卒業論文

題目

ビームスポット列画像エンコーダの開発

指導教員

近藤 利夫 教授

2014年

三重大学 工学部 情報工学科 計算機アーキテクチャ研究室

佐藤 徹 (410820)

内容梗概

レーザープリンタの製造において,治具を用いることによりレーザー ビームの微調整作業の効率化が期待されている.

当研究室で開発の支援を行っている全領域治具は,レーザーの照射状況 を18個の単眼センサで画像データとしてキャプチャし,そのキャプチャ 画像をまとめて1台のコンピュータに転送し,そこで画像の解析を行う. キャプチャ画像が大量で,既存の高速転送手段の上限速度を超えるため, リアルタイムでキャプチャ画像を圧縮伸張ができるコーデックシステム の実現が要求される.

そこで,本研究では,センサから出力された画像データをコンピュー タに圧縮転送するハードウェアエンコーダを,ハードウェア規模の低減 やアルゴリズムの単純化を考慮し,FPGA ベースで設計した.

コンピュータ側のデコードの負担を低減するためには,複数のセンサ から同時に入力される画像信号を FPGA ボードで並列にエンコードした 上で,それらの圧縮データを別々に出力しなければならない.しかし,出 力時にセンサ毎の個々の圧縮データの連続出力が容易にできるように,セ ンサ毎に別個の格納領域を割り当てようとすると,SDRAM のロウアド レスが頻繁に変化してしまい,読み書きオーバーヘッドが大きくなると いう問題がある.データの格納領域をセンサ毎に割り当てず,同一セン サの隣接データ距離を一定の短い間隔にし,シーケンシャルアクセスで 書き込みを行うことにより,書き込みと読み出し時のオーバーヘッドを 低減した.

Abstract

In the laser printer production line, a jig capturing whole irradiated area is required for efficient fine adjustment of the laser beam focus.

In this jig system, the whole irradiated area is captured by 18 monocularvision sensors, and the captured image data are transferred to a single computer for laser forcusing analysis.

Since the capturing image data rate is too high, compression and decompression are required for real-time transfer from the jig to the computer.

The purpose of this study was to design hardware encoder including simple algorithm and reduction of hardware size.

In order to reduce a burden of decode on the computer, the sequence of input data needs to be classified according to the signal of each sensor on the FPGA board.

If row address of SDRAM changes frequently, read/write overhead increases. To solve this problem, we determined the short distance between the adjacent data of the same sensor on the SDRAM. As a result, read/write overhead has been reduced.

目 次

1	まえ	がき	1
	1.1	研究背景...........................	1
	1.2	研究目的.............................	1
2	対象	マ画像の概要とエンコーダの必要性	2
	2.1	レーザープリンタのレーザービームの重要性......	2
	2.2	全領域測定治具	4
	2.3	符号化対象画像	4
	2.4	転送データ量の見積と画像圧縮の必要性 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
3	圧縮	『アルゴリズムとエンコーダの要求条件	6
	3.1	圧縮アルゴリズムの決定	6
	3.2	圧縮アルゴリズム......................	7
	3.3	エンコーダの必要条件	10
4	専用	ヨハードウェアの必要性	10
5	設計	上の条件	11
	5.1	FPGA ボードのスペック	11
	5.2	ハードウェアエンコーダの構成	12
	5.3	データ入出力形式	12
	5.4	データの流れ	14
	5.5	モジュールの動作周波数	14
6	設計	上の課題とその解決	15
	6.1	エンコーダ部	15
		6.1.1 符号化方式の決定	15
		6.1.2 24 ビット単位にデータをまとめる	16
		6.1.3 フレーム末尾の例外処理	18
	6.2	データ入出力部	19
		6.2.1 バッファの 設置と容量の決定	19
		6.2.2 データ転送スケジュールの提案	19
		6.2.3 SDRAM 読み書きオーバーヘッド低減	21
		• • • • • • • • • • •	20
		6.2.4 コンピュータ側での出力データ識別	26

7	評価	実験と考察	28
	7.1	実験の条件と評価内容・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	28
	7.2	バッファの容量決定	28
	7.3	評価結果	30
8	エン	コーダ部搭載数の増加	32
9	あと	がき	34
謝	锌		34
参	考文南	τ	35

