal register latches the key divided into two 32-bit words given to Kin.
Drdy	In	1	Turning Drdy to 1 for one clock cycle begins taking the contiguous 6 32-bit data into the internal registers from the next clock, and subsequently elliptic scalar multiplications take place.
MODE	In	3	For the present version, MODE shall be set "000" for the Montgomery powering ladder method. It is planned that the future version will support MODE="001" for operations with side-channel attack countermeasures.
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 ECC macro.
CLK	In	1	Every internal register captures input data synchronously on the rising edge of the clock signal CLK.
BSY	Out	1	The busy status flag BSY falls to 0 immediately after producing the result of elliptic scalar multiplication, as well as when resetting. Otherwise it indicates 1.
Kvld	Out	1	When key loading completes, Kvld goes to 1 for one clock cycle and returns to 0. Immediately after Kvld turns to 0, data loading can be started.
Dvld	Out	1	When scalar multiplication completes, Dvld goes to 1 for one clock cycle and returns to 0. Meanwhile, Dout exports the x-coordinate of the elliptic scalar multiplication point in 2 clock cycles.

Table 5.20 Memory Map of ECC

Address	Purpose	Address	Purpose
0	Reserved	8	Reserved
1	Reserved	9	Z_1
2	Reserved	A	Z_2
3	x	B	Reserved
4	b	C	Reserved
5	t_1 (Rnd)	D	X_1
6	t_2	E	X_2
7	t_3	F	Reserved

The ECC macro adopts, for elliptic scalar multiplication, the Lopez and Dahab's algorithm[16], which is the Montgomery Powering Ladder method[15] in projective coordinates. It also employs the Montgomery multiplication algorithm[17] proposed by Knezevic et al, for modular multiplication. The following demonstrates these algorithms as Algorithm 1, Algorithm2, and Algorithm3:

- Algorithm 1: Montgomery Powering Ladder method

Input: A point on the elliptic curve: P
Positive integers: $k = (1k_{n-2} \cdots d_1 d_0)_2$

Output: X-coordinate of kP: kP_x

1: $P_1 \leftarrow P, P_2 \leftarrow 2P$
2: **for** i=n-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** P_{1x}

- Algorithm 2: Lopez and Dahab's algorithm. Montgomery Powering Ladderin Projective Coordinates

Input: $P_1 = (X_1, Z_1), P_2 = (X_2, Z_2), x = (P_2 - P_1)_X$
Output: $P_1 = P_1 + P_2$

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: $X_1 \leftarrow xZ_1 + t_1$
7: **return** P_1

Input: $P_1 = (X_1, Z_1)$
Output: $P_1 = 2P_1$

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: $X_1 \leftarrow bt_3 + t_2$
7: **return** P_1

- Algorithm 3: Knezevic's algorithm. Barrett Reduction over $GF(2^n)$ without precomputation

Input: Polynomial basis: $A(x) = \sum_{i=0}^{2n} a_i x^i,\ M(x) = x^n + \sum_{i=0}^{l} m_i x^i$

where $l = \left\lfloor \frac{n}{2} \right\rfloor,\ a_i, m_i \in \{0,1\}$

Output: $R(x) = A(x) \bmod M(x)$

1: $Q_1(x) \leftarrow A(x) \operatorname{div} x^n$
2: $Q_2(x) \leftarrow M(x) Q_1(x)$
3: $Q_3(x) \leftarrow Q_2(x) \operatorname{div} x^n$
4: $R_1(x) \leftarrow A(x) \bmod x^n$
5: $R_2(x) \leftarrow M(x) Q_3 \bmod x^n$
6: $R(x) \leftarrow R_1(x) + R_2(x)$
7: **return** $R(x)$

