琉球大学ＨＤＬデザイン・コンテスト2000結果報告

琉球大学HDLデザイン・コンテスト２０００結果報告

和田　知久*、翁長　健治**、宮城 隼夫*、吉田 たけお*、尾知 博***

University of the Ryukyus HDL Design Contest 2000 Summary

Tomohisa Wada*, Kenji Onaga**, Hayao Miyagi*, Takeo Yoshida*, and Hiroshi Ochi***

Abstract

Universities nationwide gathered in Okinawa, Japan, on March 3rd to participate in University of the Ryukyus HDL Design Contest 2000 hosted by the University of the Ryukyus' Department of Information Engineering. Of the 15 applicants, ten finalists were chosen for the contest with their original designs on Galois Field matrix multiplier for Redundant Array of Inexpensive Disks System. A proud and triumphant team of juniors from the Department of Information Engineering at the University of the Ryukyus (the team "Turtlenecks") was awarded first prize, defeating the more experienced Masters students from other universities.

Key Words: HDL, VHDL, Verilog HDL, Design Contest

１．コンテスト概要

琉球大学では、日本シノプシス社のサポートによりハードウエア記述言語（VHDLまたはVerilog HDL）を用いたデザイン・コンテストを開催しました．コンテストは今年度が３回目であり、今年度より琉球大学の学生だけでなく、他大学や高等専門学校の学生も参加できるオープンなコンテストにしました．オープン化初年度の本年度はホームページ[1]によるデザイン・コンテストのPRと主に口コミでの宣伝を行い、２月18日の締切りまでに１５チーム（３０名）の学生達の参加がありました．事前選考の結果、京大・阪大・九工大・豊田工業高等専門学校より６チームの代表を沖縄に招待し、琉大の４チームと合わせて、計10チームによる発表会を2000年３月３日に琉球大学工学部にて行いました[2]．10チームを学部・高専部門と修士部門に分けて、それぞれの部門の２チームに「めっちゃはやいで賞」（動作速度が高速）と「ちっちゃいで賞」（回路規模が小さい）を授与し副賞として各メンバーに３万円相当の賞品を授与しました．また、全10チームから最優秀チームを選び、「最優秀賞（シノプシス大賞）」を授与し副賞として各メンバーに５万円相当の賞品を授与しました．また参加者全員にオリジナルTシャツ（図１）を授与しました．

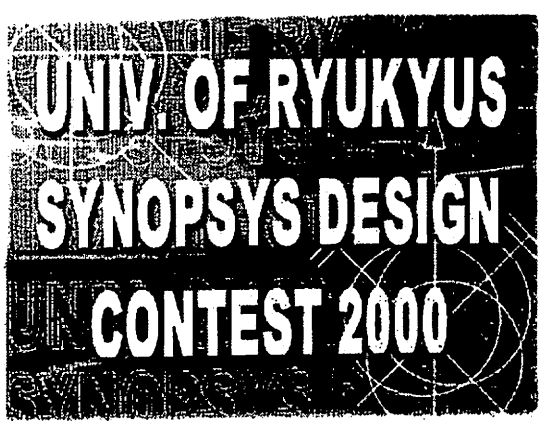

図１　Tシャツ・デザイン

受理：2000年６月26日
*工学部情報工学科
(Dept. of Information Eng., Fac. Of Eng.)
**デジタル社会総合研究所
(Digital Society Research Institite)
***九州工業大学電子情報工学科
(Dept. of Computer Science and Electronics, Kyusyu Institute of Technology)
電子情報通信学会集積回路研究会において平成12年５月発表済み

2．コンテストの趣旨

SOC(システム・オン・チップ)と言われる大規模な集積回路の実現が可能な今、小指の爪のようなシリコン・チップに何千万ゲートからなる大規模なシステムが集積可能です．マルチメディア・アイランドを目指す沖縄県にとっては，マルチメディアを支えるKEYデバイスである．そのような超LSIシステム設計のできる人材育成に力を入れ、VHDLを用いたデジタル設計を通常の学部の講義として提供しています[3]．コンテストの目的は学生達にやや難易度と規模のある課題を与えて、デザイン能力を試すチャンスを与えることと同時に、他の大学や高専の学生達と交流し視野を広げることにあります．