図目次

2.1	レーザープリンタの動作原理 [1]
2.2	露光[1]
2.3	転写[1]
2.4	定着 [1]
2.5	全領域治具のブロック図
2.6	圧縮対象画像 5
3.7	差分符号化
3.8	ランレングス符号化6
3.9	差分符号化データ 形式7
3.10	ランレングス符号化データ形式7
3.11	圧縮アルゴリズム
4.12	圧縮伸張時間
5.13	ハードウェアエンコーダの構成 12
5.14	データ入力形式 13
5.15	データ出力形式 13
5.16	データ転送スケジュール
6.17	圧縮行程の分割 16
6.18	符号化方式決定ハードウェア
6.19	24 ビット単位のデータ生成手順 17
6.20	データセットの 入力データ形式
6.21	frameEnd 信号 18
6.22	バッファの配置
6.23	データ転送スケジューリング
6.24	エンコーダ番号付与 21
6.25	データ分割 22
6.26	書き込みアルゴリズム 24
6.27	読み出しアルゴリズム 25
6.28	B1 から B2 ヘデータ転送するハードウェア 26
6.29	転送プロトコル
7.30	バッファ B3 の使用状況
7.31	手法1改良後 29
7.32	手法 2
7.33	手法1改良前に対する相対割合

表目次

3.1	差分符号化のパターン8	3
5.2	DE0NANO のスペック 11	L
5.3	入力信号	3
5.4	出力信号	1
5.5	動作周波数 15	5
6.6	オフセットと rnum の関連 22	2
6.7	比較照合結果	7
7.8	手法1改良前のバッファの最大使用量と最適化29)
7.9	手法2のバッファの最大使用量と最適化 30)
7.10	評価結果	L
7.11	消費電力内訳 31	L
8.12	手法3のバッファの最大使用量と最適化 33	3
8.13	手法1改良後と手法3の比較33	3

1 まえがき

1.1 研究背景

レーザープリンタの印刷精度は感光ドラムへのレーザーの照射状況に 依存することから,プリンタの製造においてレーザービームの強度・ピ ント・角度等の微調整に多大な時間を要している.そのため,照射状況 を画像としてリアルタイムにキャプチャし,その画像をコンピュータを 用いて詳細に解析することで微調整作業の効率を向上させる治具の開発 が急務となっている.

1.2 研究目的

レーザーの照射状況を解析する治具は,レーザー光照射領域のキャプ チャ用に単眼センサを用いる.1つのセンサでキャプチャできる空間領域 は狭いため,18個のセンサを用いて対象領域の全体をキャプチャする必 要がある.また,ハードウェアの維持管理の手間と費用の削減のため,解 析作業は1台のコンピュータで済ませるのが望ましい.しかし,18個の センサがキャプチャする画像のデータ量は,既存の転送手段で転送する には大量すぎる.このため,短時間で画像の圧縮伸張ができるシステム の実現が必須である.

そこで,本研究では,センサから出力された画像データをコンピュータ に圧縮転送するハードウェアエンコーダを設計する.複雑で大きなハー ドウェアを構成してしまうと電力の増大や拡張性の低下につながるため, ハードウェア規模の低減やアルゴリズムの単純化がシステム設計上の課 題となる.

2 対象画像の概要とエンコーダの必要性

2.1 レーザープリンタのレーザービームの重要性

レーザープリンタでの印刷行程には,帯電・露光・現像・転写・定着・清掃と いう6つの行程が存在する.

レーザープリンタの動作原理を図 2.1 に示す.



図 2.1: レーザープリンタの動作原理 [1]

帯電はトナーをドラムに付着させるために感光ドラムの表面全体に均一に帯 電させる行程である.露光はレーザービームで印刷パターンを帯電した感光ド ラムに照射し電気的潜在画像を感光ドラム表面に形成する行程である(図 2.2).



図 2.2: 露光 [1]

現像は,トナーを攪拌しながら感光ドラム表面に形成された潜在画像に付着 させる行程である(図 2.3).転写と定着の行程は,トナーが付着したドラム表面



図 2.3: 転写 [1]

のそばを用紙が通過するときに感光ドラム上のトナーを用紙に転写し定着させる行程である(図 2.4).