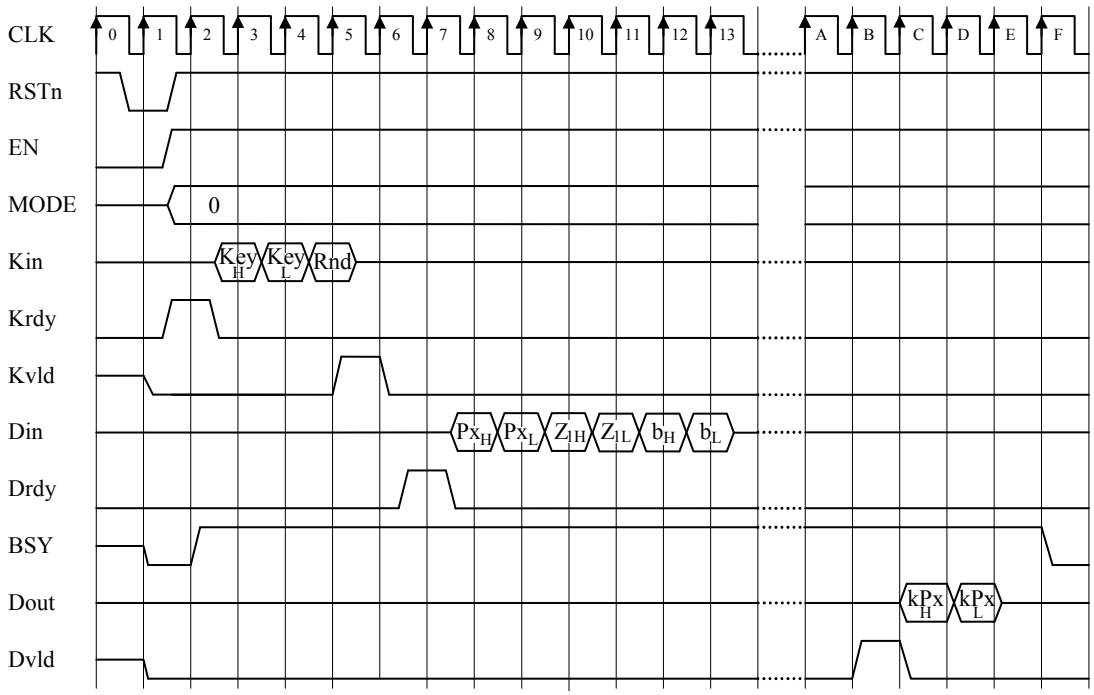


Figure 5.26 Timing Chart for the Elliptic Scalar Multiplication of ECC

Figure 5.26 illustrates the timing for the ECC macro. Only the Montgomery Powering Ladder method (MODE=0) is allowed to specify the elliptic scalar multiplication algorithm. The operation(s) within each clock cycle follow:

CLK1: RSTn=0 resets the control circuit.
CLK2: When EN=1, Krdy=1 loads the key (scalar) after this clock.
CLK3: Key_H, the upper 32 bits of the Key, presented on the 32-bit port Kin is latched. The busy signal BSY turns to 1.
CLK4: Key_L the lower 32 bits of the Key, are latched.
CLK5: The random number Rnd for side-channel attack countermeasure is latched. (The countermeasure is not implemented yet.)
CLK6: Inputting the key has finished, and Kvld goes to 1 for one clock cycle.
CLK7: Drdy=1 makes the circuit ready to begin entering data in Din from the next clock.
CLK8,9: Din continuously takes $P_{x\mathrm{H}}$ and $P_{x\mathrm{L}}$, the upper and lower 32bits, respectively, of P_x, the x-coordinate of P.
CLK10~13: Din takes the Z_1 in projective coordinates and the curve parameter b, each divided into two 32-bit words, in 4 clock cycles.
CLK14~: The macro performs the scalar multiplications in approximately 7,800 clock cycles.
CLKC: The multiplications complete, and Dvld becomes 1 to indicate that the result will be come out from the next clock.
CLKD: Dout exports $kP_{x\mathrm{H}}$, the upper 32 bits of the x-coordinate of the point kP, the result of the elliptic multiplying with k (k is the value of the key Key).
CLKE: Dout exports kP_{xL}, the lower 32 bits of the x-coordinate of kP.

5.15 MISTY1

The overview specifications and I/O ports of the cryptographic circuit macro MISTY1[18] are shown in Table 5.21 and Table 5.22, respectively. MISTY1 is a block cipher with a nested Feistel structure.