3．コンテストの設計課題／ポイント

今回の設計課題は「リードソロモンCODEC用のガロア体での行列乗算器の設計」です．リードソロモンはコンパクト・ディスク（CD）やデジタル通信・放送のエラー訂正、RAID（Redundant Array of Inexpensive Disks）のような磁気ディスクの信頼性向上等の用途に用いられており、ガロア体という特殊な代数での計算が必要です．今回、データとしては4ビットの数値を取り扱い、アーキテクチャと回路合成後の回路規模やクロックサイクルタイム、処理に必要なサイクル数、そして設計中のオリジナリティや工夫したポイントで設計結果を競いました．

デジタル設計として有名なものにマイクロプロセッサがありますが、マイクロプロセッサ設計では設計対象とアプリケーションの間にソフトウエアが介在し、デジタル設計とアプリケーションの関係は希薄です．SOCという時代では、アルゴリズムをチップ化（H/W化）するということが容易になり、重要になるということもあり今回は通信等でポピュラーなリードソロモンという方式をH/W化するという課題を選びました．

リードソロモンCODECと名前は難しそうですが、今回設計を行った演算器は次式に示される3入力IN1, IN2, IN3と3×3の行列の行列乗算器で、各変数は4ビット幅です．

$$\begin{bmatrix} OUT1 \\ OUT2 \\ OUT3 \end{bmatrix} = \begin{bmatrix} ELE(0) & ELE(1) & ELE(2) \\ ELE(3) & ELE(4) & ELE(5) \\ ELE(6) & ELE(7) & ELE(8) \end{bmatrix} \begin{bmatrix} IN1 \\ IN2 \\ IN3 \end{bmatrix} \quad \cdots \quad (1)$$

演算器には9つのアドレスを持つレジスタ・ファイルELEが必要であり、3×3の行列要素をあらかじめ書き込み保存しておきます．式（1）から加算と乗算が必要とわかりますが、ガロア体で演算を行うので、ガロア体の加算はビットごとの排他的論理和となり、乗算はガロア体での特殊な乗算となります．したがって、ガロア体乗算器の設計がひとつの設計のキーポイントとなります．

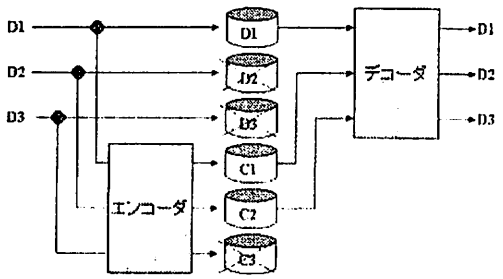

図2　RAIDシステム

この演算器をRAID（Redundant Array of Inexpensive Disks）というデータを記憶するハード・ディスクに冗長なチェック・サムを記憶するハード・ディスクを追加して信頼性を向上した記憶システム（図2）を想定し動作検証を行います．データ入力D1, D2, D3から式(2)にしたがってチェックサムC1, C2, C3を生成し、D1, D2, D3, C1, C2, C3をハード・ディスクに記憶します．

$$\begin{bmatrix} C1 \\ C2 \\ C3 \end{bmatrix} = \begin{bmatrix} 1^0 & 2^0 & 3^0 \\ 1^1 & 2^1 & 3^1 \\ 1^2 & 2^2 & 3^2 \end{bmatrix} \begin{bmatrix} D1 \\ D2 \\ D3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 5 \end{bmatrix} \begin{bmatrix} D1 \\ D2 \\ D3 \end{bmatrix} \quad \cdots \quad (2)$$

$$\begin{bmatrix} D1 \\ D2 \\ D3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} D1 \\ C1 \\ C2 \end{bmatrix} \quad \cdots \quad (3)$$

図2のようにD2, D3, C3のディスクが故障しデータが読み出せなくなった場合には、式(3)によりD1, C1, C2からオリジナルデータであるD1, D2, D3を再生成することができます．(2)式の3×3の行列はVandermonde行列であり、チェックサムの生成に用いられ、(3)式の3×3の行列はハード・ディスクの故障状態により決定される行列です．したがって、同一の演算器を2個用いて、異なった行列要素をあらかじめ設定することで、エンコーダとデコーダを実現し、RAID（図2）の入力データと出力データが同じであることをチェックすることで検証を行います．

