HyRAL FPGA **設計仕様書**

株式会社ローレルインテリジェントシステムズ様	承認	
株式会社ケーアイテクノロジー	承認	

Knowledge Imaging Technology KIT

変更履歴

版数	日付	変更内容	作成者
初版	2009/12/ 13	初版作成	石井　均
2 版	2010/01/11	3. FPGA 入出力仕様変更 3.1. Const1, 2,3 値変更 3.3.ciphergen モジュール名称変更 3.3.6. ガロア体演算方法変更 3.4. Decrypt モジュール追記 4 章追加	石井　均
3 版	2010/01/26	表紙 社名間違いを訂正 1.概要 支給元名称を変更 ファイル名の誤植を修正	石井　均

目次

HyRAL FPGA 設計仕様書

1. 概要

本仕様は、HyRAL(Hybrid Randomize Algorithm)を FPGA に実装するための FPGA 設計仕様書です。
本 FPGA 設計仕様書は、CRYPTREC 電子政府推奨暗号リスト改訂のための暗号技術公募要項(2009 年度)が要求する事項についての各種仕様を規定するものである。

関連文書　HyRAL 最終仕様(株式会社ローレルインテリジェントシステムズ様支給)
HyRAL 暗号_20091214.xls
HyRAL 復号_20091214.xls
電子政府推奨暗号リスト改訂のための暗号技術公募要項(2009 年度) (CRYPTREC)

2. 開発環境

2.1. FPGA

ターゲットとする FPGA は、Xilinx 社製 XC5VLX50 もしくは XCVLX30 とする。

2.2. 開発言語

ハードウェア記述言語は、VerilogHDL を使用する。

2.3. 開発ソフトウェア

シミュレーターツールは、Mentor Graphics 社製 Model-Sim を使用する。
論理合成および配置配線ツールは、Xilinx 社製 ISE Foundation を使用する。

2.4. IP

IP(Intellectual Property)は、Xilinx 社が無償で提供する IP のみを使用し、ローヤリティが発生する IP は使用しないものとする。

2.5. 開発体系

本 FPGA の開発に関し、下記フォルダ構成とし管理を行う。

Root
- doc（各種文書を保存します。）
- ~~src（Verilog-HDLで記述したソースコードを保存します。）~~ *2 版
- testbench(シミュレーションに使用するテストベンチを保存します。)
- ISE(論理合成・配置配線用プロジェクトデータ)
- ModelSim(シミュレーションプロジェクトデータ)

3. FPGA 仕様 *2 版

本 FPGA は、128Bit の平文に対し暗号化および復号化を行う。
また、128Bit Single Key モードおよび 256Bit (192Bit) Double Key モード機能を有する。
概略ブロック図を図 3 に示す。

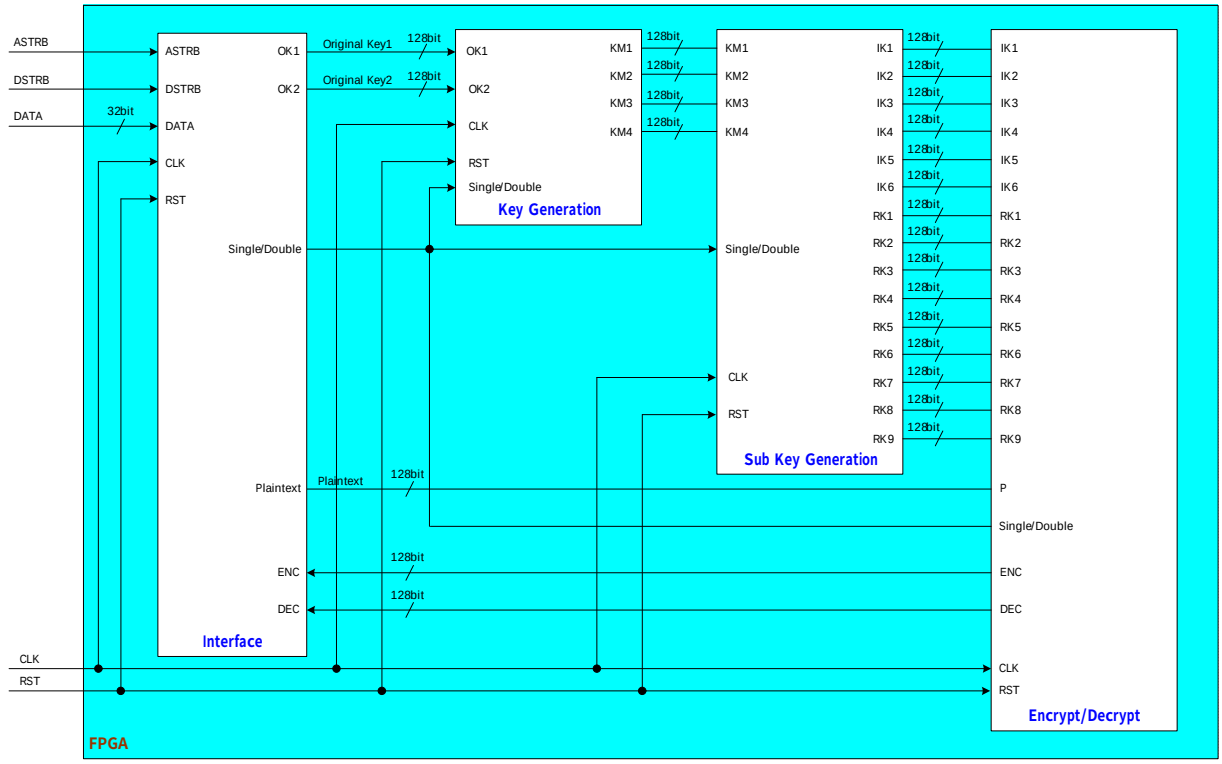


図 3 FPGA 入出力周辺暗号化概略ブロック図 *2 版

FPGA へ入出力は、システムクロックおよびシステムリセットの他、内部レジスタアクセス用のインターフェース信号および暗号化/復号化処理終了信号があります。4.1.を参照。

Intface モジュールは、外部インターフェースアクセス制御機能を持ち、すべての暗号化/復号化に必要な条件設定情報および暗号化/復号化結果を格納します。

Key Generation モジュールは Original Key1,2 から Single/Double Key モード別に Key Material 信号を生成します。さらに Sub Key Generation モジュールで IK,RK 信号を生成しコード生成し Encrypt/Decrypt モジュールに信号を渡します。*2 版

3.1. Key Generation モジュール

本モジュールは、Key Material を生成します。

本モジュールは図 3.1-1 に示す Base Key Generation モジュールを組み合わせて構成する。

図内 G1 Func.および G2 Func.については、後述する。

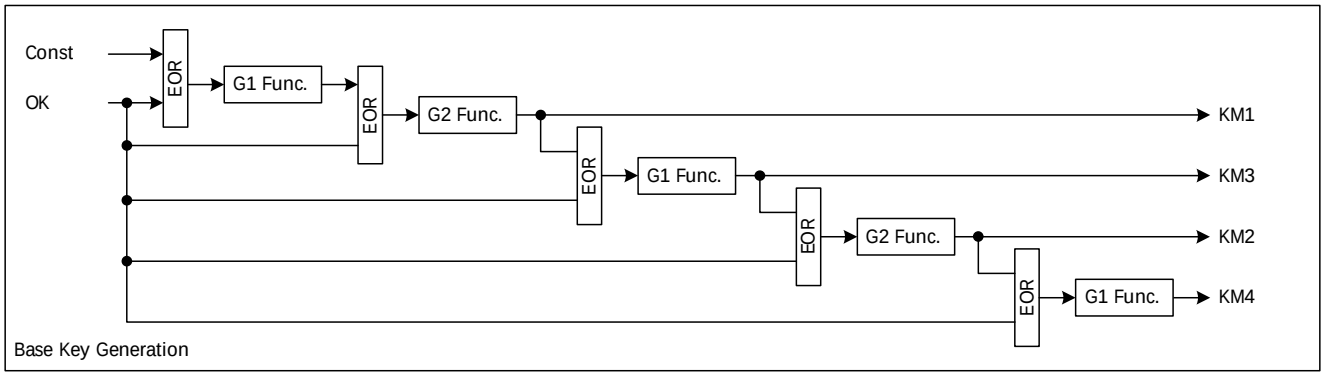
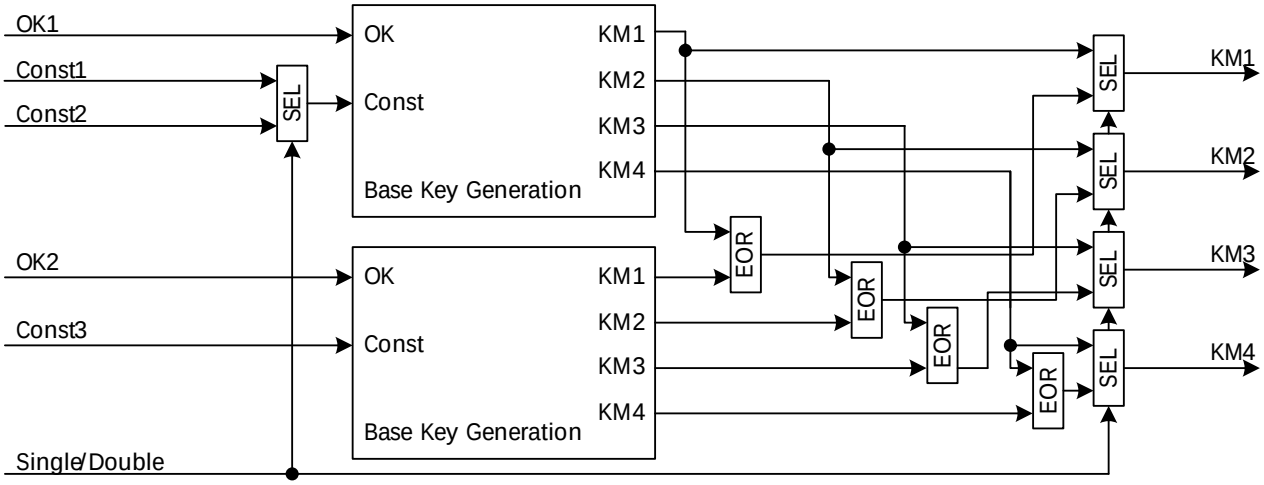


図 3.1-1 Base Key Generation 概略ブロック図

Key Generation モジュールは図 3.1-2 に示すように Base Key Generation の組み合わせで構成される。



Const1＝0x628C_CDA0_3B15_65C1_3BAD_2D4F_B880_6AC5　*2版
Const2＝0xF925_1A23_65CD_3C2E_8066_CBBB_FE31_6B7B　*2版
Const3＝0x5DE2_8625_656B_71FF_9FFB_1E12_EEF1_27F5　*2版

Single Key モードか Double Key モードかにより、KeyMaterial(KM)の値が異なる。

Single Key モードの場合、OK1 と Const1 を用いて生成した KM を用い、Double Key モードの場合

OK1 および OK2 と Const2 および Const3 を用いて生成した KM を各々EOR した結果を KM として用いる。

3.2. Sub Key Generation

本モジュールは、Sub Key を生成します。

Single Key モードと Double Key モードで生成方法が異なる。

KM の 1~4 は 128bit データであり、次のように表現すると定義する。

KMi[127:0] (i=1~4)

Sub Key は、KM をさらに 32Bit ずつに分解し EOR することで Sub Key を生成する。

以下に、Sub Key(IK(1~6),RK(1~8))の生成方法を示す。

3.2.1. Single Keyモード

Sub Key	算出方法	=	KM1	EOR	KM4
RK4	RK4[127:96]	=	KM1[127:96]	EOR	KM4[31:0]
	RK4[95:64]	=	KM1[95:64]	EOR	KM4[63:32]
	RK4[63:32]	=	KM1[63:32]	EOR	KM4[95:64]
	RK4[31:0]	=	KM1[31:0]	EOR	KM4[127:96]
IK1	IK1[127:96]	=	KM1[95:64]	EOR	KM4[127:96]
	IK1[95:64]	=	KM1[63:32]	EOR	KM4[31:0]
	IK1[63:32]	=	KM1[31:0]	EOR	KM4[63:32]
	IK1[31:0]	=	KM1[127:96]	EOR	KM4[95:64]
IK4	IK4[127:96]	=	KM1[63:32]	EOR	KM4[127:96]
	IK4[95:64]	=	KM1[31:0]	EOR	KM4[95:64]
	IK4[63:32]	=	KM1[127:96]	EOR	KM4[63:32]
	IK4[31:0]	=	KM1[95:64]	EOR	KM4[31:0]

Sub Key	算出方法	=	KM3	EOR	KM4
RK1	RK1[127:96]	=	KM3[31:0]	EOR	KM4[127:96]
	RK1[95:64]	=	KM3[63:32]	EOR	KM4[95:64]
	RK1[63:32]	=	KM3[95:64]	EOR	KM4[63:32]
	RK1[31:0]	=	KM3[127:96]	EOR	KM4[31:0]
RK6	RK6[127:96]	=	KM3[127:96]	EOR	KM4[95:64]
	RK6[95:64]	=	KM3[31:0]	EOR	KM4[63:32]
	RK6[63:32]	=	KM3[63:32]	EOR	KM4[31:0]
	RK6[31:0]	=	KM3[95:64]	EOR	KM4[127:96]
RK3	RK3[127:96]	=	KM3[127:96]	EOR	KM4[63:32]
	RK3[95:64]	=	KM3[95:64]	EOR	KM4[31:0]
	RK3[63:32]	=	KM3[63:32]	EOR	KM4[127:96]
	RK3[31:0]	=	KM3[31:0]	EOR	KM4[95:64]
IK2	IK2[127:96]	=	KM3[31:0]	EOR	KM4[31:0]
	IK2[95:64]	=	KM3[127:96]	EOR	KM4[127:96]
	IK2[63:32]	=	KM3[95:64]	EOR	KM4[95:64]
	IK2[31:0]	=	KM3[63:32]	EOR	KM4[63:32]

Sub Key	算出方法	=	KM1	EOR	KM2
RK7	RK7[127:96]	=	KM1[31:0]	EOR	KM2[127:96]
	RK7[95:64]	=	KM1[63:32]	EOR	KM2[95:64]
	RK7[63:32]	=	KM1[95:64]	EOR	KM2[63:32]
	RK7[31:0]	=	KM1[127:96]	EOR	KM2[31:0]
RK2	RK2[127:96]	=	KM1[127:96]	EOR	KM2[95:64]
	RK2[95:64]	=	KM1[31:0]	EOR	KM2[63:32]
	RK2[63:32]	=	KM1[63:32]	EOR	KM2[31:0]
	RK2[31:0]	=	KM1[95:64]	EOR	KM2[127:96]
RK5	RK5[127:96]	=	KM1[127:96]	EOR	KM2[63:32]
	RK5[95:64]	=	KM1[95:64]	EOR	KM2[31:0]
	RK5[63:32]	=	KM1[63:32]	EOR	KM2[127:96]
	RK5[31:0]	=	KM1[31:0]	EOR	KM2[95:64]
IK3	IK3[127:96]	=	KM1[31:0]	EOR	KM2[31:0]
	IK3[95:64]	=	KM1[127:96]	EOR	KM2[127:96]
	IK3[63:32]	=	KM1[95:64]	EOR	KM2[95:64]
	IK3[31:0]	=	KM1[63:32]	EOR	KM2[63:32]