Table 5.21 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 5.22 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 5.27 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.

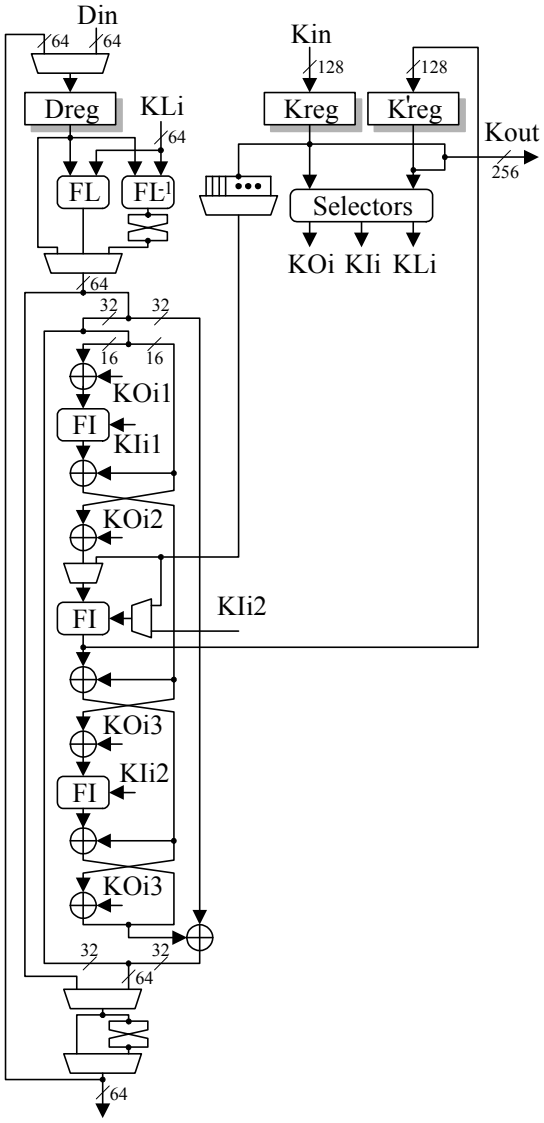


Figure 5.27 Datapath Architecture of MISTY1

Figure 5.28 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 K'reg. 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.