表1　発表チームの設計のまとめ

	学校名 チーム名	サイクル数	サイクル時間	トータル時間	面積	特徴
1	琉大2年 RU07	3	9.24	27.27	2739	gflog 6 個 gfilog 3 個 Adder 3 個
2	琉大2年 RU09	4	21.9	87.60	2220	gflog 4 個 gfilog 3 個 Adder 3 個
3	豊田高専 仲野研C	2	7.35	14.70	1203	簡略テーブルによる乗算
4	琉大3年 タートルネックス	1	3.40	3.40	1205	素直な乗算器9個
5	琉大3年 和田研	9	7.31	65.79	743	素直な乗算器1個
6	京大 くまたいよう	3	9.22	27.66	1220	gflog 6 個 gfilog 3 個 Adder 3 個
7	九工大 ホークス	7	5.89	17.67	1521	gflog 6 個 gfilog 3 個 Adder 3 個
8	阪大 谷口研	4	14.54	58.16		gflog 18 個 gfilog 9 個 Adder 9 個
9	京大 ちょビット	1	2.40	2.40	2413	素直な乗算器9個
10	阪大 白川研	11	6.87	75.24	966	gflog 1 個 gfilog 1 個 Adder 1 個

4．参加各チームの設計結果レビュー

4－1．各チームのアーキテクチャ設計の特徴

各チームの設計の結果と特徴を表1に示します．最も単純な構成は（1）式である行列演算を同時に行う方式で、回路規模は大きくなりますが約1サイクルで計算を完了できます．図3はガロア行列演算の出力をレジスタで受けて2サイクルで動作する構成例です．図の下方には9つの行列要素 ELE を保存するレジスタ・ファイルがあります．発表を行った10チーム中、豊田高専「仲野研 C」チーム、琉大3年「タートルネックス」チーム、阪大「谷口研」チーム、京大「ちょビット」チームがこのような構成でスピード重視の設計を行いました．

琉大2年「RU07」チーム、「RU09」チーム、京大「くまたいよう」チーム、九工大「ホークス」チームはガロア乗算器を3個用いて、OUT1, OUT2, OUT3 を順次計算する約3サイクルで計算を実行する方式を採用し、スピードと面積とも中庸の値を目指しました．

琉大3年「和田研」チーム、阪大「白川研」チームはガロア体乗算器を1つのみ用いて、約9サイクルで計算を実行する方式を採用し、面積最小を重視する設計を行いました．

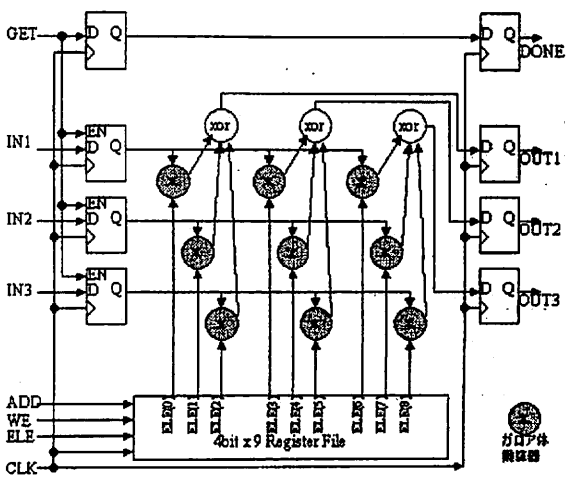

図3　2サイクル動作の全体構成の例

4－2．ガロア乗算器の構成方法

4ビットのガロア体は $\alpha^4+\alpha+1=0$（特性方程式という）を満たす α を用いて、その4ビットを係数にもつ以下のような3次の多項式に対応します．

$$a_3\cdot\alpha^3+a_2\cdot\alpha^2+a_1\cdot\alpha^1+a_0 \quad \cdots \quad (4)$$

したがって、4ビット a_3,a_2,a_1,a_0 のガロア体の乗算は（4）式で示される多項式の乗算となります．設計の中心課題のあるガロア体の乗算器は以下の3種類の方法で実現されました．

（1）「GFLOG/GFILOG を用いる方法」
（2）「素直に多項式の乗算をする方法」
（3）「演算結果の表を用いる方法」

ガロア乗算器として、GFLOG/GFILOG を用いる方法を用いたチームの幾つかは、ガロア乗算器が gflog、加算、gfilog 等の幾つかの処理ステージに分かれていることを利用して、gflog の共通化等を行って回路面積削減を行っています．ガロア体乗算には2つの gflog が必要ですが、

ガロア乗算器の1つの入力はレジスタ・ファイルであるので、レジスタ・ファイルに行列要素を書き込む再にgflog にて変換された値をレジスタ・ファイルに保持するように設計し、ガロア乗算器からgflog 回路を減らしています.