図 2.4: 定着 [1]

ただし、ドラムに付着したトナー全てが用紙に付着することはなく、一部の トナーは転写されずにドラムに付着したままになる.そのため、次の印刷に備 えて付着したトナーをドラムから除去する行程が清掃である.

以上のことから,印刷精度は印刷パターンを決めるレーザー光の状態によって 左右され,レーザープリンタ製造時においてレーザー光の強度・ピント・角度を 微調整することは重要である.しかし,現在はレーザービームの微調整は行わ れているものの高精度かつ効率的に実施されていないため,レーザーの微調整 を効率的に行うための治具の導入が求められている.

2.2 全領域測定治具

全領域治具とは,レーザービームの状態を画像としてキャプチャし解析する 装置である.レーザーの照射状況はセンサでキャプチャされ画像データに変換 されるが,1つのセンサでキャプチャできる領域は小さいため,全領域治具では 18個のセンサを用いて対象領域のレーザー光をキャプチャする.

解析はコンピュータで行うが,コンピュータの維持管理のコストと手間及び 省スペースのために1台のコンピュータで画像解析を行う. 全領域治具の概略ブロック図を図2.5に示す.



図 2.5: 全領域治具のブロック図

2.3 符号化対象画像

符号化対象画像のモデルを図 2.6 に示す.画像サイズは 1944 × 1092 であり 8 ビットグレースケールで構成され,1 つのセンサからは 30fps で画像が出力される.

画像の中央ライン付近に白色のスポット列が横方向に配置されており,この白 いスポット付近が詳細に画像解析される.

一方,背景は黒で表現されているが画素値は一定ではない.また,黒色部分は 詳細な画像解析が行われない.



図 2.6: 圧縮対象画像

2.4 転送データ量の見積と画像圧縮の必要性

非圧縮でシステムを実現しようとする場合,18個のセンサから出力されるデー タのビットレートは

$$1944 \times 8[\text{bit}] \times 30[\text{fps}] \times 18 = 9170[\text{Mbps}]$$
 (1)

になる.

一方, CameraLink 拡張ボードの転送ビットレートは

$$85[MHz] \times 8[bit] = 680[Mbps]$$
⁽²⁾

になる.

入力データのビットレートが出力データのビットレートを大きく上回っている ため,非圧縮でシステムを実現すると入力データを処理しきれずコマ落ちが発 生してしまう.

そのため画像データを少なくとも 1/13.5 に圧縮する必要がある.

3 圧縮アルゴリズムとエンコーダの要求条件

3.1 圧縮アルゴリズムの決定

画像の解析方法を考慮すると圧縮に関して以下の制約が考えられる。

- 1. 画像はスポットとその付近の領域が詳細に解析されるため,スポットとその周辺は可逆圧縮を行う
- 2. 黒色の背景は詳細な解析が行われないため,完全な可逆圧縮でなくても よい

制約条件 1. により,スポット付近には,高速な可逆圧縮と伸張速度を確保するために,DPCM 符号化で採用されている予測残差を符号化するのではなく,前後の画素の差分を符号化する差分符号化方式を採用する(図 3.7).



図 3.7: 差分符号化

制約条件 2. により, 黒色の背景は詳細な解析が行われないため圧縮効率を重 視する.ランレングス符号化は,連続して現れる画素とその個数を符号化する ため,効率よく圧縮できる.ただし,黒色背景は画素値が一定ではないため,画 素値を一定に補正する処理を施した上でランレングス符号化を行う(図 3.8).



- 画素値:19 ランレングス:6 で符号化
 - 図 3.8: ランレングス符号化

1つのスポット領域は黒色の背景に比べごくわずかであり,黒色背景に対しス ポット全体が占める割合もごくわずかである.したがって,差分符号化による 符号量過大は避けられる.

差分符号化とランレングス符号化は,前後2つの画素値を用いて符号化する ため,少ない情報量で符号化が可能である.センサは33msかけて画素をライン に沿って1つずつ出力するため,少ない情報量で符号化できるこれらの符号化 手法は画像を低遅延で圧縮することを可能にする.

これらの符号化手法を用い,符号化手法を動的に選択するソフトウェアエン コーダをC言語で実装し対象画像を圧縮したところ,データサイズは1/20以下 になり,目標とする1/13.5の圧縮能力を十分に満足している.