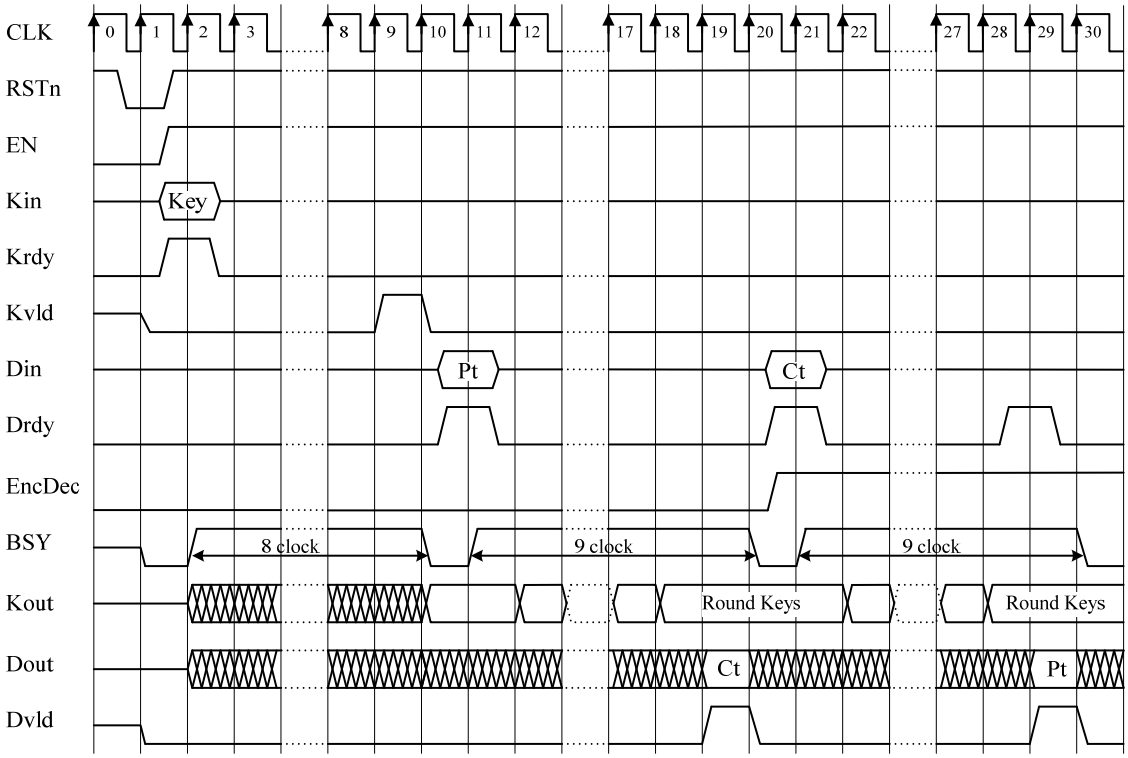


Figure 5.28 Timing Chart for MISTY1

5.16 RSA

The overview specifications and I/O ports of the cryptographic circuit macro for RSA[19] are shown in Table 5.23 and Table 5.24, 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 methods[20]), it employs the following countermeasures against side-channel attacks: the square-and-multiply always method (countermeasure with dummy operation)[21], Montgomery Powering Ladder[22], and Square-Multiply exponentiation method[23]. 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 5.23 Overview Specifications of RSA

Algorithm	RSA
Data block length	512 bits
Key length	512 bits
Function	• CRT mode (non-CRT/CRT) • Modular exponentiation operations 0) Left binary method 1) Right binary method 2) Left binary method with dummy multiplication[21)] 3) Right binary method with dummy multiplication[21)] 4) Montgomery powering ladder[22)] 5) Square-multiply exponentiation method[23)]
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 5.24 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 moduli 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.