結果からみると、GFLOG/GFILOG を用いる方法ではなく、素直に多項式の乗算をする方法をもちいた方が回路面積的にも、スピード的にも有利になっていました.

参加者の中には、設計した演算器を実際にFPGA に実現し、画像データに対して、チェックサムに対応する画像を生成し、そのチェックサム画像から源画像を復元するデモをおこなうような面白いレポートもありました.

今回のデザイン・コンテストでは簡単のために、取り扱うデータを4ビット幅と小さいものにしています. したがって、厳密には2^4のガロア拡大体$GF(2^4)$を取り扱います.

表2　$GF(2^4)$の2進数と多項式の関係

2進数	対応する多項式	10進数
0000	0	0
0001	$1=\alpha^0$	1
0010	$\alpha=\alpha^1$	2
0011	$\alpha+1=\alpha^4$	3
0100	α^2	4
0101	$\alpha^2+1=\alpha^8$	5
0110	$\alpha^2+\alpha=\alpha^5$	6
0111	$\alpha^2+\alpha+1=\alpha^{10}$	7
1000	α^3	8
1001	$\alpha^3+1=\alpha^{14}$	9
1010	$\alpha^3+\alpha=\alpha^9$	10
1011	$\alpha^3+\alpha+1=\alpha^7$	11
1100	$\alpha^3+\alpha^2=\alpha^6$	12
1101	$\alpha^3+\alpha^2+1=\alpha^{13}$	13
1110	$\alpha^3+\alpha^2+\alpha=\alpha^{11}$	14
1111	$\alpha^3+\alpha^2+\alpha+1=\alpha^{12}$	15

4－3．GFLOG/GFILOG を用いる方法

この方法はデザイン・コンテストの参考文献としたJames S. Plank の論文[4]で解説されている方法で、多くのチームがこの方式に従って乗算器を構成しました.

表2に示すように4ビットの2進数はそれぞれに多項式に対応します. 今、特性方程式$\alpha^4+\alpha+1=0$が成立するので、この関係を用いて多項式を変形すると、2進数"0000"を除いてすべての多項式は$\alpha^n,(0\le n\le 14)$という形式に変形できます.

したがって、2つの多項式の乗算は以下のように指数の通常の加算に変換することができます.

$$(a_3\cdot\alpha^3+a_2\cdot\alpha^2+a_1\cdot\alpha^1+a_0)$$
$$\times(b_3\cdot\alpha^3+b_2\cdot\alpha^2+b_1\cdot\alpha^1+b_0)$$
$$=\alpha^{gf\log(a)}\cdot\alpha^{gf\log(b)}=\alpha^{gf\log(a)+gf\log(b)}\quad\cdots\quad(5)$$

また、特性方程式$\alpha^4+\alpha+1=0$より$\alpha^{15}=1$であるので、gflog(a)+gflog(b)はmod15（15で割った余り）を取ることにより、0から14の整数に丸めることができます. この値をgflog の逆関数gfilog で変換すれば、ガロア乗算の結果の4ビットの2進数を得ることができます.

この方法では2進数"0000"に対応する指数は存在しませんので（あえて言えば負の無限大）、乗算入力の少なくとも一方が2進数"0000"ならば、出力を2進数"0000"にする必要があります.

図4にこの方式のブロック図を示します. gflog は4ビット入力、4ビット出力の組み合わせ回路となり、gfilog も同じく4ビット入力、4ビット出力の組み合わせ回路となります. 加算は通常の加算であり、mod15は加算結果が15を超えた場合に15の減算を行う回路です. Zero?ブロックは入力の少なくとも1つの2進数"0000"入力を検知し、その場合は出力を強制的に2進数"0000"とします.