以上のことから,ランレングス符号化と差分符号化を適宜使い分けた圧縮ア ルゴリズムを採用する.

3.2 圧縮アルゴリズム

符号化の種別を判別するためのマーカーを符号の先頭に付与する.符号のデー タ形式を図 3.9 と 3.10 に示す.

符号[3bit]	110	符号[6bit]		111	符号[9bit]
3ビット符号化	65	ビット符号化	-		

図 3.9: 差分符号化データ形式

101	差分値[9bit]	符号[24bit]									

ランレングス符号

図 3.10: ランレングス符号化データ形式

差分値が負の値になることを避けるために差分値にオフセットを加えた値を 符号化する.ランレングス符号化のオフセットは255とする. 差分符号化は差分値に応じて,3種類の符号化パターンに分かれる. 差分符号化パターンを表3.1に示す.

表 3.1: 差分符号化のパターン

差分値	符号化パターン	適用優先順位	マーカー	オフセット	総符号長
-2~2	 3 ビット符号化 	1	なし	2	3
-31 ~ 32	6 ビット符号化	2	110	31	9
-255 ~ 255	 9 ビット符号化 	3	111	255	12

ラン長のビット数を長くすると,短い間隔でランレングス符号が発生したと きの符号が冗長になる.対象画像の場合,ラン長の最大値は12ビットで表現で きる場合が多いため,12ビットで表現できる最大の整数4095をラン長の最大値 とする.

圧縮対象画像の背景の画素値は 19以下が大半を占めているため,画素値の補 正値を 19とした.

圧縮アルゴリズムを図 3.11 に示す.



図 3.11: 圧縮アルゴリズム

3.3 エンコーダの必要条件

次のフレームの入力が開始されるまでに原画像を少なくとも 1/13.5 に圧縮し コンピュータにデータを転送及び伸張を終了しなければならない. また,システムは継続して使用されるため,小さなシステムで低消費電力と保 守管理の容易性を両立し,機能の拡張性にも優れていることが望ましい.

4 専用ハードウェアの必要性

ソフトウェアエンコーダを使用する場合,33msの間に18枚のフレームを圧縮伸張するためには,メモリ入出力オーバーヘッドを無視し圧縮と伸張にかかる時間が等しいと仮定すると,1フレームあたりの圧縮時間は

$$\frac{33[\text{ms}]}{18} \times \frac{1}{2} = 0.91[\text{ms}] \tag{3}$$

になる.しかし,実際はセンサからの並列に出力される画像データを一旦コン ピュータのメモリに展開してから圧縮する必要があるため,許容される時間は この値より小さくなる.

ソフトウェアエンコーダを使用した場合の圧縮時間は 28[ms] であるため,許容時間を大きく超えている.

0.91ms という圧縮時間は, ソフトウェアエンコーダを用い, ハイスペックマル チプロセッサによる PC ベースの構成を採用したとしても容易に実現できない レベルであり, ハードウェア規模低減の要求も満たすことはできない.

一方,八ードウェアエンコーダを用いた場合,センサからの出力データを即 座に圧縮し,コンピュータにデータを転送することができる.そのため,コン ピュータは33msの間に18枚の圧縮画像のメモリ入出力と伸張を行えばよいた め,画像1枚あたりの伸張にかけられる時間がおよそ2倍に増える(図4.12).

ハードウェアで実装されたシステムは特定の並列処理を効率よく行うことが できるため,ハードウェアエンコーダを用いれば複数のセンサから並列に入力 される画像を効率よく圧縮できる.

また, FPGA でハードウェアを構成した場合,高い拡張性の確保や開発期間の 短縮及び開発コストの低減が期待できる.

以上より, FPGA ベースのハードウェア構成は必要不可欠であり, 圧縮に特化したハードウェアエンコーダが必要である.





5 設計上の条件

5.1 FPGAボードのスペック

FPGA ボードは Terasic 社の DE0NANO である. FPGA は Altera 社製 cyclone4E が搭載されている. DE0NANO のスペック [2] を表 5.2 に示す.