3.2.2. Double Keyモード

Sub Key	算出方法	=	KM1	EOR	KM4
RK5	RK5[127:96]	=	KM1[127:96]	EOR	KM4[31:0]
	RK5[95:64]	=	KM1[95:64]	EOR	KM4[63:32]
	RK5[63:32]	=	KM1[63:32]	EOR	KM4[95:64]
	RK5[31:0]	=	KM1[31:0]	EOR	KM4[127:96]
IK1	IK1[127:96]	=	KM1[95:64]	EOR	KM4[127:96]
	IK1[95:64]	=	KM1[63:32]	EOR	KM4[31:0]
	IK1[63:32]	=	KM1[31:0]	EOR	KM4[63:32]
	IK1[31:0]	=	KM1[127:96]	EOR	KM4[95:64]
IK6	IK6[127:96]	=	KM1[63:32]	EOR	KM4[127:96]
	IK6[95:64]	=	KM1[31:0]	EOR	KM4[95:64]
	IK6[63:32]	=	KM1[127:96]	EOR	KM4[63:32]
	IK6[31:0]	=	KM1[95:64]	EOR	KM4[31:0]

Sub Key	算出方法	=	KM3	EOR	KM4
RK1	RK1[127:96]	=	KM3[31:0]	EOR	KM4[127:96]
	RK1[95:64]	=	KM3[63:32]	EOR	KM4[95:64]
	RK1[63:32]	=	KM3[95:64]	EOR	KM4[63:32]
	RK1[31:0]	=	KM3[127:96]	EOR	KM4[31:0]
RK8	RK8[127:96]	=	KM3[127:96]	EOR	KM4[95:64]
	RK8[95:64]	=	KM3[31:0]	EOR	KM4[63:32]
	RK8[63:32]	=	KM3[63:32]	EOR	KM4[31:0]
	RK8[31:0]	=	KM3[95:64]	EOR	KM4[127:96]
RK3	RK3[127:96]	=	KM3[127:96]	EOR	KM4[63:32]
	RK3[95:64]	=	KM3[95:64]	EOR	KM4[31:0]
	RK3[63:32]	=	KM3[63:32]	EOR	KM4[127:96]
	RK3[31:0]	=	KM3[31:0]	EOR	KM4[95:64]
RK4	RK4[127:96]	=	KM3[31:0]	EOR	KM4[31:0]
	RK4[95:64]	=	KM3[127:96]	EOR	KM4[127:96]
	RK4[63:32]	=	KM3[95:64]	EOR	KM4[95:64]
	RK4[31:0]	=	KM3[63:32]	EOR	KM4[63:32]

Sub Key	算出方法	=	KM1	EOR	KM2
RK9	RK9[127:96]	=	KM1[31:0]	EOR	KM2[127:96]
	RK9[95:64]	=	KM1[63:32]	EOR	KM2[95:64]
	RK9[63:32]	=	KM1[95:64]	EOR	KM2[63:32]
	RK9[31:0]	=	KM1[127:96]	EOR	KM2[31:0]
RK2	RK2[127:96]	=	KM1[127:96]	EOR	KM2[95:64]
	RK2[95:64]	=	KM1[31:0]	EOR	KM2[63:32]
	RK2[63:32]	=	KM1[63:32]	EOR	KM2[31:0]
	RK2[31:0]	=	KM1[95:64]	EOR	KM2[127:96]
RK7	RK7[127:96]	=	KM1[127:96]	EOR	KM2[63:32]
	RK7[95:64]	=	KM1[95:64]	EOR	KM2[31:0]
	RK7[63:32]	=	KM1[63:32]	EOR	KM2[127:96]
	RK7[31:0]	=	KM1[31:0]	EOR	KM2[95:64]
RK6	RK6[127:96]	=	KM1[31:0]	EOR	KM2[31:0]
	RK6[95:64]	=	KM1[127:96]	EOR	KM2[127:96]
	RK6[63:32]	=	KM1[95:64]	EOR	KM2[95:64]
	RK6[31:0]	=	KM1[63:32]	EOR	KM2[63:32]

Sub Key	算出方法	=	KM2	EOR	KM3
IK2	IK2[127:96]	=	KM2[31:0]	EOR	KM3[127:96]
	IK2[95:64]	=	KM2[63:32]	EOR	KM3[95:64]
	IK2[63:32]	=	KM2[95:64]	EOR	KM3[63:32]
	IK2[31:0]	=	KM2[127:96]	EOR	KM3[31:0]
IK3	IK3[127:96]	=	KM2[127:96]	EOR	KM3[95:64]
	IK3[95:64]	=	KM2[31:0]	EOR	KM3[63:32]
	IK3[63:32]	=	KM2[63:32]	EOR	KM3[31:0]
	IK3[31:0]	=	KM2[95:64]	EOR	KM3[127:96]
IK4	IK4[127:96]	=	KM2[127:96]	EOR	KM3[63:32]
	IK4[95:64]	=	KM2[95:64]	EOR	KM3[31:0]
	IK4[63:32]	=	KM2[63:32]	EOR	KM3[127:96]
	IK4[31:0]	=	KM2[31:0]	EOR	KM3[95:64]
IK5	IK5[127:96]	=	KM2[31:0]	EOR	KM3[31:0]
	IK5[95:64]	=	KM2[127:96]	EOR	KM3[127:96]
	IK5[63:32]	=	KM2[95:64]	EOR	KM3[95:64]
	IK5[31:0]	=	KM2[63:32]	EOR	KM3[63:32]

3.3. Encrypt モジュール *2 版

Encrypt モジュールは、3.2 で述べた Sub Key を用い、暗号化を行うブロックである。

図 3.3-1 に暗号化フローを示す。

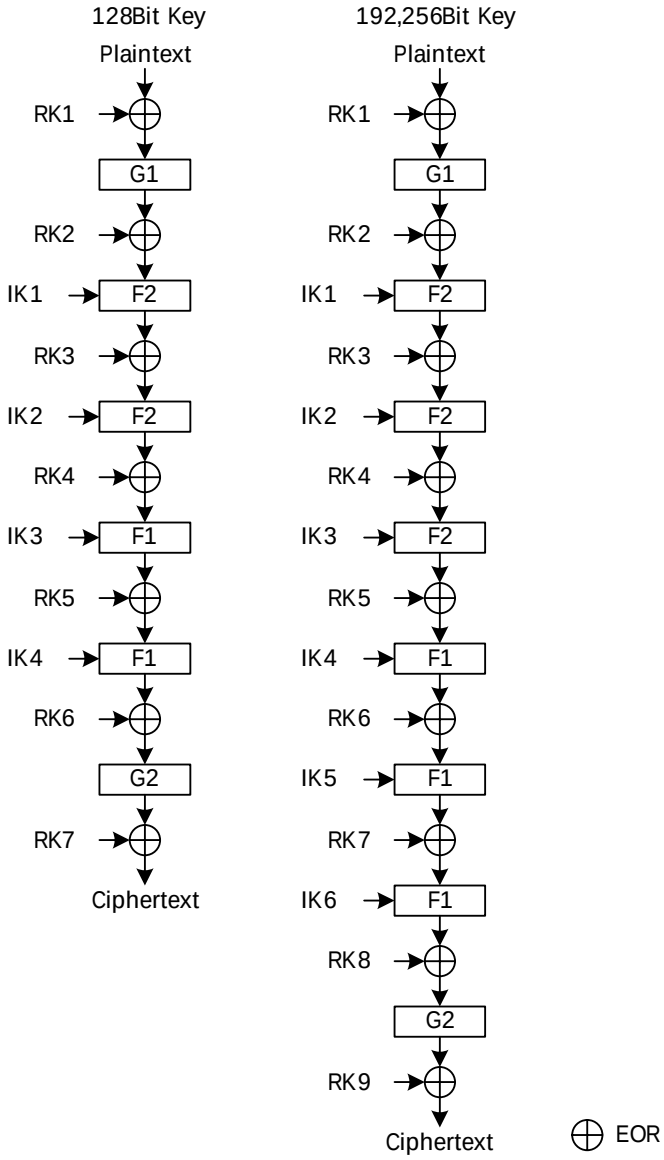


図 3.3-1 暗号化フロー

図 3.3-1 内の F1,F2 および G1,G2 は各々F 関数、G 関数と呼ばれる暗号化を行うモジュールである。

3.3.1. F関数

F 関数は、IK を引数に持つ暗号化を行う関数です。

F 関数は F1 および F2 の 2 種類を持つ。図 3.3.1-1 に各 F 関数の詳細を示す。

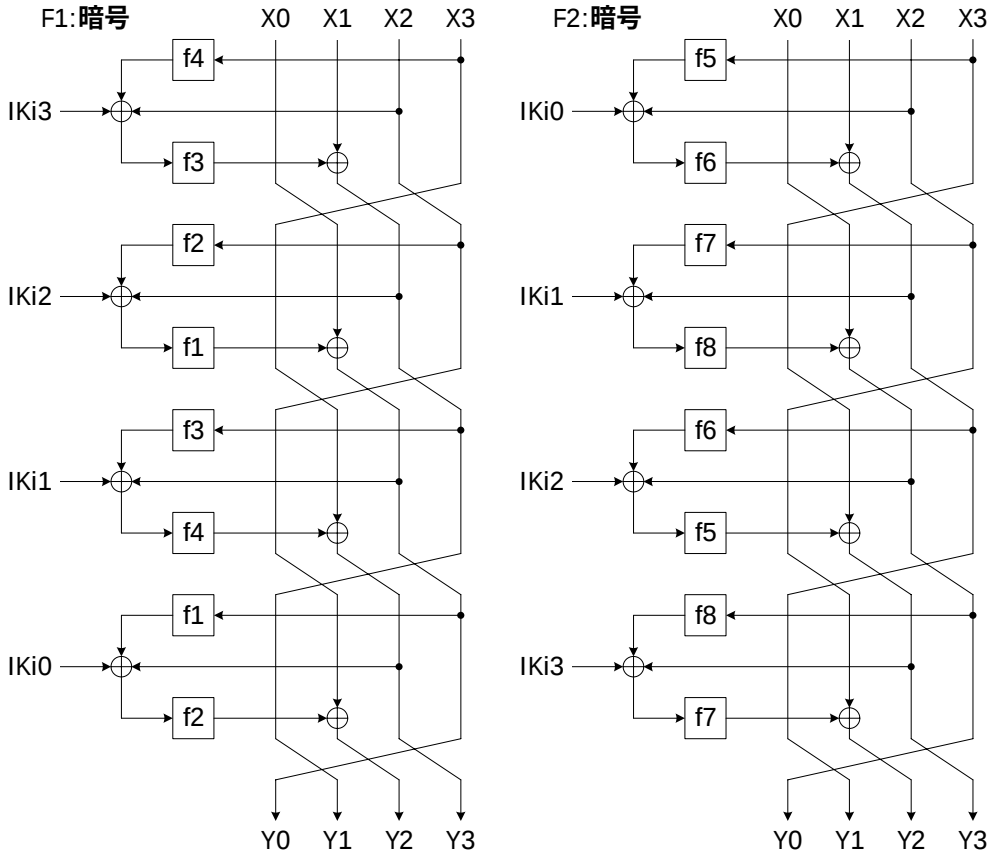


図 3.3.1-1　F 関数詳細

図 3.3.1-1 内の X0~3 は入力する 128bit データを意味し、X0=X[127:96], X1=X[95:64], X2=X[63:32], X3=X[31:0]を意味する。同様に、IKi0=IKi[127:96], IKi1= IKi[95:64], IKi2= IKi[63:32], IKi3= IKi[31:0]を意味する。Y0~3 は Y0=Y[127:96], Y1=Y[95:64], Y2=Y[63:32], Y3=Y[31:0]を意味し、F 関数の出力である。f1~8 については後述する。

