詳細評価報告書
TOYOCRYPT-HS1 H/W 実装評価

平成１３年1月１０日

暗号 H/W 実装

１．　目的

　Ｃプログラムで記述された暗号アルゴリズムを、ＦＰＧＡ（ハードウェア）に実装します。

２．　概要

２．１　暗号アルゴリズム

　Ｃで作成された、「032Call61.c」および「032Call62.c」のプログラムをＨ／Ｗに実装します。

２．２　Ｈ／Ｗ設計方針

・Ｈ／Ｗ設計記述言語は、VerilogHDL を使用
・「032Call61.c」は回路規模優先、「032Call62.c」は実行速度優先
・Ｈ／Ｗへの実装範囲は、暗号と復号の処理部分（Key_Setup は含みません）
・ターゲットデバイスは、アルテラ社の「EP20K600E」

３．　使用ツール

・**ModelSim VHDL/Verilog Version 5.4e (Model Technology)**
・LeonardoSpectrum Level 1 Altera - v1999.1j（Exemplar Logic, Inc.)

４．　資料

４．１　Ｃプログラム

　「032Call61.c」および「032Call62.c」のプログラムリストを、それぞれ「list-1」「list-2」として添付します。

４．２　VerilogHDL ソース

　「call61_1.v」および「call62_1.v」のソースリスト（抜粋）を、それぞれ「list-3」「list-4」として添付します。

５．　結果

「ModelSim」を使用して、シミュレーションを実行し、「Ｃプログラム」の実行結果と同じデータ（暗号と復号のデータ）が得られることを確認しました。

論理合成ツールの結果（抜粋）を、「list-5」および「list-6」として添付します。

ＦＰＧＡによる評価結果を以下に示します。

５．１　「call61_1.v」

１）　動作クロック　：５８．１ＭＨｚ
２）　実行速度　　：５０．２４８ＭＢＰＳ（３２ｂｉｔ／３７クロック）
３）　回路規模　　：１１８８３／２４３２０　ＬＣｓ

５．２　「call62_1.v」

１）　動作クロック　：４５．２ＭＨｚ
２）　実行速度　　：１４４６．４ＭＢＰＳ（３２ｂｉｔ／１クロック）
３）　回路規模　　：１６１４４／２４３２０　ＬＣｓ

５．３　評価結果の見積もり条件

１）　実行速度の見積もりには、Ｋｅｙデータの設定時間は含みません。
　　実際の動作には、Ｋｅｙ設定用データの書き込みが必要です。
２）　動作クロックと回路規模の見積もりは、「LeonardoSpectrum」の
　　実行結果です。

６．　ＦＰＧＡ仕様

６．１　「call61_1.v」

（１）　入出力信号

ＣＬＯＣＫ　：　入力　System_Clock
ｎＲＥＳＥＴ　：　入力　System_Reset
ＰＴＣＴ　：　入力　入力データ(32bit)
ＲＡＤＲ　：　入力　Ｋｅｙ設定用レジスタ・アドレス(9bit)
ＲＤＩＮ　：　入力　Ｋｅｙ設定用レジスタ・データ(32bit)
ＲＷＲＩＴＥ　：　入力　Ｋｅｙ設定用書き込み信号
ＰＲＳＴＡＲＴ：　入力　シーケンス開始信号
ＰＲＥＮＤ　：　出力　シーケンス終了信号
ＲＥＳＵＬＴ　：　出力　出力データ(32bit)

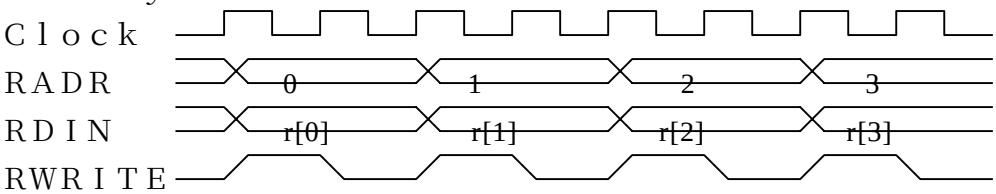
（２）　Ｋｅｙ設定用レジスタのアドレス

レジスタへの書き込みデータは、ＲＡＤＲ＝０～１２７については「ＫｅｙＳｅｔｕｐ」処理後のデータとし、それ以降は「ｆｉｘｅｄ」のＫｅｙデータとなり、以下に示します。