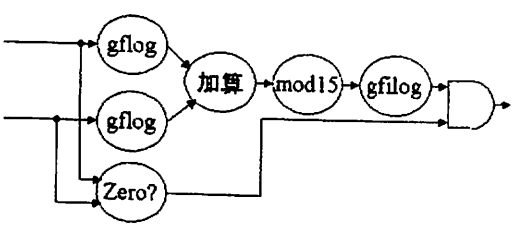

図4　gflog/gfilog を用いたガロア乗算器

4－4．素直に多項式の乗算をする方法

この方式はガロア体の性質に戻って素直に計算する方式です．デザイン・コンテストでは3つのチームがこの計算方法を使用して、結果的によりシンプルで回路規模の小さい乗算器を実現しました．2つの多項式を乗算すると以下のようなります．

$$(a_3\cdot\alpha^3+a_2\cdot\alpha^2+a_1\cdot\alpha^1+a_0)\cdot(b_3\cdot\alpha^3+b_2\cdot\alpha^2+b_1\cdot\alpha^1+b_0)$$
$$=(a_3\cdot b_3)\alpha^6+(a_3\cdot b_2\oplus a_2\cdot b_3)\alpha^5$$
$$+(a_3\cdot b_1\oplus a_2\cdot b_2\oplus a_1\cdot b_3)\alpha^4$$
$$+(a_3\cdot b_0\oplus a_2\cdot b_1\oplus a_1\cdot b_2\oplus a_0\cdot b_3)\alpha^3$$
$$+(a_2\cdot b_0\oplus a_1\cdot b_1\oplus a_0\cdot b_2)\alpha^2$$
$$+(a_1\cdot b_0\oplus a_0\cdot b_1)\alpha+a_0\cdot b_0 \quad \cdots \quad (6)$$

特性方程式$\alpha^4+\alpha+1=0$が成立するので、式（7a, 7b, 7c）が成立します．

$$\alpha^6=\alpha^3+\alpha^2 \quad \cdots \quad (7a)$$
$$\alpha^5=\alpha^2+\alpha \quad \cdots \quad (7b)$$
$$\alpha^4=\alpha+1 \quad \cdots \quad (7c)$$

ここで、(7a, 7b, 7c) 式を用いて（6）式のα^4以上の項を3次以下の多項式に変形すると以下の (8) 式のようになります．

$$(a_3\cdot\alpha^3+a_2\cdot\alpha^2+a_1\cdot\alpha^1+a_0)\cdot(b_3\cdot\alpha^3+b_2\cdot\alpha^2+b_1\cdot\alpha^1+b_0)$$
$$=(a_3\cdot b_0\oplus a_2\cdot b_1\oplus a_1\cdot b_2\oplus a_0\cdot b_3\oplus a_3\cdot b_3)\alpha^3$$
$$+(a_2\cdot b_0\oplus a_1\cdot b_1\oplus a_0\cdot b_2\oplus a_3\cdot b_3\oplus a_3\cdot b_2\oplus a_2\cdot b_3)\alpha^2$$
$$+(a_1\cdot b_0\oplus a_0\cdot b_1\oplus a_3\cdot b_2\oplus a_2\cdot b_3\oplus a_3\cdot b_1\oplus a_2\cdot b_2\oplus a_1\cdot b_3)\alpha$$
$$+a_0\cdot b_0\oplus a_3\cdot b_1\oplus a_2\cdot b_2\oplus a_1\cdot b_3 \quad \cdots \quad (8)$$

したがって、（8）式の各係数を計算することでガロア体の乗算器が構成できます．結果的に、AND ゲート16個と XOR ゲート18でガロア体乗算器を構成することができました．

5．課題にとらわれない工夫／発想

京大「ちょビット」チームは、設計した演算器を実際にFPGA でPCI デバイスとして実現し、画像データに対して、チェックサムに対応する画像を生成し、そのチェックサム画像から源画像を復元するデモを紹介し注目を集めていました（図5）．

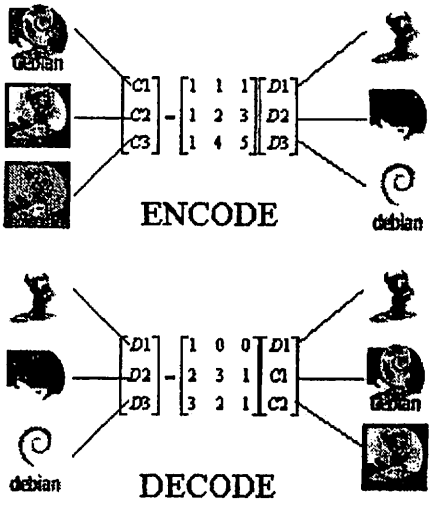

図5　「ちょビット」チームの画像デモ

阪大「白川研」チームは小面積を狙い11サイクルという長めの処理レイテンシではあるが、パイプライン処理を行い9サイクルごとにデータの処理をできるように実現しました．