3.3.2. G関数

G 関数は、暗号化を行う関数で、G1 および G2 の 2 種類の関数を持つ。

図 3.3.2-1 に各 G 関数の詳細を示す。

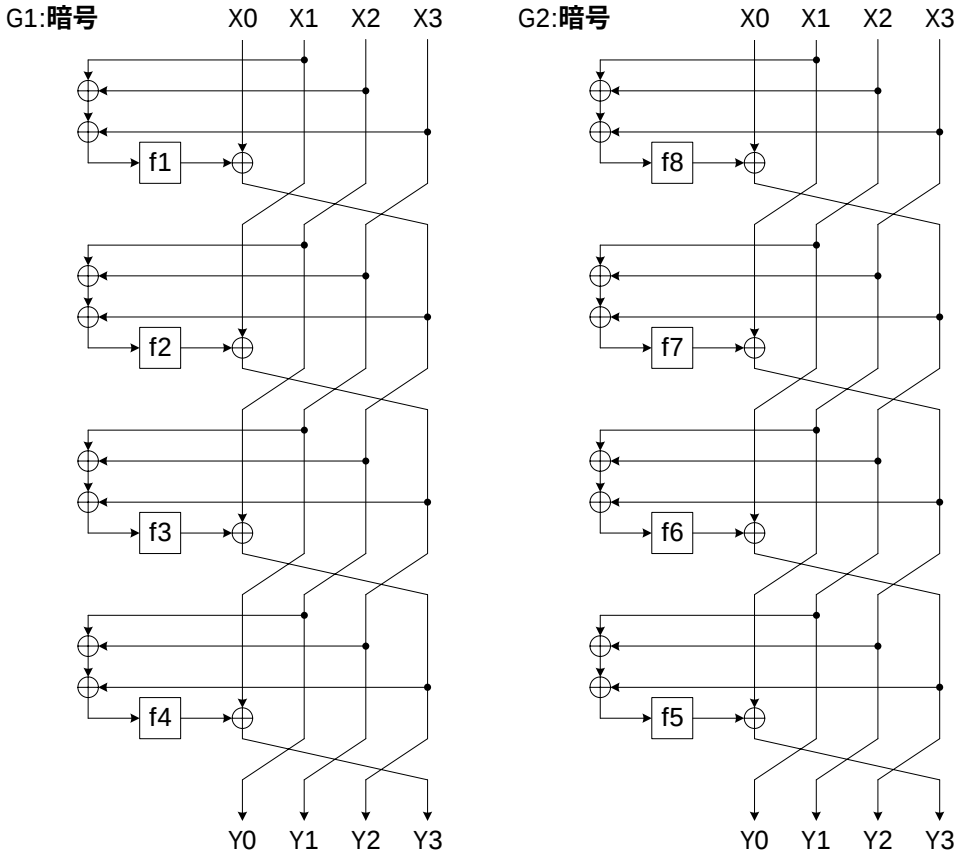


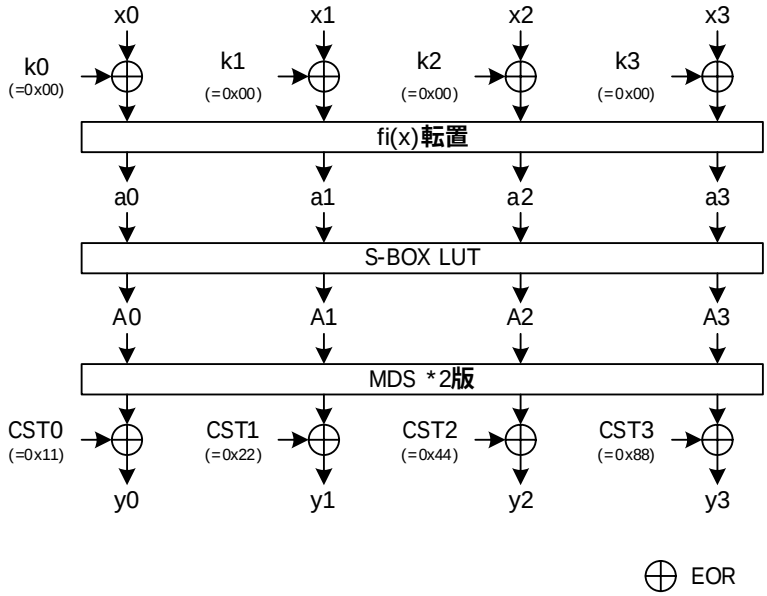
図 3.3.2-1　G 関数詳細

図 3.3.2-1 内の X0~3 は入力する 128bit データを意味し、X0=X[127:96], X1=X[95:64], X2=X[63:32], X3=X[31:0]を意味する。Y0~3 は Y0=Y[127:96], Y1=Y[95:64], Y2=Y[63:32], Y3=Y[31:0]を意味し、G 関数の出力である。f1~8 については後述する。

3.3.3. f関数

f 関数は、32bit 入力と 32bit 出力のデータ変換モジュールです。

図 3.3.3-1 にf関数の概略変換フローを示す。Fi(x)転置は 8 種類あり、f1~8 と定義する。



3.3.4. fi(x)転置

fi(x)転置は、8 種類あり f1~8 と定義する。

各転置は以下のように行われる。

f1(x)=(x0, x1, x2, x3)　　f5(x)=(x3, x2, x1, x0)

f2(x)=(x1, x2, x3, x0)　　f6(x)=(x2, x1, x0, x3)

f3(x)=(x2, x3, x0, x1)　　f7(x)=(x1, x0, x3, x2)

f4(x)=(x3, x0, x1, x2)　　f8(x)=(x0, x3, x2, x1)

3.3.5. S-box LUT

S-box LookUpTable を図 3.3.5-1 に示す。

上位＼下位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	5e	d3	af	36	43	a6	49	33	93	3b	21	91	df	47	f4
1	b6	70	06	d0	81	82	fa	a1	10	b5	3c	ba	97	85	b7	79
2	ed	5c	ca	05	87	bf	24	4c	51	ec	17	61	22	f0	3e	18
3	a7	64	13	ab	e9	09	25	54	2d	31	69	f5	37	67	fe	1d
4	0b	28	a3	2f	e4	0f	d4	da	1b	fc	e6	ac	53	04	27	a9
5	94	8b	d5	c4	90	6b	f8	9d	c5	db	ea	e2	ae	63	07	7a
6	5b	23	34	38	03	8c	46	68	cd	1a	1c	41	7d	a0	9c	dd
7	08	4e	e3	d7	1e	b3	50	5d	c6	0e	ad	cf	d6	eb	0d	b1
8	fb	7c	c3	2e	65	48	b8	8f	ce	e7	62	d2	12	4a	c8	26
9	a5	8e	3d	76	86	57	bc	bd	11	75	71	78	1f	ef	e0	0c
A	de	6a	6d	32	84	72	8a	d8	f9	dc	9a	89	9f	88	14	2a
B	9b	9e	d9	95	b9	a4	02	f7	96	73	56	be	7f	80	7e	83
C	00	01	f6	8d	7b	d1	52	cb	b0	e1	c7	e5	29	c0	4f	e8
D	58	3f	cc	fd	ee	b2	40	ff	99	2b	5f	60	aa	4b	b4	74
E	2c	45	6c	92	66	42	39	f3	77	bb	19	59	20	6f	35	f2
F	c1	0a	15	98	a2	c2	44	30	55	4d	c9	a8	5a	f1	6e	3a

入力する a0~a3 の上位バイト、下位バイトを用いテーブルを引く。

例） a0 = 0x16 ⇒ A0 = 0xfa

3.3.6. MDS ~~LUT~~*2 およびCST演算部

MDS LUT および CSTi 演算は、行列を用い下記のように表記できる。

$$\begin{pmatrix} y0 \\ y1 \\ y2 \\ y3 \end{pmatrix} = \begin{pmatrix} 0x03 & 0x03 & 0x02 & 0x01 \\ 0x01 & 0x02 & 0x02 & 0x02 \\ 0x07 & 0x03 & 0x01 & 0x02 \\ 0x07 & 0x04 & 0x05 & 0x03 \end{pmatrix} \begin{pmatrix} A0 \\ A1 \\ A2 \\ A3 \end{pmatrix} \oplus \begin{pmatrix} 0x11 \\ 0x22 \\ 0x44 \\ 0x88 \end{pmatrix}$$

本行列は、ガロア体上での演算であり、下記のように展開できる。

y0 = 0x03・A0 ⊕ 0x03・A1 ⊕ 0x02・A2 ⊕ 0x01・A3 ⊕ 0x11

y1 = 0x01・A0 ⊕ 0x02・A1 ⊕ 0x02・A2 ⊕ 0x02・A3 ⊕ 0x22

y2 = 0x07・A0 ⊕ 0x03・A1 ⊕ 0x01・A2 ⊕ 0x02・A3 ⊕ 0x44

y3 = 0x07・A0 ⊕ 0x04・A1 ⊕ 0x05・A2 ⊕ 0x03・A3 ⊕ 0x88

また、0X03・A0 の・はGF(2^8)の乗算であり、通常の乗算ではないため 10 進数計算では求まらない。

FPGAでは~~倍数テーブルを用意し~~、上記固定行列でのガロア体演算を行う。 *2 版

3.4. Decrypt モジュール *2 版

Decrypt モジュールは、3.2 で述べた Sub Key を用い、復号化を行うブロックである。

図 3.4-1 に復号化フローを示す。

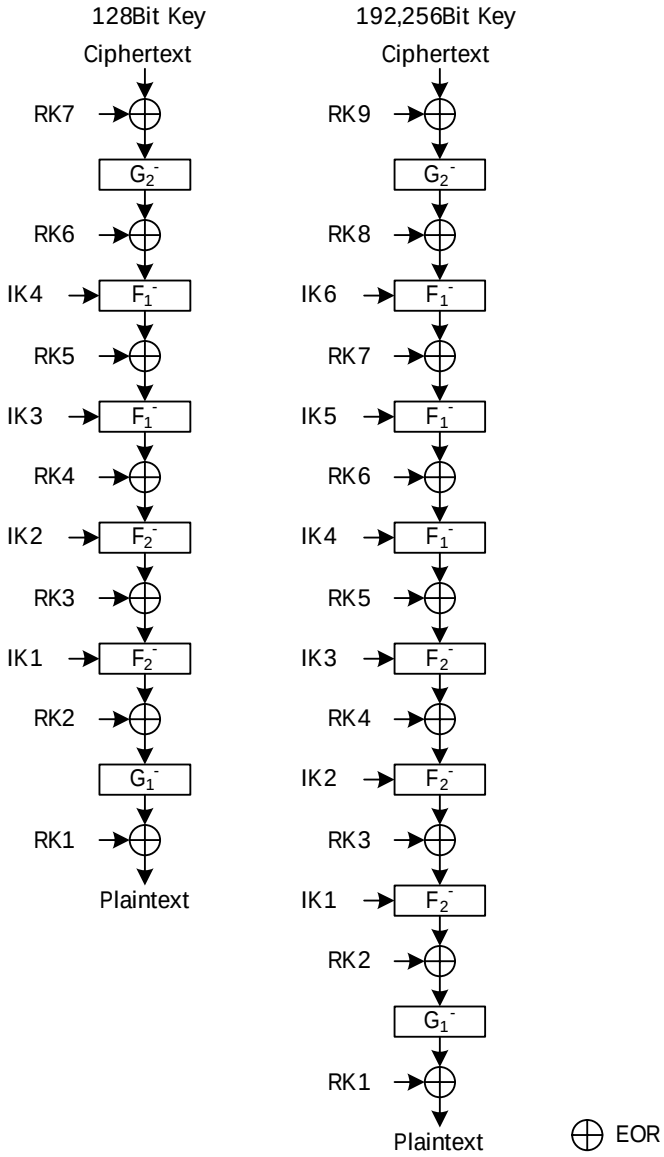


図 3.4-1 復号化フロー

図 3.4-1 内のF_1^-,F_2^-およびG_1^-,G_2^-は各々F^-関数、G^-関数と呼ばれる復号化を行うモジュールである。

3.4.1. F⁻関数 *2 版

F⁻関数は、IKを引数に持つ復号化を行う関数です。

F⁻関数はF_1^-およびF_2^-の 2 種類を持つ。図 3.4.1-1 に各F⁻関数の詳細を示す。

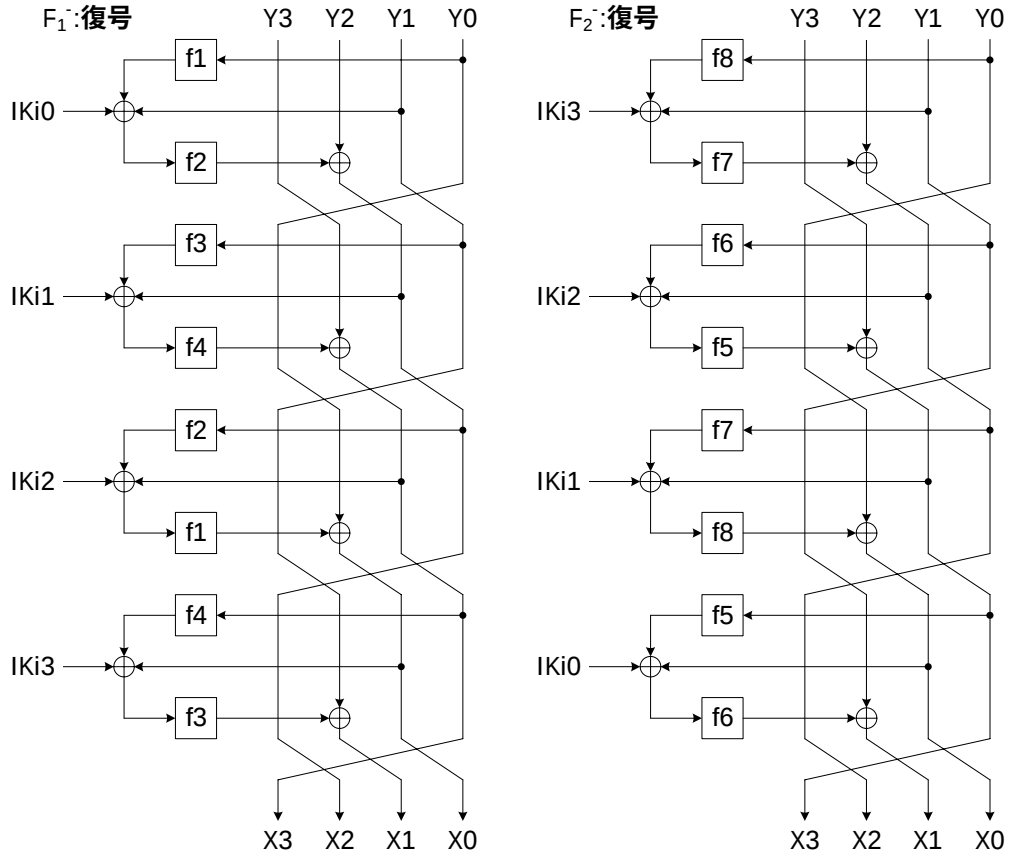


図 3.4.1-1　F 関数詳細

図 3.4.1-1 内のY0~3 は入力する 128bitデータを意味し、Y0=Y[127:96], Y1=Y[95:64], Y2=Y[63:32], Y3=Y[31:0]を意味する。同様に、IKi0=IKi[127:96], IKi1= IKi[95:64], IKi2= IKi[63:32], IKi3= IKi[31:0]を意味する。X0~3 はX0=X[127:96], X1=X[95:64], X2=X[63:32], X3=X[31:0]を意味し、F⁻関数の出力である。