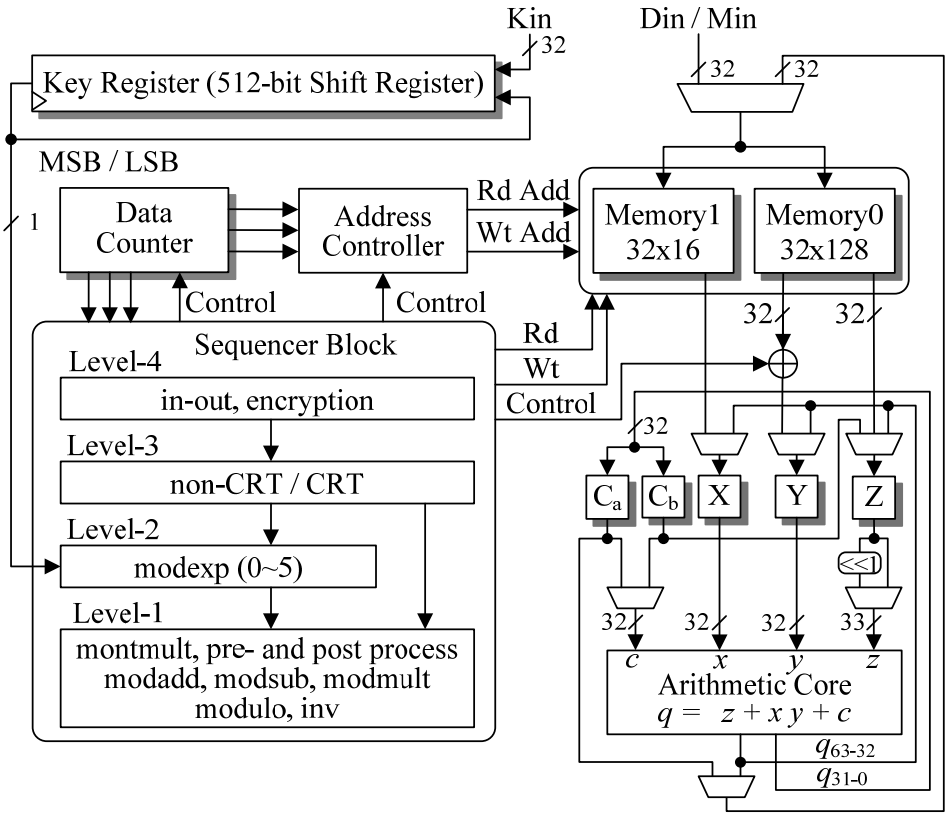


Figure 5.29 Circuit Architecture of RSA

Figure 5.29 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 5.30 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.

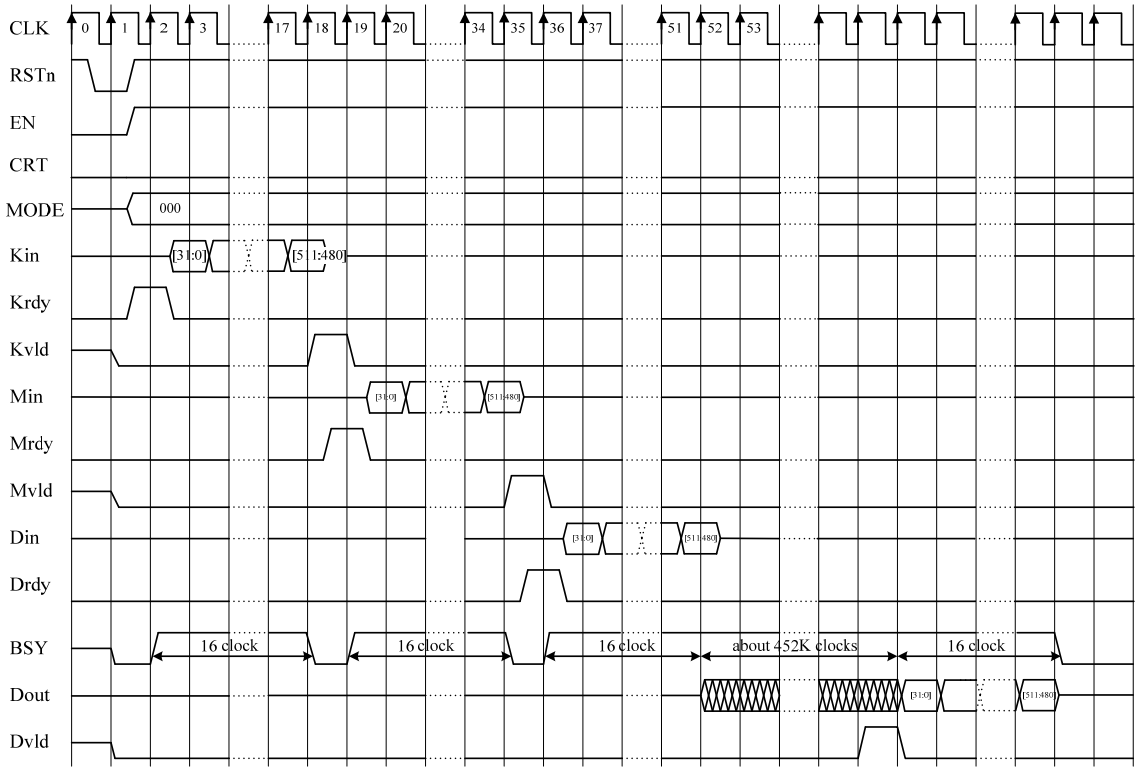


Figure 5.30 Timing Chart for RSA (non-CRT)

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 5.31 illustrates the timing for the RSA macro in the CRT mode. The timing is almost the same as in Figure 5.30 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.