琉大3年「和田研」チームは構成部品ごとの回路合成後の回路規模を紹介し、種々の構成を検討した中での最適解を求めたことを紹介しました．琉大3年「タートルネックス」チームはレジスタファイルなどにシフトレジスタを適用することで、回路規模を削減できることを紹介し注目を浴びていました．

豊田高専「仲野研C」チームは（3）「演算結果の表を用いる方法」でガロア演算を実現する時に、（2）、（3）式の行列要素に5以下の整数値しか現れないことに着目し、演算結果の表を大幅に簡略化し小面積、高速化を実現しました．

6．コンテストのまとめ

結果的に学部・高専部門の琉大3年「和田研」チームと修士部門の阪大「白川研」チームが「ちっちゃいで賞」（小面積）を獲得し、学部・高専部門で豊田高専「仲野研C」チームと修士部門の京大「ちょビット」チームが「めっちゃはやいで賞」（高速・高性能）を獲得し、「最優秀賞（シノプシス大賞）」を琉大3年「タートルネックス」チームが他大学の修士学生らを抑えて獲得しました．

今回の課題はガロア体という聞きなれない代数を実現する必要があり、設計経験の少ない学生にはやや困難な

内容であったようです．言い換えれば、HDLによる設計以前の仕様書理解の障壁が高かったように感じます．しかしながら、学部2年生でも設計を完了したチームも多数あり、コンテスト課題としてはやりがいのある内容であったと思います．このような困難な内容を実現した学生の満足感を発表から感じることができ、次年度も「ぜひ参加したい」という意見を多数の学生から聞くことができました．

発表会で、普段アナログ回路設計を主にやっている阪大「谷口研」チームの学生より，合成ツールの凄さやデジタル設計の面白さの感想も聞くことができ、彼らにとってこのコンテストが沖縄旅行だけでなく、技術の幅を広げる非常に有意義なものになったように思います．

自分達と異なったアルゴリズムで実現したチームの発表、徹夜して面積や性能を最適化した年下の学生のがんばり、経験ある修士学生らのパイプライン等の高度な設計内容や発表会でのプレゼンテーションの上手さ、そして他大学／高専との交流で、学生同士で大きな刺激を受けたようです．

研究会等では異なる各発表者が異なる課題について発表しますが、このような同一の課題を異なる大学／高専の学生が一堂に会して発表会を持つことは、学生同士の大きな刺激となり非常に有効な機会でした（図6）．

図6　発表者と審査員一同

7. 次回コンテストの予定

琉大情報工学科主催のデザイン・コンテストは次年度以降もオープンな形で継続してゆく予定であり、次回はもう少し時期的に早い時点で広報活動も積極的に行い参加者数を増やしてゆく予定です．優秀チームの代表者を琉球大学へ招待し、本年同様に参加者全員にオリジナルTシャツを、優勝・準優勝者に豪華賞品を出す予定であり、HDLによるデザインに興味ある学生諸君の参加を期待しています．課題等はまだ検討中であり、良いヒントがあればぜひ私のところまで教えて頂ければ幸いです[5]．

琉球大学のある沖縄は本土から感覚的に遠く、このようなチャンスで学生・教官の交流を深めることができ、非常に有意義なイベントと考えています．沖縄に遊びに来たいという不純な動機も含めて歓迎しますので、ドシドシ参加をお願いします．

なお、このコンテストは日本シノプシス社からの奨学寄付金でサポートされており、このような機会を与えて頂いた日本シノプシス社にこの場を借りて深く感謝いたします．

参考文献

[1]http://bw-www.ie.u-ryukyu.ac.jp/~wada/design/contest99.html
[2]http://bw-www.ie.u-ryukyu.ac.jp/~wada/design/Schedule.html
[3]翁長,宮城,和田,吉田,尾知,"沖縄県マルチメディア・アイランド構想と琉球大学 SOC 設計教育"、信学会総合大会 A-3-21, p.89, 2000.3
[4]James S. Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems," Department of Computer Science, University of Tennessee, February 19, 1999
[5] wada@ie.u-ryukyu.ac.jp