f1~8 については 3.3.5.参照。

3.4.2. G-関数 *2 版

G-関数は、復号化を行う関数で、G_1-およびG_2-の 2 種類の関数を持つ。

図 3.4.2-1 に各G-関数の詳細を示す。

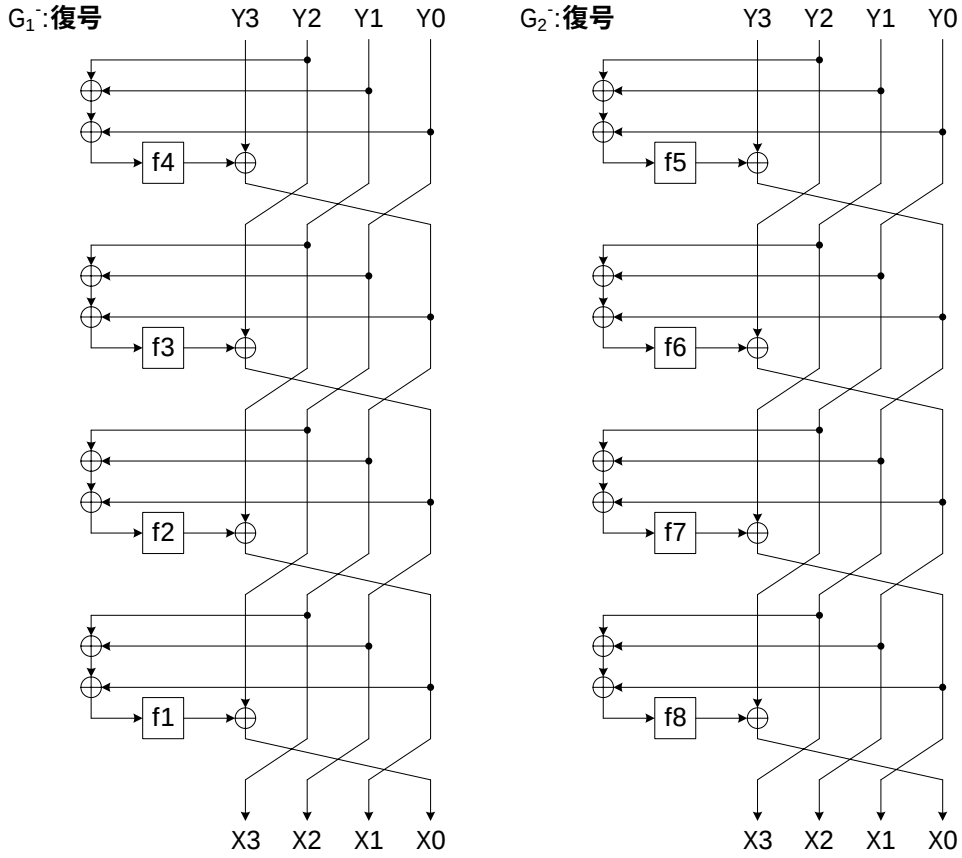


図 3.4.2-1　G 関数詳細

図 3.4.2-1 内のX0~3 は入力する 128bitデータを意味し、Y0=Y[127:96], Y1=Y[95:64], Y2=Y[63:32], Y3=Y[31:0]を意味する。Y0~3 はX0=X[127:96], X1=X[95:64], X2=X[63:32], X3=X[31:0]Y0=Y[127:96], Y1=Y[95:64], Y2=Y[63:32], Y3=Y[31:0]を意味し、G-関数の出力である。

f1~8 については 3.3.5.参照。

4. FPGA 詳細設計仕様

本 FPGA は、32bit データバスおよび 2bit の制御信号を用いたインターフェースを用い、128bit の平文入力と 128Bit または 256bit(192bit)KEY を設定することで暗号化および復号化を行う機能をもつ。
モジュール構成は HyRAL128_main をトップモジュールとし、以下のようになっている。

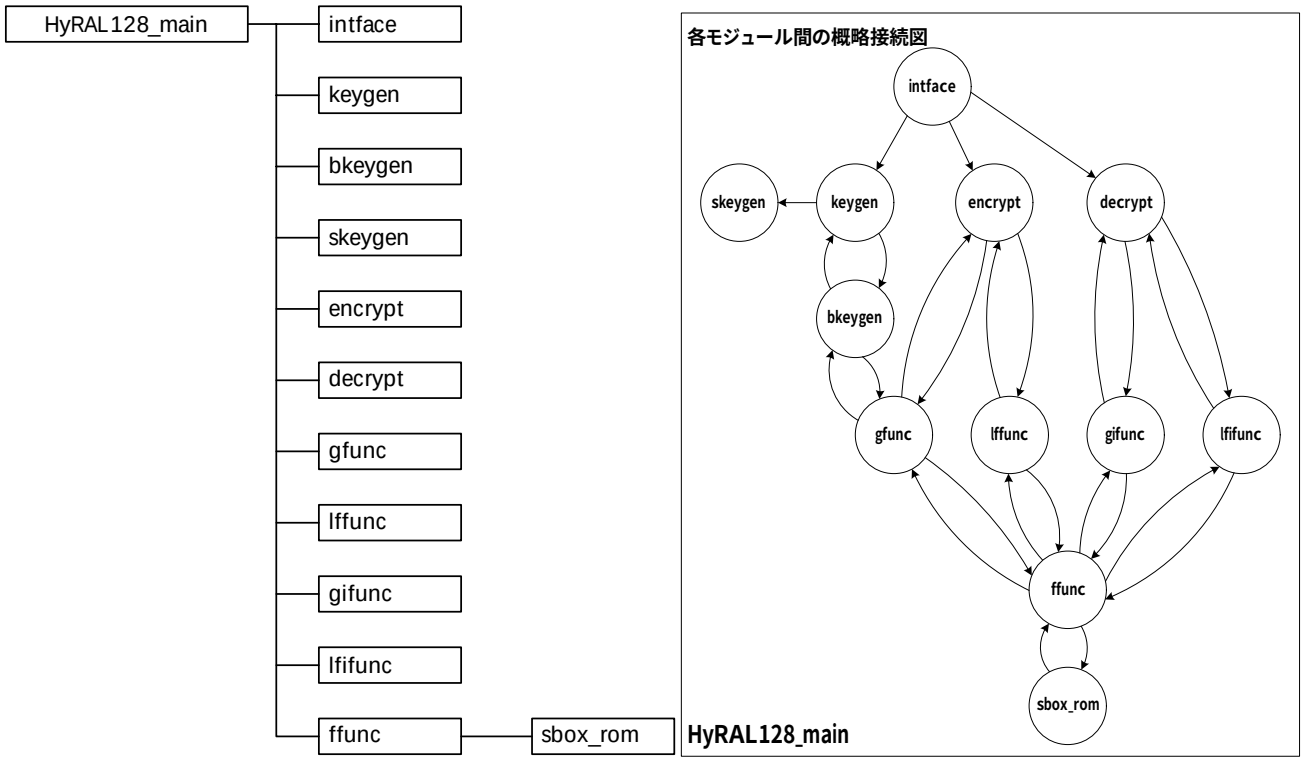


図 4 モジュール構成

極力小面積で FPGA に実装するため、共通利用が可能なモジュール(G 関数、F 関数、G-関数、F-関数、f 関数)を独立させ、各制御モジュール(bkeygen, encrypt,decrypt)から制御信号を発行する形をとり回路実装する。

以下に各モジュールの概略機能を示す。

モジュール	概要
HyRAL128_main	本 FPGA のトップファイルである。
intface	インターフェースモジュールで、各種レジスタ値の設定および処理開始信号発行などを行う。
keygen	Key Material を生成するブロックである。 生成の基本となる bkeygen モジュールを利用し KM1~4 を生成する。 生成の際 Single Key Mode か Double Key Mode かの選択を行うことで Mode にあった Key を生成する。
bkeygen	base key generator を意味し、Key 生成の冗長な部分をモジュール化したものである。
skeygen	Sub key generator を意味し、KM1~4 を利用し IK,RK を生成する。 Single Key Mode か Double Key Mode かで生成する IK,RK の種類を切り替えできる。
encrypt	暗号化の Main モジュールである。 G 関数(gfunc)、F 関数(lffunc)を制御し暗号化を行う。 Single Key Mode か Double Key Mode かで暗号化処理を切り替えできる。
decrypt	復号化の Main モジュールである。 G^{-}関数(gifunc)、F^{-}関数(lfifunc)を制御し暗号化を行う。 Single Key Mode か Double Key Mode かで暗号化処理を切り替えできる。
gfunc	G 関数本体であり、f 関数(ffunc)の制御を行う。
lffunc	F 関数本体であり、f 関数(ffunc)の制御を行う。
gifunc	G^{-}関数本体であり、f関数(ffunc)の制御を行う。
lfifunc	F^{-}関数本体であり、f関数(ffunc)の制御を行う。
ffunc	転置、SBOX 変換、MDS 演算を行うモジュールである。
sbox_rom	変換テーブル用 ROM である。

4.1. インターフェース

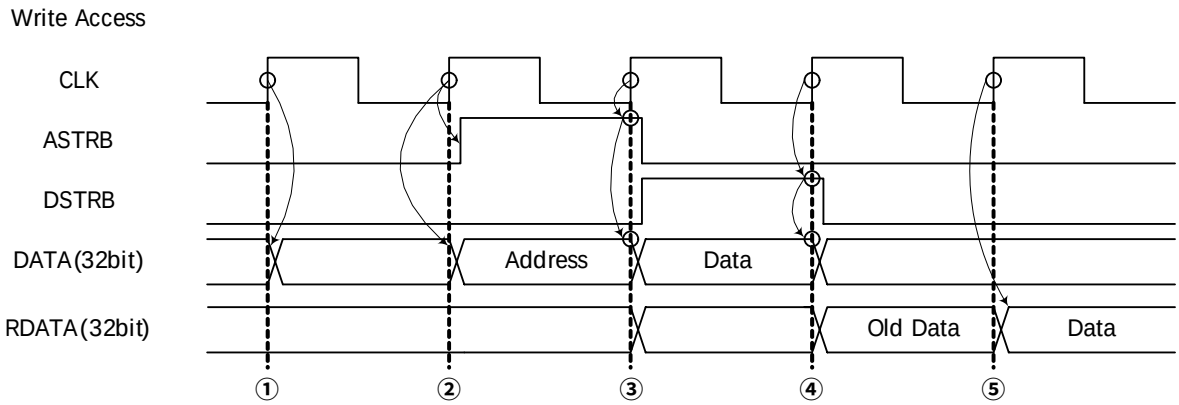
FPGA のピンアサインは以下表のとおりとなっています。

信号	ピン数	入出力	意味
CLK	1	入力	システムクロック
RST	1	入力	システムリセット
ASTRB	1	入力	アドレスストローブ。ASTRB="1"のときの DATA をアドレスとして取り込みます。
DSTRB	1	入力	データストローブ。DSTRB="1"のときの DATA をデータとして取り込みます。
DATA	32	入力	データバス
RDATA	32	出力	リード用データバス
KEYEND	1	出力	KEY Generation の終了信号
EEND	1	出力	暗号化の終了信号
DEND	1	出力	復号化の終了信号

暗号文や復号された平文はレジスタをリードすることで確認できます。
また、暗号化および復号化が終了したことは、EEND および DEND で知ることができます。

4.1.1. ライトアクセス方法

本 FPGA 内へのライトアクセスは下記に示すタイミングチャートによる。



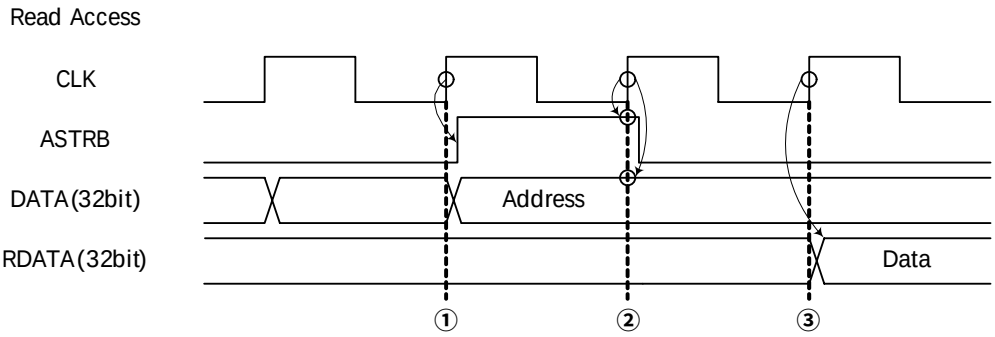
① すべての入力する信号は CLK 同期です。
② ASTRB(Address STRoBe)がアサートし、DATA バスに Adderss をロードします。
③ CLK 立ち上がりタイミングで ASTRB="1"のときに DATA バスの値を Address として FPGA は取り込みます。
④ CLK 立ち上がりタイミングで DSTRB(Data STRoBe)="1"のときに DATA バスの値を Data として FPGA は取り込みます。このとき、RDATA には、前回指定されたアドレスに格納されていたデータが出力されます。

⑤ CLK 立ち上がりタイミングで RDATA にライト後の新データを出力します。

※ ASTRB と DSTRB を同タイミングでアサートおよびネゲートした場合は、DATA バスの値を Address および Data として取り込みますので、注意が必要です。

4.1.2. リードアクセス方法

本 FPGA 内へのリードアクセスは下記に示すタイミングチャートによる。



① すべての入力する信号は CLK 同期です。ASTRB(Address STRoBe)がアサートし、DATA バスに Adderss をロードします。

② CLK 立ち上がりタイミングで ASTRB="1"のときに DATA バスの値を Address として FPGA は取り込みます。このとき、RDATA には、指定するアドレスに格納されていたデータが出力されます。

③ CLK 立ち上がりタイミングで RDATA に指定されたアドレスのデータを出力します。

※ DSTRB を発行するとライト動作を行いますので、注意が必要です。

4.1.3. アドレスマップ

本 FPGA のアドレスマップを下記表に示します。

バイトアクセスはできず、常に 4 バイト分のライトとなります。

アドレス	初期値	ニーモニック	概要
0x00000000	-	Bit 毎に定義	各種スタート信号
0x00000004	0x00000000	Bit 毎に定義	制御レジスタ
0x00000008	0x00000000	PT[31:0]	平文 0~31bit
0x0000000C	0x00000000	PT[63:32]	平文 32~63bit
0x00000010	0x00000000	PT[95:64]	平文 64~95bit
0x00000014	0x00000000	PT[127:96]	平文 96~127bit
0x00000018	0x00000000	OK1[31:0]	Original Key1 0~31bit
0x0000001C	0x00000000	OK1[63:32]	Original Key1 32~63bit
0x00000020	0x00000000	OK1[95:64]	Original Key1 64~95bit
0x00000024	0x00000000	OK1[127:96]	Original Key1 127~96bit
0x00000028	0x00000000	OK2[31:0]	Original Key2 0~31bit
0x0000002C	0x00000000	OK2[63:32]	Original Key2 32~63bit
0x00000030	0x00000000	OK2[95:64]	Original Key2 64~95bit
0x00000034	0x00000000	OK2[127:96]	Original Key2 127~96bit
0x00000038	0x00000000	-	Reserve
0x0000003C	0x00000000	-	Reserve
0x00000040	0xB8806AC5	Y2[31:0]	Y2 0~31bit
0x00000044	0x3BAD2D4F	Y2[63:32]	Y2 32~63bit
0x00000048	0x3B1565C1	Y2[95:64]	Y2 64~95bit
0x0000004C	0x628CCDA0	Y2[127:96]	Y2 96~127bit
0x00000050	0xFE316B7B	Z2[31:0]	Z2 0~31bit
0x00000054	0x8066CBBB	Z2[63:32]	Z2 32~63bit
0x00000058	0x65CD3C2E	Z2[95:64]	Z2 64~95bit
0x0000005C	0xF9251A23	Z2[127:96]	Z2 96~127bit
0x00000060	0xEEF127F5	W2[31:0]	W2 0~31bit
0x00000064	0x9FFB1E12	W2[63:32]	W2 32~63bit
0x00000068	0x656B71FF	W2[95:64]	W2 64~95bit
0x0000006C	0x5DE28625	W2[127:96]	W2 96~127bit
0x00000070	-	ENC[31:0]	暗号化データ 0~31bit
0x00000074	-	ENC[63:32]	暗号化データ 32~63bit
0x00000078	-	ENC[95:64]	暗号化データ 64~95bit