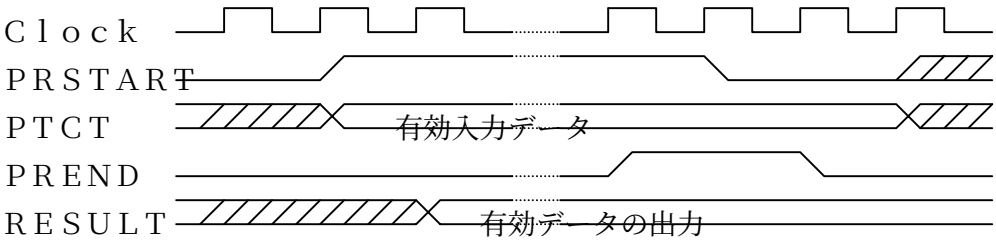
ＲＡＤＲ＝０～１２７　：　ｒ［０］～　ｒ［１２７］
ＲＡＤＲ＝１２８～１３１　：　ｆｉｘｅｄ［０］～ｆｉｘｅｄ［３］

（３）　タイミングチャート

１）　Ｋｅｙ設定用データの書き込み



２）　暗号（または復号）処理



６．２　「call62_1.v」

（１）　入出力信号

ＣＬＯＣＫ　：　入力　System_Clock
ｎＲＥＳＥＴ　：　入力　System_Reset
ＰＴＣＴ　：　入力　入力データ(32bit)
ＲＡＤＲ　：　入力　Ｋｅｙ設定用レジスタ・アドレス(9bit)
ＲＤＩＮ　：　入力　Ｋｅｙ設定用レジスタ・データ(32bit)
ＲＷＲＩＴＥ　：　入力　Ｋｅｙ設定用書き込み信号
ＰＲＳＴＡＲＴ：　入力　シーケンス開始信号（入力データのイネーブル）
ＰＲＥＮＤ　：　出力　シーケンス終了信号（出力データのイネーブル）
ＲＥＳＵＬＴ　：　出力　出力データ(32bit)

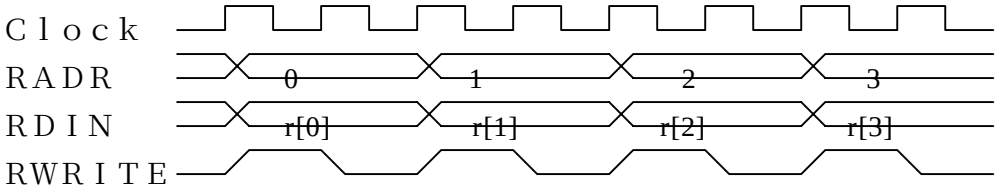
（２）　Ｋｅｙ設定用レジスタのアドレス

レジスタへの書き込みデータは、ＲＡＤＲ＝０～１２７については「ＫｅｙＳｅｔｕｐ」処理後のデータとし、それ以降は「ｆｉｘｅｄ」のＫｅｙデータとなり、以下に示します。

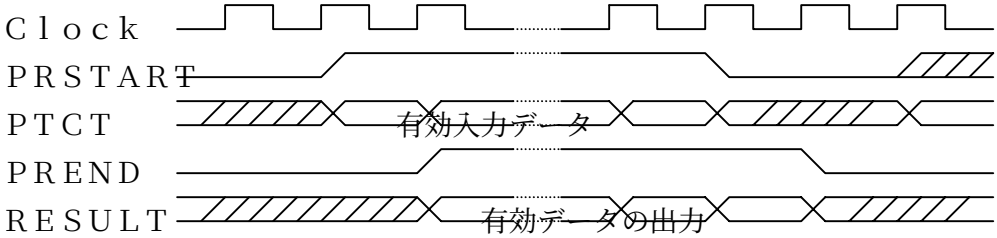
ＲＡＤＲ＝０～１２７　：　ｒ［０］～　ｒ［１２７］
ＲＡＤＲ＝１２８～２５５　：　ｆｉｘｅｄ［０］～ｆｉｘｅｄ［２５５］