R 0.2: DE(MANO OX (97
デバイス	EP4CE22F17C6
周波数	$50 \mathrm{~MHz}$
LE 数	22,320
メモリ	608,256 bit
PLL	4
SDRAM	32 MB
ピン数	154

表 5.2: DE0NANO のスペック

5.2 ハードウェアエンコーダの構成

センサは 12 ビット幅の画素データを 121.5MHz で 1 ピクセルずつ出力する. 画素データは,シリアルパラレル変換器で 4 つのピクセルが 1 セットにまとめ られる.パラレルデータは圧縮されたのち,SDRAM に一旦格納される. 格納された圧縮画像は,次のフレームの入力が開始されたら読み出される.読 み出されたデータはデータ転送順序が調整されコンピュータに入力される.本 研究では,1台の FPGA ボードに組み込むシステムを設計実装する. 設計するハードウェアエンコーダは,画像圧縮を行うエンコーダ部とデータの 入出力を制御するデータ入出力部に分割される.1台の FPGA ボードに組み込 むハードウェアエンコーダ全体を図 5.13 に示す.



図 5.13: ハードウェアエンコーダの構成

5.3 データ入出力形式

ハードウェアエンコーダに入力される信号は,4ピクセル分48ビットのデー タ信号,フレームの区切りを表す信号 vSync とデータの有効無効を表す信号 dataValid、20.25[MHz]のinclockである.なお,各センサからデータ信号,vSync, dataValid, inclockが出力されるため,これらの信号線は3セット存在する.

出力信号は,12 ビット幅の圧縮画像データ信号,フレームの区切りを表す FVAL,データの有効無効を表す DVAL,81[MHz]のclock81 である.

入力データと出力データの形式を図 5.14 と図 5.15 に示す.



図 5.14: データ入力形式

clock81				7/	<u>۲</u> -۱		1	 	\Box	<u> </u>	Γ	<u> </u>	$\overline{\Gamma}$	1-	<u> </u>	<u> </u>	 	\overline{Z}	1-		<u> </u>	.	<u> </u>	17	/	
FVAL			1 1 1		1			 			Γ	1	1		<u> </u>	1	 				1	 		1	1	
DVAL								 	, , ,		Γ		İ		<u> </u>		 	, ,						<u> </u>		
dataToCameraLink	 	Ξ	י -בע	ーダ	- 1圧約	i 個個	象					! :ン=	<u>'</u> iーダ	。 2圧	・ 縮画	r 像				_ 	' :ンコ	- 一タ	' 3圧	縮画	r 像	j

図 5.15: データ出力形式

信号名	幅	役割									
pixelDataGR1	48	センサ1の画像データ									
inclock1	1	センサ1のデータの入力周波数									
vSync1	1	センサ1のフレームの区切り									
dataValid1	1	センサ1のデータの有効無効									
pixelDataGR2	48	センサ2の画像データ									
inclock2	1	センサ2のデータの入力周波数									
vSync2	1	センサ2のフレームの区切り									
dataValid2	1	センサ2のデータの有効無効									
pixelDataGR3	48	センサ3の画像データ									
inclock3	1	センサ3のデータの入力周波数									
vSync3	1	センサ3のフレームの区切り									
dataValid3	1	センサ3のデータの有効無効									

表 5.3: 入力信号

表 5.4: 出力信号

信号名	幅	役割
dataToCameraLink	12	圧縮データ
FVAL	1	フレームの区切り
DVAL	1	データの有効無効
clock81	1	データの出力周波数

5.4 データの流れ

FPGA1 台のデータ転送スケジュールを図 5.16 に示す. 画像データは,ピクセル単位で 33ms かけてセンサから出力される.エンコーダ は受け取ったピクセルを随時符号化しバッファに転送する.圧縮画像はエンコー ダに新しいフレームの入力が開始されたら読み出される.

コンピュータはブロック単位でデータをメモリに格納するため,ハードウェアエンコーダから出力されるデータ長は固定長でなければならない.

よって,データ長を整えるためにダミーデータを符号データの直後に付与する.

vSync		<u> </u>		
pixelDataGR1	最初の画像データ]	2番目の画像データ	
pixelDataGR2	最初の画像データ	1	2番目の画像データ	
pixelDataGR3	最初の画像データ		2番目の画像データ	
dataToCameraLink			最初の圧縮データ 最初の圧縮データ 最初の圧縮データ エンコーダ番号1 エンコーダ番号2 エンコーダ番号3	

図 5.16: データ転送スケジュール

5.5 モジュールの動作周波数

シリアルパラレル変換の出力周波数が 20.25[MHz] であり, 符号化は隣接差分 値に対して行われるため, エンコーダの動作周波数は

$$20.25[MHz] \times 4 = 81[MHz]$$
 (4)

である必要がある.