0x0000007C	-	ENC[127:96]	**暗号化データ** 96~127bit
0x00000080	-	DEC[31:0]	**復号化データ** 0~31bit
0x00000084	-	DEC[63:32]	**復号化データ** 32~63bit
0x00000088	-	DEC[95:64]	**復号化データ** 64~95bit
0x0000008C	-	DEC[127:96]	**復号化データ** 96~127bit

4.1.4. レジスタ詳細

アドレス 0x00000000

D31	D30	D29	D28	D27	D26	D25	D24
-	-	-	-	-	-	-	-
D23	D22	D21	D20	D19	D18	D17	D16
-	-	-	-	-	-	-	-
D15	D14	D13	D12	D11	D10	D9	D8
-	-	-	-	-	-	-	-
D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	CSTART	BSTART

ビット	初期値	ニーモニック	意味
D0	-	BSTART	Sub Key **生成スタート信号。**” 1”**ライトをすると処理開始する。処理が終了すると** KEYEND **ピンから** 1CLK **幅の**”1”**が出力される。**
D1	-	CSTART	**暗号化処理スタート信号。**”1”**ライトをすると処理開始する。処理が終了すると** EEND **ピンから** 1CLK **幅の**”1”**が出力される。**
D2	-	DSTART	**復号化処理スタート信号。**”1”**ライトをすると処理開始する。処理が終了すると** DEND **ピンから** 1CLK **幅の**”1”**が出力される。**
D31-D3			

アドレス 0x00000004

D31	D30	D29	D28	D27	D26	D25	D24
-	-	-	-	-	-	-	-

D23	D22	D21	D20	D19	D18	D17	D16
-	-	-	-	-	-	-	-

D15	D14	D13	D12	D11	D10	D9	D8
-	-	-	-	-	-	-	-

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	-	SDSW

ビット	初期値	ニーモニック	意味
D0	0x0	SDSW	Single/Double Key モード選択。 “0”：Single Key モード ”1”：Double Key モード
D31-D1	-	-	Reserve

アドレス 0x00000008

D31	D30	D29	D28	D27	D26	D25	D24
PT31	PT30	PT29	PT28	PT27	PT26	PT25	PT24
D23	D22	D21	D20	D19	D18	D17	D16
PT23	PT22	PT21	PT20	PT19	PT18	PT17	PT16
D15	D14	D13	D12	D11	D10	D9	D8
PT15	PT14	PT13	PT12	PT11	PT10	PT9	PT8
D7	D6	D5	D4	D3	D2	D1	D0
PT7	PT6	PT5	PT4	PT3	PT2	PT1	PT0

ビット	初期値	ニーモニック	意味
D31-D0	0x0	PT[31:0]	**暗号処理:平文** 0~31 **ビット設定** **復号処理:暗号文** 0~31 **ビット設定**

アドレス 0x0000000C

D31	D30	D29	D28	D27	D26	D25	D24
PT63	PT62	PT61	PT60	PT59	PT58	PT57	PT56
D23	D22	D21	D20	D19	D18	D17	D16
PT55	PT54	PT53	PT52	PT51	PT50	PT49	PT48
D15	D14	D13	D12	D11	D10	D9	D8
PT47	PT46	PT45	PT44	PT43	PT42	PT41	PT40
D7	D6	D5	D4	D3	D2	D1	D0
PT39	PT38	PT37	PT36	PT35	PT34	PT33	PT32

ビット	初期値	ニーモニック	意味
D31-D0	0x0	PT[63:32]	**暗号処理:平文** 32~63 **ビット設定** **復号処理:暗号文** 32~63 **ビット設定**

アドレス 0x00000010

D31	D30	D29	D28	D27	D26	D25	D24
PT95	PT94	PT93	PT92	PT91	PT90	PT89	PT88
D23	D22	D21	D20	D19	D18	D17	D16
PT87	PT86	PT85	PT84	PT83	PT82	PT81	PT80
D15	D14	D13	D12	D11	D10	D9	D8
PT79	PT78	PT77	PT76	PT75	PT74	PT73	PT72
D7	D6	D5	D4	D3	D2	D1	D0
PT71	PT70	PT69	PT68	PT67	PT66	PT65	PT64

ビット	初期値	ニーモニック	意味
D31-D0	0x0	PT[95:64]	**暗号処理**:**平文** 64~95 **ビット設定** **復号処理**:**暗号文** 64~95 **ビット設定**

アドレス 0x00000014

D31	D30	D29	D28	D27	D26	D25	D24
PT127	PT126	PT125	PT124	PT123	PT122	PT121	PT120
D23	D22	D21	D20	D19	D18	D17	D16
PT119	PT118	PT117	PT116	PT115	PT114	PT113	PT112
D15	D14	D13	D12	D11	D10	D9	D8
PT111	PT110	PT109	PT108	PT107	PT106	PT105	PT104
D7	D6	D5	D4	D3	D2	D1	D0
PT103	PT102	PT101	PT100	PT99	PT98	PT97	PT96

ビット	初期値	ニーモニック	意味
D31-D0	0x0	PT[127:96]	**暗号処理**:平文 96~127 ビット設定 **復号処理**:**暗号文** 64~95 **ビット設定**

アドレス 0x00000018

D31	D30	D29	D28	D27	D26	D25	D24
OK1_31	OK1_30	OK1_29	OK1_28	OK1_27	OK1_26	OK1_25	OK1_24
D23	D22	D21	D20	D19	D18	D17	D16
OK1_23	OK1_22	OK1_21	OK1_20	OK1_19	OK1_18	OK1_17	OK1_16
D15	D14	D13	D12	D11	D10	D9	D8
OK1_15	OK1_14	OK1_13	OK1_12	OK1_11	OK1_10	OK1_9	OK1_8
D7	D6	D5	D4	D3	D2	D1	D0
OK1_7	OK1_6	OK1_5	OK1_4	OK1_3	OK1_2	OK1_1	OK1_0

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK1[31:0]	Original Key1 0~31 ビット設定 Single Key モードではこちらの値のみが使用される。

アドレス 0x0000001C

D31	D30	D29	D28	D27	D26	D25	D24
OK1_63	OK1_62	OK1_61	OK1_60	OK1_59	OK1_58	OK1_57	OK1_56
D23	D22	D21	D20	D19	D18	D17	D16
OK1_55	OK1_54	OK1_53	OK1_52	OK1_51	OK1_50	OK1_49	OK1_48
D15	D14	D13	D12	D11	D10	D9	D8
OK1_47	OK1_46	OK1_45	OK1_44	OK1_43	OK1_42	OK1_41	OK1_40
D7	D6	D5	D4	D3	D2	D1	D0
OK1_39	OK1_38	OK1_37	OK1_36	OK1_35	OK1_34	OK1_33	OK1_32

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK1[63:32]	Original Key1 32~63 ビット設定 Single Key モードではこちらの値のみが使用される。

アドレス 0x00000020

D31	D30	D29	D28	D27	D26	D25	D24
OK1_95	OK1_94	OK1_93	OK1_92	OK1_91	OK1_90	OK1_89	OK1_88
D23	D22	D21	D20	D19	D18	D17	D16
OK1_87	OK1_86	OK1_85	OK1_84	OK1_83	OK1_82	OK1_81	OK1_80
D15	D14	D13	D12	D11	D10	D9	D8
OK1_79	OK1_78	OK1_77	OK1_76	OK1_75	OK1_74	OK1_73	OK1_72
D7	D6	D5	D4	D3	D2	D1	D0
OK1_71	OK1_70	OK1_69	OK1_68	OK1_67	OK1_66	OK1_65	OK1_64

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK1[95:64]	Original Key1 64~95 ビット設定 Single Key モードではこちらの値のみが使用される。

アドレス 0x00000024

D31	D30	D29	D28	D27	D26	D25	D24
OK1_127	OK1_126	OK1_125	OK1_124	OK1_123	OK1_122	OK1_121	OK1_120
D23	D22	D21	D20	D19	D18	D17	D16
OK1_119	OK1_118	OK1_117	OK1_116	OK1_115	OK1_114	OK1_113	OK1_112
D15	D14	D13	D12	D11	D10	D9	D8
OK1_111	OK1_110	OK1_109	OK1_108	OK1_107	OK1_106	OK1_105	OK1_104
D7	D6	D5	D4	D3	D2	D1	D0
OK1_103	OK1_102	OK1_101	OK1_100	OK1_99	OK1_98	OK1_97	OK1_96

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK1[127:96]	Original Key1 96~127 ビット設定 Single Key モードではこちらの値のみが使用される。

アドレス 0x00000028

D31	D30	D29	D28	D27	D26	D25	D24
OK2_31	OK2_30	OK2_29	OK2_28	OK2_27	OK2_26	OK2_25	OK2_24
D23	D22	D21	D20	D19	D18	D17	D16
OK2_23	OK2_22	OK2_21	OK2_20	OK2_19	OK2_18	OK2_17	OK2_16
D15	D14	D13	D12	D11	D10	D9	D8
OK2_15	OK2_14	OK2_13	OK2_12	OK2_11	OK2_10	OK2_9	OK2_8
D7	D6	D5	D4	D3	D2	D1	D0
OK2_7	OK2_6	OK2_5	OK2_4	OK2_3	OK2_2	OK2_1	OK2_0

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK2[31:0]	Original Key2 0~31 ビット設定 Double Key モードでのみが使用される。

アドレス 0x0000002C

D31	D30	D29	D28	D27	D26	D25	D24
OK2_63	OK2_62	OK2_61	OK2_60	OK2_59	OK2_58	OK2_57	OK2_56
D23	D22	D21	D20	D19	D18	D17	D16
OK2_55	OK2_54	OK2_53	OK2_52	OK2_51	OK2_50	OK2_49	OK2_48
D15	D14	D13	D12	D11	D10	D9	D8
OK2_47	OK2_46	OK2_45	OK2_44	OK2_43	OK2_42	OK2_41	OK2_40
D7	D6	D5	D4	D3	D2	D1	D0
OK2_39	OK2_38	OK2_37	OK2_36	OK2_35	OK2_34	OK2_33	OK2_32

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK2[63:32]	Original Key2 32~63 ビット設定 Double Key モードでのみが使用される。

アドレス 0x00000030

D31	D30	D29	D28	D27	D26	D25	D24
OK2_95	OK2_94	OK2_93	OK2_92	OK2_91	OK2_90	OK2_89	OK2_88
D23	D22	D21	D20	D19	D18	D17	D16
OK2_87	OK2_86	OK2_85	OK2_84	OK2_83	OK2_82	OK2_81	OK2_80
D15	D14	D13	D12	D11	D10	D9	D8
OK2_79	OK2_78	OK2_77	OK2_76	OK2_75	OK2_74	OK2_73	OK2_72
D7	D6	D5	D4	D3	D2	D1	D0
OK2_71	OK2_70	OK2_69	OK2_68	OK2_67	OK2_66	OK2_65	OK2_64

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK2[95:64]	Original Key2 64~95 ビット設定 Double Key モードでのみが使用される。

アドレス 0x00000034

D31	D30	D29	D28	D27	D26	D25	D24
OK2_127	OK2_126	OK2_125	OK2_124	OK2_123	OK2_122	OK2_121	OK2_120
D23	D22	D21	D20	D19	D18	D17	D16
OK2_119	OK2_118	OK2_117	OK2_116	OK2_115	OK2_114	OK2_113	OK2_112
D15	D14	D13	D12	D11	D10	D9	D8
OK2_111	OK2_110	OK2_109	OK2_108	OK2_107	OK2_106	OK2_105	OK2_104
D7	D6	D5	D4	D3	D2	D1	D0
OK2_103	OK2_102	OK2_101	OK2_100	OK2_99	OK2_98	OK2_97	OK2_96

ビット	初期値	ニーモニック	意味
D31-D0	0x0	OK2[127:96]	Original Key2 96~127 ビット設定 Double Key モードでのみが使用される。

アドレス 0x00000040

D31	D30	D29	D28	D27	D26	D25	D24
Y2_31	Y2_30	Y2_29	Y2_28	Y2_27	Y2_26	Y2_25	Y2_24
D23	D22	D21	D20	D19	D18	D17	D16
Y2_23	Y2_22	Y2_21	Y2_20	Y2_19	Y2_18	Y2_17	Y2_16
D15	D14	D13	D12	D11	D10	D9	D8
Y2_15	Y2_14	Y2_13	Y2_12	Y2_11	Y2_10	Y2_9	Y2_8
D7	D6	D5	D4	D3	D2	D1	D0
Y2_7	Y2_6	Y2_5	Y2_4	Y2_3	Y2_2	Y2_1	Y2_0

ビット	初期値	ニーモニック	意味
D31-D0	0xB8806AC5	Y2[31:0]	Single Key モード時の Label1 計算値 0~31 ビット設定

アドレス 0x00000044

D31	D30	D29	D28	D27	D26	D25	D24
Y2_63	Y2_62	Y2_61	Y2_60	Y2_59	Y2_58	Y2_57	Y2_56
D23	D22	D21	D20	D19	D18	D17	D16
Y2_55	Y2_54	Y2_53	Y2_52	Y2_51	Y2_50	Y2_49	Y2_48
D15	D14	D13	D12	D11	D10	D9	D8
Y2_47	Y2_46	Y2_45	Y2_44	Y2_43	Y2_42	Y2_41	Y2_40
D7	D6	D5	D4	D3	D2	D1	D0
Y2_39	Y2_38	Y2_37	Y2_36	Y2_35	Y2_34	Y2_33	Y2_32

ビット	初期値	ニーモニック	意味
D31-D0	0x3BAD2D4F	Y2[63:32]	Single Key モード時の Label1 計算値 32~63 ビット設定

アドレス 0x00000048

D31	D30	D29	D28	D27	D26	D25	D24
Y2_95	Y2_94	Y2_93	Y2_92	Y2_91	Y2_90	Y2_89	Y2_88
D23	D22	D21	D20	D19	D18	D17	D16
Y2_87	Y2_86	Y2_85	Y2_84	Y2_83	Y2_82	Y2_81	Y2_80
D15	D14	D13	D12	D11	D10	D9	D8
Y2_79	Y2_78	Y2_77	Y2_76	Y2_75	Y2_74	Y2_73	Y2_72
D7	D6	D5	D4	D3	D2	D1	D0
Y2_71	Y2_70	Y2_69	Y2_68	Y2_67	Y2_66	Y2_65	Y2_64

ビット	初期値	ニーモニック	意味
D31-D0	0x3B1565C1	Y2[95:64]	Single Key モード時の Label1 計算値 64~95 ビット設定