（３）　タイミングチャート

１）　Ｋｅｙ設定用データの書き込み



２）　暗号（または復号）処理



A．　添付資料

「list-1」　：　論理合成の対象部分のみを、「032Call61.c」から抜粋して記載します。

―――　転載の許可を得ていないので、省略します　―――

「list-2」　：　論理合成の対象部分のみを、「032Call62.c」から抜粋して記載します。

―――　転載の許可を得ていないので、省略します　―――

「List-3」　：　「call61_1.v」から抜粋して記載します。

―――　ここから始まり　―――

```
    always @(posedge CLOCK or negedge nRESET) begin
        if (nRESET == 1'b0) RESULT <= 32'h00000000;
        else if ((PRSTART == 1'b1) && (CKSTART == 1'b0))
             RESULT <= PTCT ^ W5GATE;
     end
    assign W1GATE = RREG[127];
    assign X1GATE = RREG[62] & RREG[61] & RREG[60] & RREG[59] &
                        RREG[58] & RREG[57] & RREG[56] & RREG[55] &
                        RREG[54] & RREG[53] & RREG[52] & RREG[51] &
                        RREG[50] & RREG[49] & RREG[48] & RREG[47] &
                        RREG[46] & RREG[45] & RREG[44] & RREG[43] &
                        RREG[42] & RREG[41] & RREG[40] & RREG[39] &
                        RREG[38] & RREG[37] & RREG[36] & RREG[35] &
                        RREG[34] & RREG[33] & RREG[32] & RREG[31] &
                        RREG[30] & RREG[29] & RREG[28] & RREG[27] &
                        RREG[26] & RREG[25] & RREG[24] & RREG[23] &
                        RREG[22] & RREG[21] & RREG[20] & RREG[19] &
                        RREG[18] & RREG[17] & RREG[16] & RREG[15] &
                        RREG[14] & RREG[13] & RREG[12] & RREG[11] &
                        RREG[10] & RREG[9]   & RREG[8]   & RREG[7] &
                        RREG[6]   & RREG[5]   & RREG[4]   & RREG[3] &
                        RREG[2]   & RREG[1]   & RREG[0];
    assign W2GATE = W1GATE ^ X1GATE;
    assign X2GATE = RREG[DEG17_16] & RREG[DEG17_15] & RREG[DEG17_14] &
                        RREG[DEG17_13] & RREG[DEG17_12] & RREG[DEG17_11] &
                        RREG[DEG17_10] & RREG[DEG17_9] & RREG[DEG17_8] &
                        RREG[DEG17_7] & RREG[DEG17_6] & RREG[DEG17_5] &
                        RREG[DEG17_4] & RREG[DEG17_3] & RREG[DEG17_2] &
                        RREG[DEG17_1] & RREG[DEG17_0];
    assign W3GATE = W2GATE ^ X2GATE;
    assign W4GATE = W3GATE ^ (RREG[DEG4_3] & RREG[DEG4_2] &
                           RREG[DEG4_1] &   RREG[DEG4_0]);
    assign W5GATE = ((((((((( (((((((((( ((((((((((
```

```
((((((((( ((((((((( ((((((((((
((( W4GATE ^ (RREG[62+63] & RREG[BP_62])) ^
             (RREG[61+63] & RREG[BP_61])) ^
             (RREG[60+63] & RREG[BP_60])) ^
             (RREG[59+63] & RREG[BP_59])) ^
             (RREG[58+63] & RREG[BP_58])) ^
             (RREG[57+63] & RREG[BP_57])) ^
             (RREG[56+63] & RREG[BP_56])) ^
             (RREG[55+63] & RREG[BP_55])) ^
             (RREG[54+63] & RREG[BP_54])) ^
             (RREG[53+63] & RREG[BP_53])) ^
             (RREG[52+63] & RREG[BP_52])) ^
             (RREG[51+63] & RREG[BP_51])) ^
             (RREG[50+63] & RREG[BP_50])) ^
             (RREG[49+63] & RREG[BP_49])) ^
             (RREG[48+63] & RREG[BP_48])) ^
             (RREG[47+63] & RREG[BP_47])) ^
             (RREG[46+63] & RREG[BP_46])) ^
             (RREG[45+63] & RREG[BP_45])) ^
             (RREG[44+63] & RREG[BP_44])) ^
             (RREG[43+63] & RREG[BP_43])) ^
             (RREG[42+63] & RREG[BP_42])) ^
             (RREG[41+63] & RREG[BP_41])) ^
             (RREG[40+63] & RREG[BP_40])) ^
             (RREG[39+63] & RREG[BP_39])) ^
             (RREG[38+63] & RREG[BP_38])) ^
             (RREG[37+63] & RREG[BP_37])) ^
             (RREG[36+63] & RREG[BP_36])) ^
             (RREG[35+63] & RREG[BP_35])) ^
             (RREG[34+63] & RREG[BP_34])) ^
             (RREG[33+63] & RREG[BP_33])) ^
             (RREG[32+63] & RREG[BP_32])) ^
             (RREG[31+63] & RREG[BP_31])) ^
             (RREG[30+63] & RREG[BP_30])) ^
             (RREG[29+63] & RREG[BP_29])) ^
             (RREG[28+63] & RREG[BP_28])) ^
```

```
                                (RREG[27+63] & RREG[BP_27])) ^
                                (RREG[26+63] & RREG[BP_26])) ^
                                (RREG[25+63] & RREG[BP_25])) ^
                                (RREG[24+63] & RREG[BP_24])) ^
                                (RREG[23+63] & RREG[BP_23])) ^
                                (RREG[22+63] & RREG[BP_22])) ^
                                (RREG[21+63] & RREG[BP_21])) ^
                                (RREG[20+63] & RREG[BP_20])) ^
                                (RREG[19+63] & RREG[BP_19])) ^
                                (RREG[18+63] & RREG[BP_18])) ^
                                (RREG[17+63] & RREG[BP_17])) ^
                                (RREG[16+63] & RREG[BP_16])) ^
                                (RREG[15+63] & RREG[BP_15])) ^
                                (RREG[14+63] & RREG[BP_14])) ^
                                (RREG[13+63] & RREG[BP_13])) ^
                                (RREG[12+63] & RREG[BP_12])) ^
                                (RREG[11+63] & RREG[BP_11])) ^
                                (RREG[10+63] & RREG[BP_10])) ^
                                (RREG[9+63] & RREG[BP_9])) ^
                                (RREG[8+63] & RREG[BP_8])) ^
                                (RREG[7+63] & RREG[BP_7])) ^
                                (RREG[6+63] & RREG[BP_6])) ^
                                (RREG[5+63] & RREG[BP_5])) ^
                                (RREG[4+63] & RREG[BP_4])) ^
                                (RREG[3+63] & RREG[BP_3])) ^
                                (RREG[2+63] & RREG[BP_2])) ^
                                (RREG[1+63] & RREG[BP_1])) ^
                                (RREG[0+63] & RREG[BP_0]));
―――　途中省略　―――
    always @(posedge CLOCK or negedge nRESET) begin
        if (nRESET == 1'b0) RREG[0] <= 32'h00000000;
        else if ((RADR == 9'd0) && (RWRITE == 1'b1))
            RREG[0] <= RDIN;
        else if (CKFLAG == 1'b1)
            RREG[0] <= RREG[127] & C0INI;
    end
```