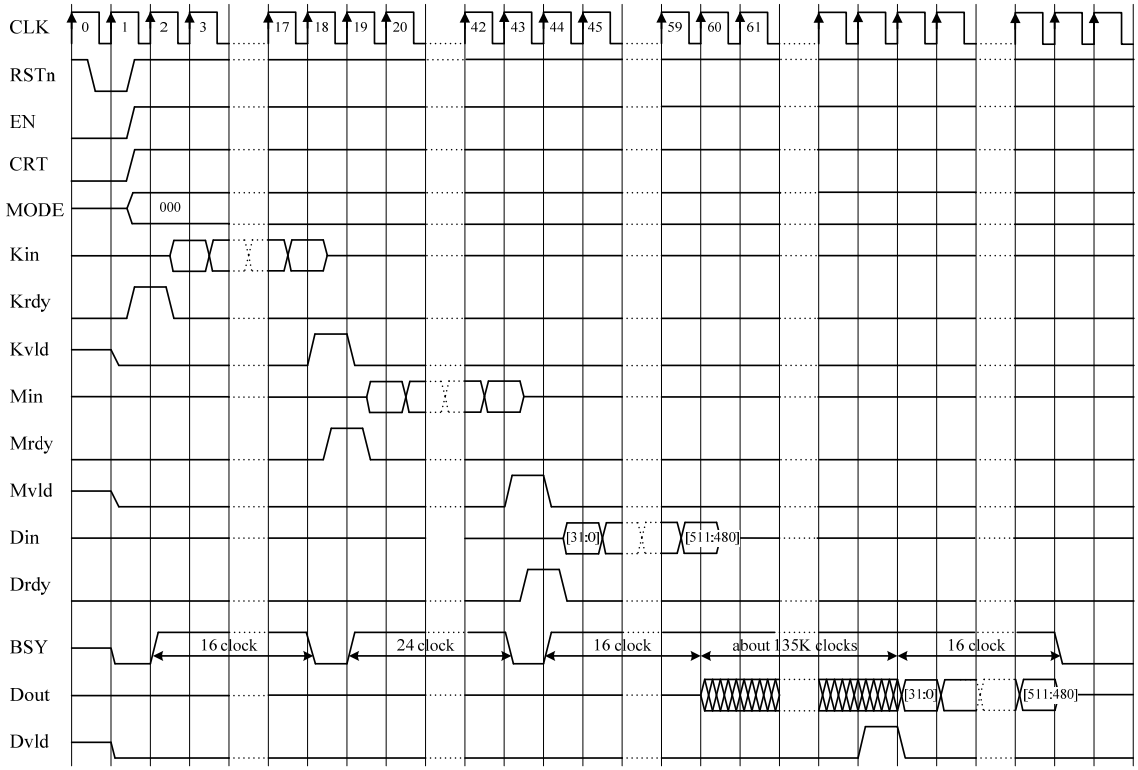


Figure 5.31 Timing Chart for RSA (CRT)

5.17 SEED

The overview specifications and I/O ports of the cryptographic circuit macro for SEED are shown in Table 5.25 and Table 5.26, respectively. SEED is a block cipher with the Feistel structure, proposed by KISA (Korea Information Security Agency).

Table 5.25 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 5.26 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 5.32 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.

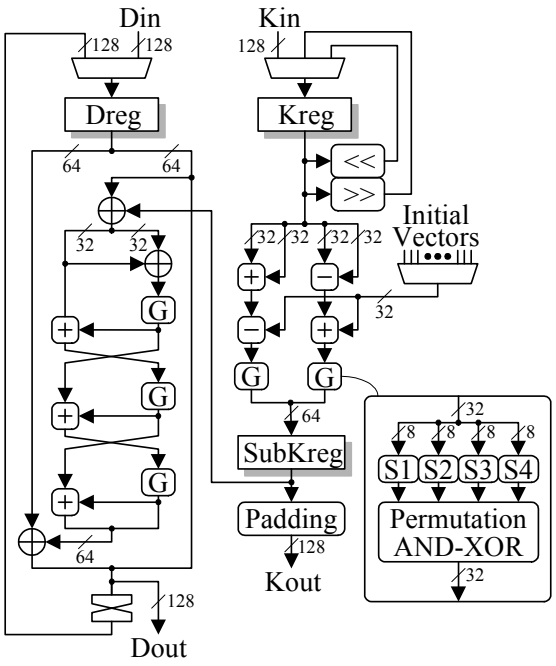


Figure 5.32 Datapath Architecture of SEED

Figure 5.33 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.