アドレス 0x0000004C

D31	D30	D29	D28	D27	D26	D25	D24
Y2_127	Y2_126	Y2_125	Y2_124	Y2_123	Y2_122	Y2_121	Y2_120
D23	D22	D21	D20	D19	D18	D17	D16
Y2_119	Y2_118	Y2_117	Y2_116	Y2_115	Y2_114	Y2_113	Y2_112
D15	D14	D13	D12	D11	D10	D9	D8
Y2_111	Y2_110	Y2_109	Y2_108	Y2_107	Y2_106	Y2_105	Y2_104
D7	D6	D5	D4	D3	D2	D1	D0
Y2_103	Y2_102	Y2_101	Y2_100	Y2_99	Y2_98	Y2_97	Y2_96

ビット	初期値	ニーモニック	意味
D31-D0	0x628CCDA0	Y2[127:96]	Single Key モード時の Label1 計算値 96~127 ビット設定

アドレス 0x00000050

D31	D30	D29	D28	D27	D26	D25	D24
Z2_31	Z2_30	Z2_29	Z2_28	Z2_27	Z2_26	Z2_25	Z2_24
D23	D22	D21	D20	D19	D18	D17	D16
Z2_23	Z2_22	Z2_21	Z2_20	Z2_19	Z2_18	Z2_17	Z2_16
D15	D14	D13	D12	D11	D10	D9	D8
Z2_15	Z2_14	Z2_13	Z2_12	Z2_11	Z2_10	Z2_9	Z2_8
D7	D6	D5	D4	D3	D2	D1	D0
Z2_7	Z2_6	Z2_5	Z2_4	Z2_3	Z2_2	Z2_1	Z2_0

ビット	初期値	ニーモニック	意味
D31-D0	0xFE316B7B	Z2[31:0]	Double Key モード時の Label2 計算値 0~31 ビット設定

アドレス 0x00000054

D31	D30	D29	D28	D27	D26	D25	D24
Z2_63	Z2_62	Z2_61	Z2_60	Z2_59	Z2_58	Z2_57	Z2_56
D23	D22	D21	D20	D19	D18	D17	D16
Z2_55	Z2_54	Z2_53	Z2_52	Z2_51	Z2_50	Z2_49	Z2_48
D15	D14	D13	D12	D11	D10	D9	D8
Z2_47	Z2_46	Z2_45	Z2_44	Z2_43	Z2_42	Z2_41	Z2_40
D7	D6	D5	D4	D3	D2	D1	D0
Z2_39	Z2_38	Z2_37	Z2_36	Z2_35	Z2_34	Z2_33	Z2_32

ビット	初期値	ニーモニック	意味
D31-D0	0x8066CBBB	Z2[63:32]	Double Key モード時の Label2 計算値 32~63 ビット設定

アドレス 0x00000058

D31	D30	D29	D28	D27	D26	D25	D24
Z2_95	Z2_94	Z2_93	Z2_92	Z2_91	Z2_90	Z2_89	Z2_88
D23	D22	D21	D20	D19	D18	D17	D16
Z2_87	Z2_86	Z2_85	Z2_84	Z2_83	Z2_82	Z2_81	Z2_80
D15	D14	D13	D12	D11	D10	D9	D8
Z2_79	Z2_78	Z2_77	Z2_76	Z2_75	Z2_74	Z2_73	Z2_72
D7	D6	D5	D4	D3	D2	D1	D0
Z2_71	Z2_70	Z2_69	Z2_68	Z2_67	Z2_66	Z2_65	Z2_64

ビット	初期値	ニーモニック	意味
D31-D0	0x65CD3C2E	Z2[95:64]	Double Key モード時の Label2 計算値 64~95 ビット設定

D31	D30	D29	D28	D27	D26	D25	D24
Z2_127	Z2_126	Z2_125	Z2_124	Z2_123	Z2_122	Z2_121	Z2_120
D23	D22	D21	D20	D19	D18	D17	D16
Z2_119	Z2_118	Z2_117	Z2_116	Z2_115	Z2_114	Z2_113	Z2_112
D15	D14	D13	D12	D11	D10	D9	D8
Z2_111	Z2_110	Z2_109	Z2_108	Z2_107	Z2_106	Z2_105	Z2_104
D7	D6	D5	D4	D3	D2	D1	D0
Z2_103	Z2_102	Z2_101	Z2_100	Z2_99	Z2_98	Z2_97	Z2_96

ビット	初期値	ニーモニック	意味
D31-D0	0xF9251A23	Z2[127:96]	Double Key モード時の Label2 計算値 96~127 ビット設定

アドレス 0x00000060

D31	D30	D29	D28	D27	D26	D25	D24
W2_31	W2_30	W2_29	W2_28	W2_27	W2_26	W2_25	W2_24
D23	D22	D21	D20	D19	D18	D17	D16
W2_23	W2_22	W2_21	W2_20	W2_19	W2_18	W2_17	W2_16
D15	D14	D13	D12	D11	D10	D9	D8
W2_15	W2_14	W2_13	W2_12	W2_11	W2_10	W2_9	W2_8
D7	D6	D5	D4	D3	D2	D1	D0
W2_7	W2_6	W2_5	W2_4	W2_3	W2_2	W2_1	W2_0

ビット	初期値	ニーモニック	意味
D31-D0	0xEEF127F5	W2[31:0]	Double Key モード時の Label3 計算値 0~31 ビット設定

アドレス 0x00000064

D31	D30	D29	D28	D27	D26	D25	D24
W2_63	W2_62	W2_61	W2_60	W2_59	W2_58	W2_57	W2_56
D23	D22	D21	D20	D19	D18	D17	D16
W2_55	W2_54	W2_53	W2_52	W2_51	W2_50	W2_49	W2_48
D15	D14	D13	D12	D11	D10	D9	D8
W2_47	W2_46	W2_45	W2_44	W2_43	W2_42	W2_41	W2_40
D7	D6	D5	D4	D3	D2	D1	D0
W2_39	W2_38	W2_37	W2_36	W2_35	W2_34	W2_33	W2_32

ビット	初期値	ニーモニック	意味
D31-D0	0x9FFB1E12	W2[63:32]	Double Key モード時の Label3 計算値 32~63 ビット設定

アドレス 0x00000068

D31	D30	D29	D28	D27	D26	D25	D24
W2_95	W2_94	W2_93	W2_92	W2_91	W2_90	W2_89	W2_88
D23	D22	D21	D20	D19	D18	D17	D16
W2_87	W2_86	W2_85	W2_84	W2_83	W2_82	W2_81	W2_80
D15	D14	D13	D12	D11	D10	D9	D8
W2_79	W2_78	W2_77	W2_76	W2_75	W2_74	W2_73	W2_72
D7	D6	D5	D4	D3	D2	D1	D0
W2_71	W2_70	W2_69	W2_68	W2_67	W2_66	W2_65	W2_64

ビット	初期値	ニーモニック	意味
D31-D0	0x656B71FF	W2[95:64]	Double Key モード時の Label3 計算値 64~95 ビット設定

アドレス 0x0000006C

D31	D30	D29	D28	D27	D26	D25	D24
W2_127	W2_126	W2_125	W2_124	W2_123	W2_122	W2_121	W2_120
D23	D22	D21	D20	D19	D18	D17	D16
W2_119	W2_118	W2_117	W2_116	W2_115	W2_114	W2_113	W2_112
D15	D14	D13	D12	D11	D10	D9	D8
W2_111	W2_110	W2_109	W2_108	W2_107	W2_106	W2_105	W2_104
D7	D6	D5	D4	D3	D2	D1	D0
W2_103	W2_102	W2_101	W2_100	W2_99	W2_98	W2_97	W2_96

ビット	初期値	ニーモニック	意味
D31-D0	0x5DE28625	W2[127:96]	Double Key モード時の Label3 計算値 96~127 ビット設定

アドレス 0x00000070

D31	D30	D29	D28	D27	D26	D25	D24
ENC_31	ENC_30	ENC_29	ENC_28	ENC_27	ENC_26	ENC_25	ENC_24
D23	D22	D21	D20	D19	D18	D17	D16
ENC_23	ENC_22	ENC_21	ENC_20	ENC_19	ENC_18	ENC_17	ENC_16
D15	D14	D13	D12	D11	D10	D9	D8
ENC_15	ENC_14	ENC_13	ENC_12	ENC_11	ENC_10	ENC_9	ENC_8
D7	D6	D5	D4	D3	D2	D1	D0
ENC_7	ENC_6	ENC_5	ENC_4	ENC_3	ENC_2	ENC_1	ENC_0

ビット	初期値	ニーモニック	意味
D31-D0	-	ENC[31:0]	**暗号化データ** 0~31 **ビット設定**

アドレス 0x00000074

D31	D30	D29	D28	D27	D26	D25	D24
ENC_63	ENC_62	ENC_61	ENC_60	ENC_59	ENC_58	ENC_57	ENC_56
D23	D22	D21	D20	D19	D18	D17	D16
ENC_55	ENC_54	ENC_53	ENC_52	ENC_51	ENC_50	ENC_49	ENC_48
D15	D14	D13	D12	D11	D10	D9	D8
ENC_47	ENC_46	ENC_45	ENC_44	ENC_43	ENC_42	ENC_41	ENC_40
D7	D6	D5	D4	D3	D2	D1	D0
ENC_39	ENC_38	ENC_37	ENC_36	ENC_35	ENC_34	ENC_33	ENC_32

ビット	初期値	ニーモニック	意味
D31-D0	-	ENC[63:32]	**暗号化データ** 32~63 **ビット設定**

アドレス 0x00000078

D31	D30	D29	D28	D27	D26	D25	D24
ENC_95	ENC_94	ENC_93	ENC_92	ENC_91	ENC_90	ENC_89	ENC_88
D23	D22	D21	D20	D19	D18	D17	D16
ENC_87	ENC_86	ENC_85	ENC_84	ENC_83	ENC_82	ENC_81	ENC_80
D15	D14	D13	D12	D11	D10	D9	D8
ENC_79	ENC_78	ENC_77	ENC_76	ENC_75	ENC_74	ENC_73	ENC_72
D7	D6	D5	D4	D3	D2	D1	D0
ENC_71	ENC_70	ENC_69	ENC_68	ENC_67	ENC_66	ENC_65	ENC_64

ビット	初期値	ニーモニック	意味
D31-D0	-	ENC[95:64]	**暗号化データ** 64~95 **ビット設定**

アドレス 0x0000007C

D31	D30	D29	D28	D27	D26	D25	D24
ENC_127	ENC_126	ENC_125	ENC_124	ENC_123	ENC_122	ENC_121	ENC_120
D23	D22	D21	D20	D19	D18	D17	D16
ENC_119	ENC_118	ENC_117	ENC_116	ENC_115	ENC_114	ENC_113	ENC_112
D15	D14	D13	D12	D11	D10	D9	D8
ENC_111	ENC_110	ENC_109	ENC_108	ENC_107	ENC_106	ENC_105	ENC_104
D7	D6	D5	D4	D3	D2	D1	D0
ENC_103	ENC_102	ENC_101	ENC_100	ENC_99	ENC_98	ENC_97	ENC_96

ビット	初期値	ニーモニック	意味
D31-D0	-	ENC[127:96]	**暗号化データ** 96~127 **ビット設定**

アドレス 0x00000080

D31	D30	D29	D28	D27	D26	D25	D24
DEC_31	DEC_30	DEC_29	DEC_28	DEC_27	DEC_26	DEC_25	DEC_24
D23	D22	D21	D20	D19	D18	D17	D16
DEC_23	DEC_22	DEC_21	DEC_20	DEC_19	DEC_18	DEC_17	DEC_16
D15	D14	D13	D12	D11	D10	D9	D8
DEC_15	DEC_14	DEC_13	DEC_12	DEC_11	DEC_10	DEC_9	DEC_8
D7	D6	D5	D4	D3	D2	D1	D0
DEC_7	DEC_6	DEC_5	DEC_4	DEC_3	DEC_2	DEC_1	DEC_0

ビット	初期値	ニーモニック	意味
D31-D0	-	DEC[31:0]	復号化データ 0~31 ビット設定

アドレス 0x00000084

D31	D30	D29	D28	D27	D26	D25	D24
DEC_63	DEC_62	DEC_61	DEC_60	DEC_59	DEC_58	DEC_57	DEC_56
D23	D22	D21	D20	D19	D18	D17	D16
DEC_55	DEC_54	DEC_53	DEC_52	DEC_51	DEC_50	DEC_49	DEC_48
D15	D14	D13	D12	D11	D10	D9	D8
DEC_47	DEC_46	DEC_45	DEC_44	DEC_43	DEC_42	DEC_41	DEC_40
D7	D6	D5	D4	D3	D2	D1	D0
DEC_39	DEC_38	DEC_37	DEC_36	DEC_35	DEC_34	DEC_33	DEC_32

ビット	初期値	ニーモニック	意味
D31-D0	-	DEC[63:32]	復号化データ 32~63 ビット設定

アドレス 0x00000088

D31	D30	D29	D28	D27	D26	D25	D24
DEC_95	DEC_94	DEC_93	DEC_92	DEC_91	DEC_90	DEC_89	DEC_88
D23	D22	D21	D20	D19	D18	D17	D16
DEC_87	DEC_86	DEC_85	DEC_84	DEC_83	DEC_82	DEC_81	DEC_80
D15	D14	D13	D12	D11	D10	D9	D8
DEC_79	DEC_78	DEC_77	DEC_76	DEC_75	DEC_74	DEC_73	DEC_72
D7	D6	D5	D4	D3	D2	D1	D0
DEC_71	DEC_70	DEC_69	DEC_68	DEC_67	DEC_66	DEC_65	DEC_64

ビット	初期値	ニーモニック	意味
D31-D0	-	DEC[95:63]	**復号化データ** 63~95 **ビット設定**

アドレス 0x0000008C

D31	D30	D29	D28	D27	D26	D25	D24
DEC_127	DEC_126	DEC_125	DEC_124	DEC_123	DEC_122	DEC_121	DEC_120
D23	D22	D21	D20	D19	D18	D17	D16
DEC_119	DEC_118	DEC_117	DEC_116	DEC_115	DEC_114	DEC_113	DEC_112
D15	D14	D13	D12	D11	D10	D9	D8
DEC_111	DEC_110	DEC_109	DEC_108	DEC_107	DEC_106	DEC_105	DEC_104
D7	D6	D5	D4	D3	D2	D1	D0
DEC_103	DEC_102	DEC_101	DEC_100	DEC_99	DEC_98	DEC_97	DEC_96