―――　以下省略　―――

―――　ここまで　―――

「List-4」　：　「call62_1.v」から抜粋して記載します。

———　ここから始まり　———

```
    always @(posedge CLOCK or negedge nRESET) begin
        if (nRESET == 1'b0) RESULT <= 32'h00000000;
        else if (PRSTART == 1'b1)
            RESULT <= PTCT ^ W5GATE;
    end

    assign W1GATE = RREG[127];
    assign X1GATE = RREG[62] & RREG[61] & RREG[60] & RREG[59] &
                    RREG[58] & RREG[57] & RREG[56] & RREG[55] &
                    RREG[54] & RREG[53] & RREG[52] & RREG[51] &
                    RREG[50] & RREG[49] & RREG[48] & RREG[47] &
                    RREG[46] & RREG[45] & RREG[44] & RREG[43] &
                    RREG[42] & RREG[41] & RREG[40] & RREG[39] &
                    RREG[38] & RREG[37] & RREG[36] & RREG[35] &
                    RREG[34] & RREG[33] & RREG[32] & RREG[31] &
                    RREG[30] & RREG[29] & RREG[28] & RREG[27] &
                    RREG[26] & RREG[25] & RREG[24] & RREG[23] &
                    RREG[22] & RREG[21] & RREG[20] & RREG[19] &
                    RREG[18] & RREG[17] & RREG[16] & RREG[15] &
                    RREG[14] & RREG[13] & RREG[12] & RREG[11] &
                    RREG[10] & RREG[9]  & RREG[8]  & RREG[7] &
                    RREG[6]  & RREG[5]  & RREG[4]  & RREG[3] &
                    RREG[2]  & RREG[1]  & RREG[0];

    assign W2GATE = W1GATE ^ X1GATE;
    assign X2GATE = RREG[DEG17_16] & RREG[DEG17_15] & RREG[DEG17_14] &
                    RREG[DEG17_13] & RREG[DEG17_12] & RREG[DEG17_11] &
                    RREG[DEG17_10] & RREG[DEG17_9] & RREG[DEG17_8] &
                    RREG[DEG17_7] & RREG[DEG17_6] & RREG[DEG17_5] &
                    RREG[DEG17_4] & RREG[DEG17_3] & RREG[DEG17_2] &
                    RREG[DEG17_1] & RREG[DEG17_0];
    assign W3GATE = W2GATE ^ X2GATE;
    assign W4GATE = W3GATE ^ (RREG[DEG4_3] & RREG[DEG4_2] &
```

```
                    RREG[DEG4_1] &  RREG[DEG4_0]);

assign W5GATE = (((((((((( ((((((((((  ((((((((((
                    ((((((((((  ((((((((((  ((((((((((
                    ((( W4GATE ^ (RREG[62+63] & RREG[BP_62])) ^
                                (RREG[61+63] & RREG[BP_61])) ^
                                (RREG[60+63] & RREG[BP_60])) ^
                                (RREG[59+63] & RREG[BP_59])) ^
                                (RREG[58+63] & RREG[BP_58])) ^
                                (RREG[57+63] & RREG[BP_57])) ^
                                (RREG[56+63] & RREG[BP_56])) ^
                                (RREG[55+63] & RREG[BP_55])) ^
                                (RREG[54+63] & RREG[BP_54])) ^
                                (RREG[53+63] & RREG[BP_53])) ^
                                (RREG[52+63] & RREG[BP_52])) ^
                                (RREG[51+63] & RREG[BP_51])) ^
                                (RREG[50+63] & RREG[BP_50])) ^
                                (RREG[49+63] & RREG[BP_49])) ^
                                (RREG[48+63] & RREG[BP_48])) ^
                                (RREG[47+63] & RREG[BP_47])) ^
                                (RREG[46+63] & RREG[BP_46])) ^
                                (RREG[45+63] & RREG[BP_45])) ^
                                (RREG[44+63] & RREG[BP_44])) ^
                                (RREG[43+63] & RREG[BP_43])) ^
                                (RREG[42+63] & RREG[BP_42])) ^