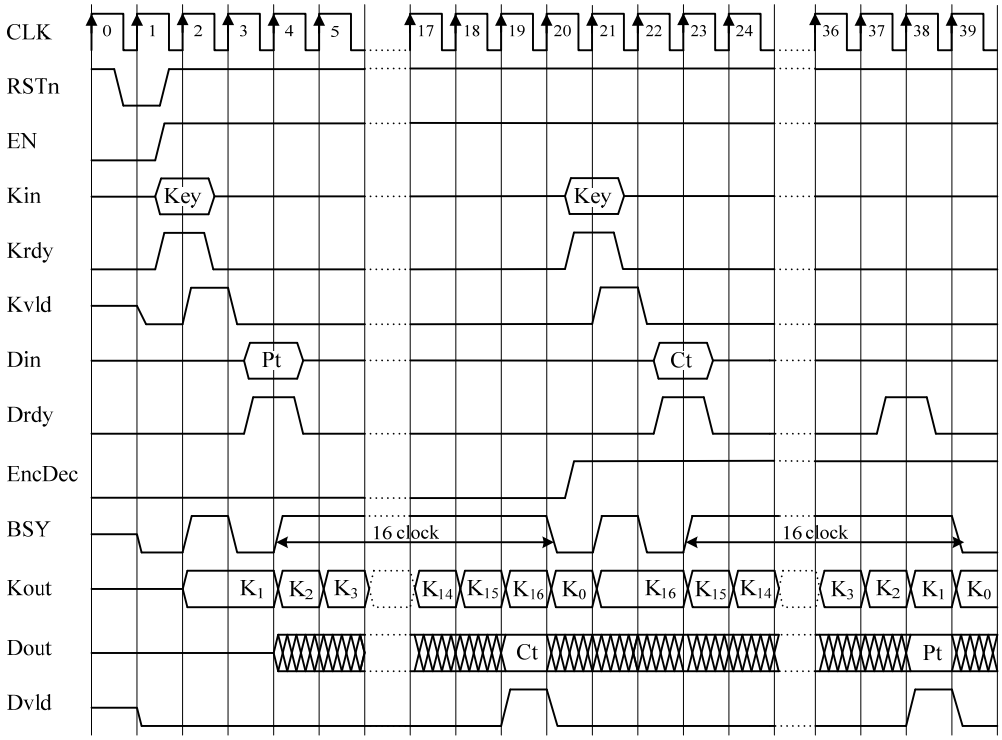


Figure 5.33 Timing Chart for SEED

5.18 TDES

The overview specifications and I/O ports of the cryptographic circuit macro for TDES (Triple-DES)[14] are shown in Table 5.27 and Table 5.28, respectively. TDES performs DES, the 56-bit-key block cipher, three times by changing the keys (3×16 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 clock 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 5.27 Overview Specifications of TDES

Algorithm	3-key Triple-DES
Data block length	64 bits
Key length	3 × 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 5.28 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 5.34 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.