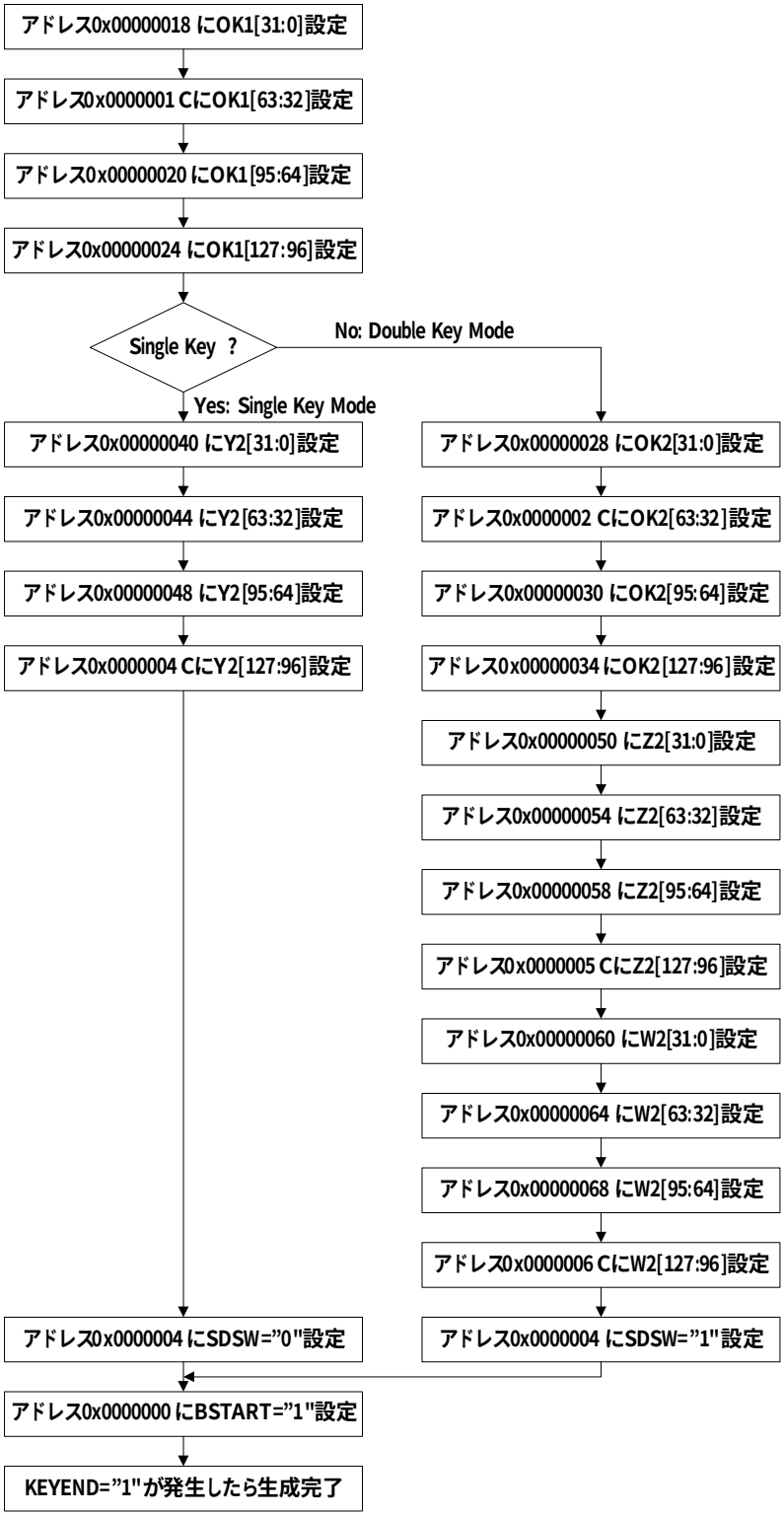
ビット	初期値	ニーモニック	意味
D31-D0	-	DEC[127:96]	**復号化データ** 96~127 **ビット設定**

4.1.5. KEY生成フロー

暗号化を行う前に SUB KEY を生成する必要がある。

KEY を生成するには、SDSW,OK1,OK2,Y2,Z2 および W2 の設定が必要である。

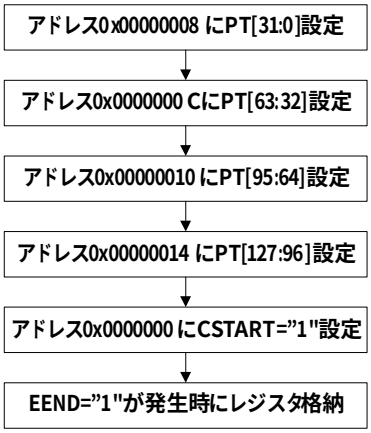
以下に KEY 生成フローを示す。



4.1.6. 暗号化フロー

暗号化は、SUB KEY を生成した後に実行する。

SUB KEY 実行した後のフローを以下に示す。

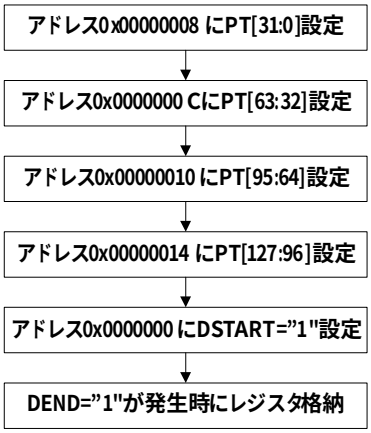


※ 0x70,74,78,7C

4.1.7. 復号化フロー

復号化は、SUB KEY を生成した後に実行する。

SUB KEY 実行した後のフローを以下に示す。



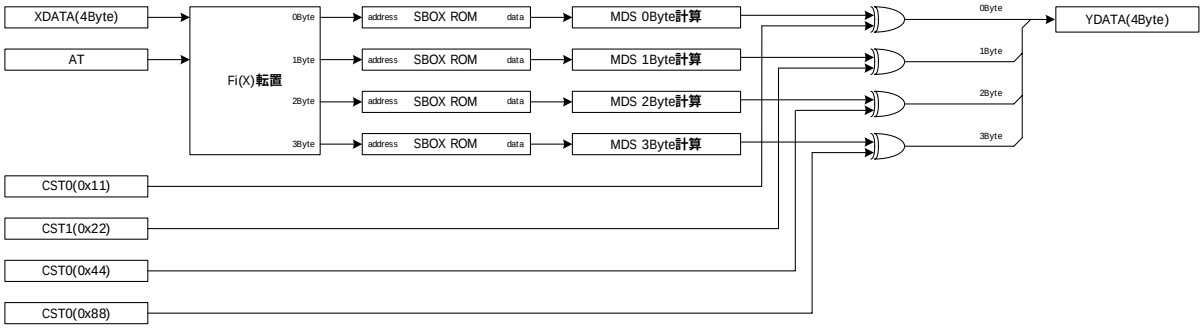
※ 0x80,84,88,8C

4.2. f関数

f 関数は、3.3.3.で規定する処理を行う。

f 関数に入力されるデータは 4Byte のデータで、Byte 単位で処理を行う。

入力する 4Byte データの転置が終わった後、各 Byte 毎に SBOX ROM から変換データを読出し MDS 計算を行う。MDS 計算結果と CST の Ex-OR 出力が f 関数出力となる。



4.2.1. SBOX ROM

SBOX は、下記に示す ROM を実装し、転置後の各 Byte データをアドレスとして用いる。

address	data	address	data	address	data	address	data
00	16	10	b6	20	ed	30	a7
01	5e	11	70	21	5c	31	64
02	d3	12	06	22	ca	32	13
03	af	13	d0	23	05	33	ab
04	36	14	81	24	87	34	e9
05	43	15	82	25	bf	35	09
06	a6	16	fa	26	24	36	25
07	49	17	a1	27	4c	37	54
08	33	18	10	28	51	38	2d
09	93	19	b5	29	ec	39	31
0A	3b	1A	3c	2A	17	3A	69
0B	21	1B	ba	2B	61	3B	f5
0C	91	1C	97	2C	22	3C	37
0D	df	1D	85	2D	f0	3D	67
0E	47	1E	b7	2E	3e	3E	fe
0F	f4	1F	79	2F	18	3F	1d

address	data	address	data	address	data	address	data
40	0b	50	94	60	5b	70	08
41	28	51	8b	61	23	71	4e
42	a3	52	d5	62	34	72	e3
43	2f	53	c4	63	38	73	d7
44	e4	54	90	64	03	74	1e
45	0f	55	6b	65	8c	75	b3
46	d4	56	f8	66	46	76	50
47	da	57	9d	67	68	77	5d
48	1b	58	c5	68	cd	78	c6
49	fc	59	db	69	1a	79	0e
4A	e6	5A	ea	6A	1c	7A	ad
4B	ac	5B	e2	6B	41	7B	cf
4C	53	5C	ae	6C	7d	7C	d6
4D	04	5D	63	6D	a0	7D	eb
4E	27	5E	07	6E	9c	7E	0d
4F	a9	5F	7a	6F	dd	7F	b1

次項に続く

address	data	address	data	address	data	address	data
80	fb	90	a5	A0	de	B0	9b
81	7c	91	8e	A1	6a	B1	9e
82	c3	92	3d	A2	6d	B2	d9
83	2e	93	76	A3	32	B3	95
84	65	94	86	A4	84	B4	b9
85	48	95	57	A5	72	B5	a4
86	b8	96	bc	A6	8a	B6	02
87	8f	97	bd	A7	d8	B7	f7
88	ce	98	11	A8	f9	B8	96
89	e7	99	75	A9	dc	B9	73
8A	62	9A	71	AA	9a	BA	56
8B	d2	9B	78	AB	89	BB	be
8C	12	9C	1f	AC	9f	BC	7f
8D	4a	9D	ef	AD	88	BD	80
8E	c8	9E	e0	AE	14	BE	7e
8F	26	9F	0c	AF	2a	BF	83

address	data	address	data	address	data	address	data
C0	00	D0	58	E0	2c	F0	c1
C1	01	D1	3f	E1	45	F1	0a
C2	f6	D2	cc	E2	6c	F2	15
C3	8d	D3	fd	E3	92	F3	98
C4	7b	D4	ee	E4	66	F4	a2
C5	d1	D5	b2	E5	42	F5	c2
C6	52	D6	40	E6	39	F6	44
C7	cb	D7	ff	E7	f3	F7	30
C8	b0	D8	99	E8	77	F8	55
C9	e1	D9	2b	E9	bb	F9	4d
CA	c7	DA	5f	EA	19	FA	c9
CB	e5	DB	60	EB	59	FB	a8
CC	29	DC	aa	EC	20	FC	5a
CD	c0	DD	4b	ED	6f	FD	f1
CE	4f	DE	b4	EE	35	FE	6e
CF	e8	DF	74	EF	f2	FF	3a

4.2.2. MDS計算

本 FPGA は、3.3.6.に基づき固定行列におけるガロア体演算回路を実装する。
演算方法は、HyRAL 暗号_20091214.xls 倍数計算シートによる。

ガロア体演算

0x01*DATA(8bit)

演算結果(8bit) = DATA

0x02*DATA(8bit)

DATA1(9bit) = DATA * 2
DATA1の最上位ビットが1であるとき 演算結果(8bit) = DATA1[7:0] @ 0x1B
DATA1の最上位ビットが0であるとき 演算結果(8bit) = DATA1[7:0]

0x03*DATA(8bit)

DATA1(9bit) = DATA * 2 @ DATA
DATA1の最上位ビットが1であるとき 演算結果(8bit) = DATA1[7:0] @ 0x1B
DATA1の最上位ビットが0であるとき 演算結果(8bit) = DATA1[7:0]

0x04*DATA(8bit)

DATA1(10bit) = DATA * 4
DATA1の上位2ビットが0x3であるとき演算結果(8bit) = DATA1[7:0] @ 0x36 @ 0x1B
DATA1の上位2ビットが0x2であるとき演算結果(8bit) = DATA1[7:0] @ 0x36
DATA1の上位2ビットが0x1であるとき演算結果(8bit) = DATA1[7:0] @ 0x1B
DATA1の上位2ビットが0x0であるとき演算結果(8bit) = DATA1[7:0]

0x05*DATA(8bit)

DATA1(10bit) = DATA * 4 @ DATA
DATA1の上位2ビットが0x3であるとき演算結果(8bit) = DATA1[7:0] @ 0x36 @ 0x1B
DATA1の上位2ビットが0x2であるとき演算結果(8bit) = DATA1[7:0] @ 0x36
DATA1の上位2ビットが0x1であるとき演算結果(8bit) = DATA1[7:0] @ 0x1B
DATA1の上位2ビットが0x0であるとき演算結果(8bit) = DATA1[7:0]

0x07*DATA(8bit)

DATA1(10bit) = DATA * 4 @ DATA * 2 @ DATA
DATA1の上位2ビットが0x3であるとき演算結果(8bit) = DATA1[7:0] @ 0x36 @ 0x1B
DATA1の上位2ビットが0x2であるとき演算結果(8bit) = DATA1[7:0] @ 0x36
DATA1の上位2ビットが0x1であるとき演算結果(8bit) = DATA1[7:0] @ 0x1B
DATA1の上位2ビットが0x0であるとき演算結果(8bit) = DATA1[7:0]