また,エンコーダ内に符号化データの滞留を防ぐため,SDRAMの動作周波数

を設定上限の100[MHz]に決定した.

ー方,コンピュータ側に接続されている CameraLink 拡張ボードの周波数は 85[MHz] 以内である必要がある.したがって,ハードウェアエンコーダの出力 データ周波数はエンコーダ部の動作周波数と同じ値を用いることができる. PLL の構成上,100[MHz] と 81[MHz] と 20.25[MHz] を一括して1つの PLL で 実装できないため,2つの PLL を用いて実装する.各モジュールの動作周波数 を表 5.5 に示す.

ていていていていていた。 ていてい ひんしん ひんしん しんしん ひんしん しんしん ひんしん しんしん しんし		
エンコーダ部	$20.25 \mathrm{~MHz}$	PLL1
	$81 \mathrm{~MHz}$	PLL1
SDRAM	$100 \mathrm{~MHz}$	PLL2
データ入出力部	$100 \mathrm{MHz}$	PLL2
	81 MHz	PLL1

表 5.5: 動作周波数

6 設計上の課題とその解決

6.1 エンコーダ部

6.1.1 符号化方式の決定

ソフトウェアエンコーダと同様に,符号化種別判定を行いながら符号化する 実装を行うとハードウェアが非常に複雑になるため,符号化方式の決定を行う 専用ハードウェアが必要である.

また,4つの画素データが並列に入力されるため,隣接画素を差分符号化すると きの処理が煩雑になり,SDRAMの読み書きオーバーヘッドを少なくするため に,SDRAMのアクセス頻度を最小限にしなければならない.

圧縮行程を図 6.17 のように,符号化種別判定,符号化部,データセット生成部 に分割できる.

画素データがシリアルであれば隣接画素の差分計算を例外処理を施さずに実現 できる.パラレル入力による差分符号化の煩雑性は,パラレルシリアル変換器 を実装しシリアルデータをエンコーダに入力することにより解消できる.

一方で,データセット生成部でデータを一定の長さに加工して SDRAM に転送 することにより, SDRAM のアクセス回数が低減できる.



図 6.17: 圧縮行程の分割

差分値0が6個以上連続して出現したときはランレングス符号化を適用し,それ以外の場合は差分符号化を用いる.

符号化方式を決定する専用ハードウェアを図 6.18 に示す.



図 6.18: 符号化方式決定ハードウェア

フリップフロップを 6 つ直列に接続し, ランレングス判定部で連続する差分値 0 の数をカウントし, 6 つ以上であれば R = 1 にし, それ以外の場合は R = 0 とする.

符号化方式判定部では,異なる差分値が出現したらS = Rとし,同じ差分値の 場合は値を保持する.

6.1.2 24 ビット単位にデータをまとめる

1度に発生する符号の長さは24ビットであり, CameraLinkのデータ幅は12 ビットである.エンコーダ部で12ビットのデータを生成すると, FPGA を使用 ボード内に12ビットデータ生成モジュールが3つ設置されることになり, ハー ドウェア規模が増大する.よって,符号化データを24ビット単位にまとめる. 24ビット単位にデータをまとめる手順を図 6.19に示す.

バッファ内の使用ビット数の計算に符号長が用いられるため,図 6.20 に示す ように符号と符号長を含むデータをデータセット生成部に入力する.



図 6.19: 24 ビット単位のデータ生成手順

符号長 符号 [6ビット] [24ビット]

図 6.20: データセットの入力データ形式

6.1.3 フレーム末尾の例外処理

有効画像データが検出されたら順次符号化を開始すればフレームの最初を検 出する必要はないが,フレームの最後を明示する信号は入力されないため,フ レームの最後を表す信号を明示的に出力する必要がある.

また,フレームの区切りを検出したときに最後のデータセットが出力されずに エンコーダ内に残る場合がある.