                                (RREG[41+63] & RREG[BP_41])) ^
                                (RREG[40+63] & RREG[BP_40])) ^
                                (RREG[39+63] & RREG[BP_39])) ^
                                (RREG[38+63] & RREG[BP_38])) ^
                                (RREG[37+63] & RREG[BP_37])) ^
                                (RREG[36+63] & RREG[BP_36])) ^
                                (RREG[35+63] & RREG[BP_35])) ^
                                (RREG[34+63] & RREG[BP_34])) ^
                                (RREG[33+63] & RREG[BP_33])) ^
                                (RREG[32+63] & RREG[BP_32])) ^
                                (RREG[31+63] & RREG[BP_31])) ^
```

```
                                    (RREG[30+63] & RREG[BP_30])) ^
                                    (RREG[29+63] & RREG[BP_29])) ^
                                    (RREG[28+63] & RREG[BP_28])) ^
                                    (RREG[27+63] & RREG[BP_27])) ^
                                    (RREG[26+63] & RREG[BP_26])) ^
                                    (RREG[25+63] & RREG[BP_25])) ^
                                    (RREG[24+63] & RREG[BP_24])) ^
                                    (RREG[23+63] & RREG[BP_23])) ^
                                    (RREG[22+63] & RREG[BP_22])) ^
                                    (RREG[21+63] & RREG[BP_21])) ^
                                    (RREG[20+63] & RREG[BP_20])) ^
                                    (RREG[19+63] & RREG[BP_19])) ^
                                    (RREG[18+63] & RREG[BP_18])) ^
                                    (RREG[17+63] & RREG[BP_17])) ^
                                    (RREG[16+63] & RREG[BP_16])) ^
                                    (RREG[15+63] & RREG[BP_15])) ^
                                    (RREG[14+63] & RREG[BP_14])) ^
                                    (RREG[13+63] & RREG[BP_13])) ^
                                    (RREG[12+63] & RREG[BP_12])) ^
                                    (RREG[11+63] & RREG[BP_11])) ^
                                    (RREG[10+63] & RREG[BP_10])) ^
                                    (RREG[9+63] & RREG[BP_9])) ^
                                    (RREG[8+63] & RREG[BP_8])) ^
                                    (RREG[7+63] & RREG[BP_7])) ^
                                    (RREG[6+63] & RREG[BP_6])) ^
                                    (RREG[5+63] & RREG[BP_5])) ^
                                    (RREG[4+63] & RREG[BP_4])) ^
                                    (RREG[3+63] & RREG[BP_3])) ^
                                    (RREG[2+63] & RREG[BP_2])) ^
                                    (RREG[1+63] & RREG[BP_1])) ^
                                    (RREG[0+63] & RREG[BP_0]));
――― 途中省略 ―――
    always @(posedge CLOCK or negedge nRESET) begin
        if (nRESET == 1'b0) RREG[0] <= 32'h00000000;
        else if ((RADR == 8'd0) && (RWRITE == 1'b1))
            RREG[0] <= RDIN;
```

```
        else if (CKFLAG == 1'b1)
            RREG[0] <= RREG[127] & C0INI;
    end

    always @(posedge CLOCK or negedge nRESET) begin
        if (nRESET == 1'b0) RREG[1] <= 32'h00000000;
        else if ((RADR == 8'd1) && (RWRITE == 1'b1))
            RREG[1] <= RDIN;
        else if (CKFLAG == 1'b1)
            RREG[1] <= RREG[0] ^ (RREG[127] & C1INI);
    end

―――　以下省略　―――
―――　ここまで　―――
```