4.3. モジュール仕様

HyRAL 暗号化モジュールは、HyRAL128_main をトップモジュールして、下記構成となっている。

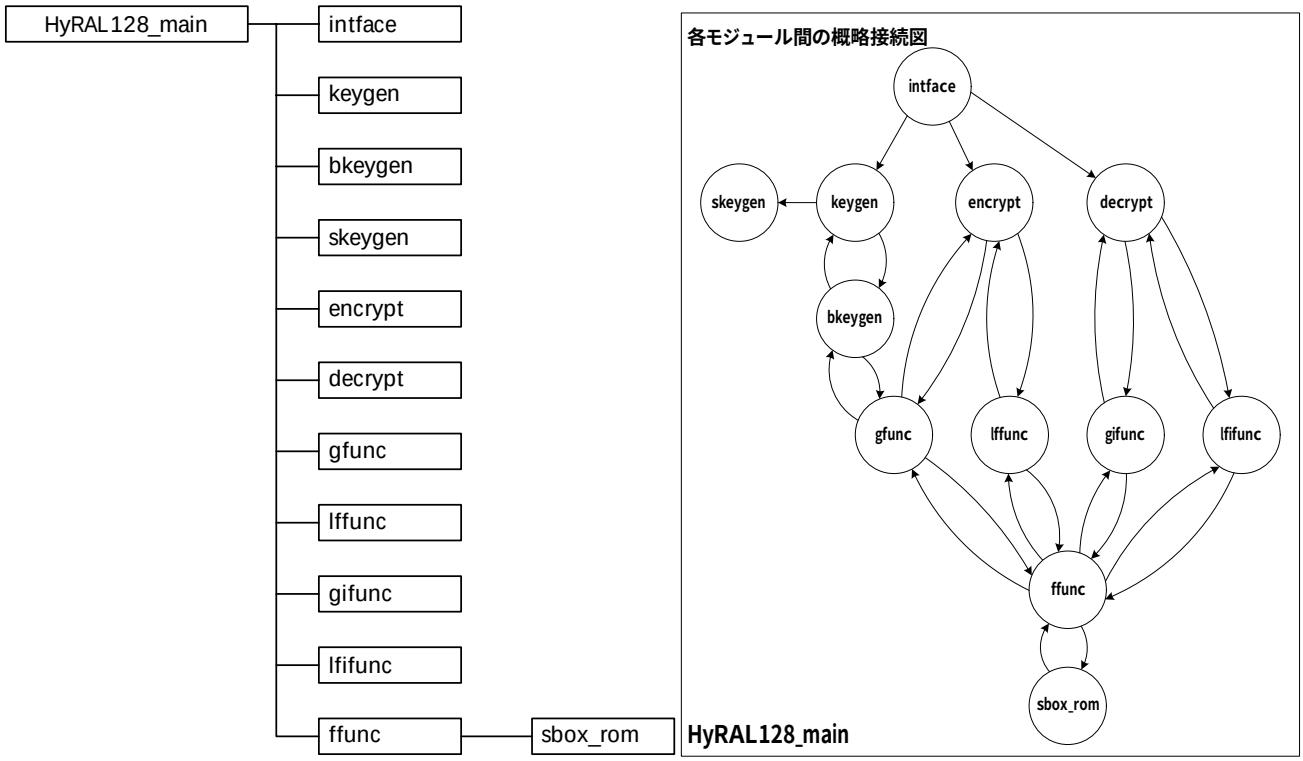


図 4-3.1 モジュール構成*2 版

4.3.1. HyRAL128_main *2 版

本 FPGA のトップファイルである。下記 IO が割り当てられる。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
DATA	Input	32	データバス
ASTRB	Input	1	アドレスストローブ
DSTRB	Input	1	データストローブ
KEYEND	Output	1	SUB KEY 生成完了信号
EEND	Output	1	暗号文生成完了信号
DEND	Output	1	復号文生成完了信号

本 FPGA ではピン定義などは行わない。

本モジュールで図 4-3.1 で示す下位モジュールを呼び出す。

4.3.2. keygen *2 版

Key Material を生成する TOP ファイルである。下記 IO が割り当てられる。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
SDSW	Input	1	Single/Double KEY Mode Select
BSTART	Input	1	SUBKEY 生成開始信号
OK1	Input	128	Original Key1
OK2	Input	128	Original Key2 Double Key Mode 時に使用されます。
CONST1	Input	128	Label1 の Y2 を事前計算した値
CONST2	Input	128	Label2 の Z2 を事前計算した値
CONST3	Input	128	Label3 の W2 を事前計算した値
CST	Input	32	f 関数用変数
IKM1	Input	128	bkeygen 出力 KM1 値
IKM2	Input	128	bkeygen 出力 KM2 値
IKM3	Input	128	bkeygen 出力 KM3 値
IKM4	Input	128	bkeygen 出力 KM4 値
KM1	Output	128	Key Material1
KM2	Output	128	Key Material2
KM3	Output	128	Key Material3
KM4	Output	128	Key Material4
KEYEND	Output	1	KEYMaterial 生成終了信号
KSTART	Output	1	Key 作成スタート信号
CONST	Output	128	Y2,Z2,W2 のいづれかの値で bkeygen で使用する
OK	Output	128	OK1 もしくは OK2 で bkeygen で使用する

Single モードでは、CONST1 を用い

Double モードでは、CONST2,CONST3 を用い KeyMaterial を生成する。

4.3.3. bkeygen

Key Material を演算する冗長な回路である。下記 IO が割り当てられる。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
KSTART	Input	1	SUBKEY 生成開始信号
OK	Input	128	Original Key
CONST	Input	128	Label から事前計算された値
GEND	Input	1	G 関数処理終了信号
GODATA	Input	128	G 関数処理結果データ
KM1	Output	128	Key Material1
KM2	Output	128	Key Material2
KM3	Output	128	Key Material3
KM4	Output	128	Key Material4
BEND	Output	1	SUB KEY 生成終了信号
GREQ	Output	1	G 関数要求信号
GSEL	Output	1	G1,G2 指定信号
GDATA	Output	128	G 関数処理データ
GSTART	Output	1	G 関数開始信号

本モジュールはステートマシンで構成され、1つの gfunc モジュールを用い順次下記フローで KM1~4 までを導く。KM4 生成と同時に BEND パルスを 1CLK パルス幅で出力する。

4.3.4. skeygen

Sub Key を生成するモジュールである。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
SDSW	Input	1	Single/Double KEY Mode Select
KM1	Input	128	Key Material1
KM2	Input	128	Key Material2
KM3	Input	128	Key Material3
KM4	Input	128	Key Material4
IK1	Output	128	Sub Key IK1
IK2	Output	128	Sub Key IK2
IK3	Output	128	Sub Key IK3
IK4	Output	128	Sub Key IK4
IK5	Output	128	Sub Key IK5
IK6	Output	128	Sub Key IK6
RK1	Output	128	Sub Key RK1
RK2	Output	128	Sub Key RK2
RK3	Output	128	Sub Key RK3
RK4	Output	128	Sub Key RK4
RK5	Output	128	Sub Key RK5
RK6	Output	128	Sub Key RK6
RK7	Output	128	Sub Key RK7
RK8	Output	128	Sub Key RK8
RK9	Output	128	Sub Key RK9

3.2.で規定する Sub Key 生成を Key Material 入力後 1CLK で IK および RK を算出する。
SDSW の値により、シングルモードかダブルモードかを判断し算出方法を変更する。

4.3.5. encrypt

暗号文を生成するモジュールである。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
SDSW	Input	1	Single/Double KEY Mode Select
CSTART	Input	1	暗号化開始信号
IK1	Input	128	Sub Key IK1
IK2	Input	128	Sub Key IK2
IK3	Input	128	Sub Key IK3
IK4	Input	128	Sub Key IK4
IK5	Input	128	Sub Key IK5
IK6	Input	128	Sub Key IK6
RK1	Input	128	Sub Key RK1
RK2	Input	128	Sub Key RK2
RK3	Input	128	Sub Key RK3
RK4	Input	128	Sub Key RK4
RK5	Input	128	Sub Key RK5
RK6	Input	128	Sub Key RK6
RK7	Input	128	Sub Key RK7
RK8	Input	128	Sub Key RK8
RK9	Input	128	Sub Key RK9
PT	Input	128	平文
GEND	Input	1	G 関数処理終了信号
GODATA	Input	128	G 関数処理結果データ
LFEND	Input	1	F 関数処理終了信号
LFODATA	Input	128	F 関数処理結果データ
CT	Output	128	暗号文
CEND	Output	1	暗号化終了信号
GREQ	Output	1	G 関数要求信号
GSEL	Output	1	G1,G2 指定信号
GDATA	Output	128	G 関数処理データ
GSTART	Output	1	G 関数開始信号
LFSTART	Output	1	F 関数開始信号
LFSEL	Output	1	F1,F2 指定信号
LFDATA	Output	128	F 関数処理データ

IK	Output	128	F 関数処理データ

3.3.に規定するように SingleKeyMode か DoubleKeyMode かにより処理が異なる。
本モジュールではステートマシンにて制御を行っている。

4.3.6. decrypt

復号文を生成するモジュールである。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
SDSW	Input	1	Single/Double KEY Mode Select
CSTART	Input	1	復号化開始信号
IK1	Input	128	Sub Key IK1
IK2	Input	128	Sub Key IK2
IK3	Input	128	Sub Key IK3
IK4	Input	128	Sub Key IK4
IK5	Input	128	Sub Key IK5
IK6	Input	128	Sub Key IK6
RK1	Input	128	Sub Key RK1
RK2	Input	128	Sub Key RK2
RK3	Input	128	Sub Key RK3
RK4	Input	128	Sub Key RK4
RK5	Input	128	Sub Key RK5
RK6	Input	128	Sub Key RK6
RK7	Input	128	Sub Key RK7
RK8	Input	128	Sub Key RK8
RK9	Input	128	Sub Key RK9
PT	Input	128	暗号文
GEND	Input	1	G-関数処理終了信号
GODATA	Input	128	G-関数処理結果データ
LFEND	Input	1	F-関数処理終了信号
LFODATA	Input	128	F-関数処理結果データ
CT	Output	128	復号文
CEND	Output	1	復号化終了信号
GREQ	Output	1	G-関数要求信号
GSEL	Output	1	G_1-,G_2-指定信号
GDATA	Output	128	G-関数処理データ

GSTART	Output	1	G-関数開始信号
LFSTART	Output	1	F-関数開始信号
LFSEL	Output	1	F_1-,F_2-指定信号
LFDATA	Output	128	F-関数処理データ
IK	Output	128	F-関数処理データ

3.4.に規定するように SingleKeyMode か DoubleKeyMode かにより処理が異なる。
本モジュールではステートマシンにて制御を行っている。

4.3.7. gfunc

G 関数を演算するモジュールである。下記 IO が割り当てられる。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
GSTART	Input	1	G 関数開始信号
XDATA	Input	128	入力データ
GSEL	Input	128	G1 , G2 選択信号 0: G1, 1:G2
FEND	Input	1	f 関数処理終了信号
FODATA	Input	32	f 関数処理結果データ
YDATA	Output	128	G 関数出力データ
GEND	Output	1	G 関数終了信号
FSTART	Output	1	f 関数処理開始信号
FDATA	Output	32	f 関数処理データ
AT	Output	3	f 関数種類指定信号
FREQ	Output	1	f 関数処理要求信号

本モジュールはステートマシンで構成され、1つの ffunc モジュールを用い 3.3.2.で規定する処理を順次実施し YDATA を返す。G 関数出力データが求まると同時に GEND を 1CLK 幅で出力する。

4.3.8. ffunc

F 関数を演算するモジュールである。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
LFSTART	Input	1	F 関数開始信号
XDATA	Input	128	入力データ
FSEL	Input	1	F1 , F2 選択信号 0: F1, 1:F2
IK	Input	128	Sub Key 入力データ
FEND	Input	1	f 関数処理終了信号
FODATA	Input	32	f 関数処理結果データ
YDATA	Output	128	F 関数出力データ
LFEND	Output	1	F 関数終了信号
FSTART	Output	1	f 関数処理開始信号
FDATA	Output	32	f 関数処理データ
AT	Output	3	f 関数種類指定信号
FREQ	Output	1	f 関数処理要求信号

本モジュールはステートマシンで構成され、1つの ffunc モジュールを用い 3.3.1.で規定する処理を順次実施し YDATA を返す。F 関数出力データが求まると同時に LFEND を 1CLK 幅で出力する。

4.3.9. gifunc

G-関数を演算するモジュールである。下記IOが割り当てられる。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
GSTART	Input	1	G-関数開始信号
XDATA	Input	128	入力データ
GSEL	Input	1	G_1^- , G_2^- 選択信号 0: G_1^-, 1:G_2^-
FEND	Input	1	f 関数処理終了信号
FODATA	Input	32	f 関数処理結果データ
YDATA	Output	128	G-関数出力データ
GEND	Output	1	G-関数終了信号
FSTART	Output	1	f 関数処理開始信号
FDATA	Output	32	f 関数処理データ
AT	Output	3	f 関数種類指定信号
FREQ	Output	1	f 関数処理要求信号

本モジュールはステートマシンで構成され、1つの ffunc モジュールを用い 3.4.2.で規定する処理を順次実施し YDATA を返す。G 関数出力データが求まると同時に GEND を 1CLK 幅で出力する。

4.3.10. ifunc

F-関数を演算するモジュールである。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
LFSTART	Input	1	F-関数開始信号
XDATA	Input	128	入力データ
FSEL	Input	1	F_1^- , F_2^- 選択信号 0: F_1^-, 1:F_2^-
IK	Input	128	Sub Key 入力データ
FEND	Input	1	f 関数処理終了信号
FODATA	Input	32	f 関数処理結果データ
YDATA	Output	128	F-関数出力データ
LFEND	Output	1	F-関数終了信号
FSTART	Output	1	f 関数処理開始信号
FDATA	Output	32	f 関数処理データ
AT	Output	3	f 関数種類指定信号
FREQ	Output	1	f 関数処理要求信号

本モジュールはステートマシンで構成され、1つのffuncモジュールを用い 3.4.1.で規定する処理を順次実施しYDATAを返す。F-関数出力データが求まると同時にLFENDを 1CLK幅で出力する。

4.3.11. ffunc

f 関数を演算するモジュールである。下記 IO が割り当てられる。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
FSTART	Input	1	f 関数開始信号
XDATA	Input	32	入力データ
AT	Input	3	変換番号指定
KEY	Input	32	SUBKEY 入力
CST1	Input	8	f 関数用変数
CST2	Input	8	f 関数用変数
CST3	Input	8	f 関数用変数
CST4	Input	8	f 関数用変数
YDATA	Output	32	f 関数出力データ
FEND	Output	1	f 関数終了信号

下位モジュールに sbox_rom モジュールを使用する。

Sbox_rom モジュールは 4.2.1.で規定する固定値が格納されている ROM である。

AT は、変換番号指定で下記に示す動作をする。

AT	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
動作	f1	f2	f3	f4	f5	f6	f7	f8

AT で指定を受けた fi(X)転置後、4.2.2.で規定する MDS 演算を行い、結果を出力する。

最終結果出力タイミングにて FEND を 1CLK 幅で出力する。

4.3.12. intface

インターフェースモジュールである。

信号名	属性	ビット幅	意味
CLK	Input	1	システムクロック
RST	Input	1	システムリセット。このリセットで初期値となる。
DATA	Input	32	データバス
ASTRB	Input	1	アドレスストローブ
DSTRB	Input	1	データストローブ
EEND	Input	1	暗号化終了信号
ENC	Input	128	暗号化データ
DEND	Input	1	復号化終了信号
DEC	Input	128	復号化データ
PT	Output	128	平文
OK1	Output	128	Original Key1
OK2	Output	128	Original Key2
SDSW	Output	1	Single Key Mode=0, Double key Mode=1
BSTART	Output	1	Sub Key 生成開始信号
CSTART	Output	1	暗号化開始信号
DSTART	Output	1	復号化開始信号
CST	Output	32	f 関数用変数
Y2	Output	128	Label1 の Y2 を事前計算した値
Z2	Output	128	Label2 の Z2 を事前計算した値
W2	Output	128	Label3 の W2 を事前計算した値
RDATA	Output	32	リードデータ

インターフェース回路で 4.1.1.で規定する回路が実装される。

アドレスマップは 4.1.2、レジスタ詳細については 4.1.3.で規定する。