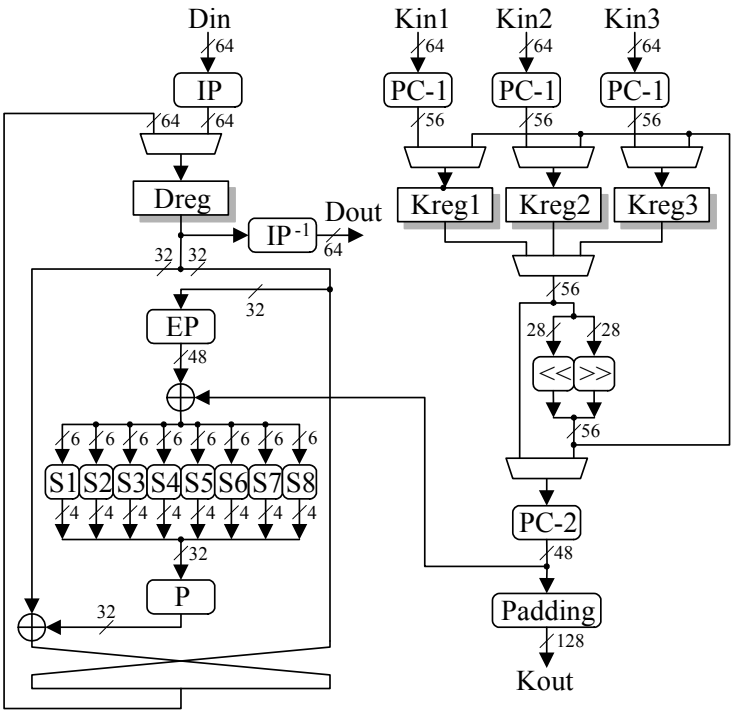


Figure 5.34 Datapath Architecture of TDES

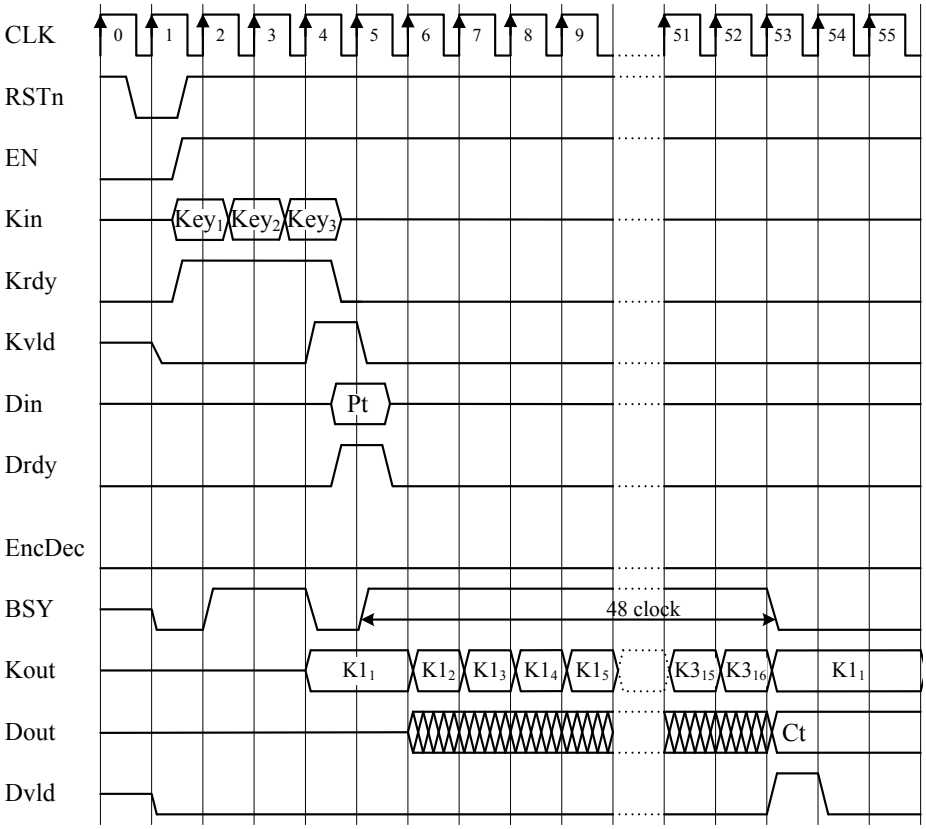


Figure 5.35 Timing Chart for TDES

Figure 5.35 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 Key_1~Key_3 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 $K1_1$ corresponding to the first secret key Key_1. 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.

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 Cryptotology (*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 Hardwar 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. L´opez, 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) M. Kne.zevi´c, K. Sakiyama, J. Fan, I. Verbauwhede, “Modular Multiplication in $GF(2^n)$ without Pre-computational Phase,” Proc. WAIFI’08, LNCS 5130, Springer-Verlag, pp. 77-87, 2008.
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. Adliman, “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

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)

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