「List-５」　：　「Exemplar.log」から抜粋して記載します。

―――　ここから始まり　―――

```
LeonardoSpectrum Level 1 Altera - v1999.1j (build 6.107, compiled Mar 26 2000 at 22:12:54)
Copyright 1990-1999 Exemplar Logic, Inc.   All rights reserved.
        Pass      LCs Delay DFFs TRIs   PIs POs      --CPU--
                                                     min:sec
        1         11915       23 4260      0     77   33         12:02
Using default wire table: apex20e_default
-- Start timing optimization for design .work.CIPHER.INTERFACE
No critical paths to optimize at this level

************************************************
Device Utilization for EP20K600EBC652
************************************************
Resource                  Used      Avail     Utilization
------------------------------------------------
IOs                       110       652         16.87%
LCs                       11883     24320       48.86%
Memory Bits               0         311296      0.00%

------------------------------------------------
                          Clock Frequency Report

          Clock                    : Frequency
        ------------------------------------

          CLOCK                    : 58.1 MHz

                          Critical Path Report

There are no paths that violate user specified options or constraints
```

―――　ここまで　―――

「List-6」　：　「Exemplar.log」から抜粋して記載します。

———　ここから始まり　———

```
LeonardoSpectrum Level 1 Altera - v1999.1j (build 6.107, compiled Mar 26 2000 at 22:12:54)
Copyright 1990-1999 Exemplar Logic, Inc.   All rights reserved.

        Pass      LCs Delay DFFs TRIs   PIs POs      --CPU--
                                                     min:sec
        1         16176       24 8225      0    76   33        09:50
Using default wire table: apex20e_default
-- Start timing optimization for design .work.CIPHER.INTERFACE
No critical paths to optimize at this level

*************************************************
Device Utilization for EP20K600EBC652
*************************************************
Resource                  Used      Avail     Utilization
-------------------------------------------------
IOs                       109       652         16.72%
LCs                       16144     24320       66.38%
Memory Bits               0         311296      0.00%

-------------------------------------------------
                          Clock Frequency Report

          Clock                    : Frequency
        -----------------------------------

          CLOCK                    : 45.2 MHz

                          Critical Path Report

There are no paths that violate user specified options or constraints
```

———　ここまで　———