したがって,フレームの最後を検出したらエンコーダ部に残っているデータセットを強制的に出力することが必要である.フレームの区切りを表す信号 vSync がセンサから出力されており,この信号線はハードウェアエンコーダに入力される.

vSync を用いてフレームの最後を表す信号 frameEnd を生成する.したがって, エンコーダ部のリセットを vSync で行うことはできないため, vSync を3サイ クル遅らせた vSync3 を用いてリセット動作を行う.vSync3 と frameEnd の動 作を図 6.21 に示す.



図 6.21: frameEnd 信号

writeEndCode はバッファの write 信号に接続されており, setMarker は強制的に符号を排出するための信号である.

setMarker が八イになったとき,データセット生成モジュールに符号長 24 ビットの 0 を入力する.これによりモジュール内に残っているデータを強制的に排 出することができる.

6.2 データ入出力部

6.2.1 バッファの設置と容量の決定

SDRAM には読み書きオーバーヘッドが存在する.したがって,データを即座に書き込んだり読み出したりすることはできない.入出力ビットレートを一定に保つために SDRAM の入力側と出力側にバッファを設ける. バッファの容量は2のべき乗単位でしか変更できないため,図6.22に示すような配置にバッファを設けるとバッファの総容量を柔軟に設定できる.



図 6.22: バッファの配置

動作周波数は, B1の入力側とB3の出力側が81[MHz]で, B1の出力側とB2・ SDRAM・B3の入力側が100[MHz]である. 符号化アルゴリズムの特性上,符 号発生はランダムであるため,最適なバッファ容量を机上で見積ることは非常 に困難である.よって,最大限の容量を実装しシミュレーション時にバッファの 使用状態をモニタリングして最適容量を決定する.

6.2.2 データ転送スケジュールの提案

SDRAM はデータの読み書きが同時にできないため, SDRAM からデータを 読み出している時はバッファに書き込みデータを一時的に溜めておく.そのた め,バッファオーバーフローが発生する可能性がある.バッファオーバーフロー が発生しないデータ転送スケジューリングが必要である. FPGA ボード番号 *m* のエンコーダ番号 *n* を *m* – *n* と表す. エンコーダ番号を固定し,各 FPGA ボードで転送タイミングを切り替えながら 転送することにより, SDRAM からの読出し時間を分散させバッファがフルに なる前に SDRAM へ書込みを行うことができる.

18枚の圧縮画像を順次転送するスケジューリングを図 6.23に提案する.



図 6.23: データ転送スケジューリング

動作周波数 100[MHz] でバッファB2 に 2[kword] データが溜るまでの時間は

$$\frac{2[\text{kword}]}{100[\text{MHz}]} = 20[\mu \text{s}] \tag{5}$$

であり,1フレームの読出し許容時間は1.8[ms] であるから,SDRAM 読み書き サイクル時間は

$$\frac{33[\text{ms}]}{\frac{1.8[\text{ms}]}{20[\mu\text{s}]} \times 3} = 122[\mu\text{s}]$$
(6)

である.したがって読み書きサイクル数は

- 2000 サイクル(読み)
- 10200 サイクル (書き)

になる.ただし,リフレッシュやプリチャージに関わるオーバーヘッドを無視 する.

6.2.3 SDRAM 読み書きオーバーヘッド低減

伸張開始前までに圧縮データをエンコーダ番号ごとに分類しなければならな い.データの並び替えをコンピュータで行うと CPU 時間が並び替えに使われ, 伸張に利用できる CPU 時間が不足する.分類処理をハードウェアエンコーダ側 で行う必要がある.

しかし,分類処理を行う専用のハードウェアを構成した場合,回路規模の増大 や遅延が発生するという問題がある.

その問題を解決するために,エンコーダ番号ごとに予め書き込み可能なアドレス を指定したうえで書き込みを行い,読み出し時は同一のエンコーダ番号のデー タが格納されているアドレスのみにアクセスを行えば,データを分類して読み 出すことが可能になる.

圧縮データは 25 ビット単位でエンコーダ部から出力される.データを分類す るために 2 ビットのエンコーダ番号を付与する (図 6.24).

|--|

図 6.24: エンコーダ番号付与

SDRAM は 16 ビット単位でデータを読み書きするため,図 6.25 に示すよう にエンコーダ番号が付与された 27 ビットデータを 2 つに分割する.

また,符号の並び順は上位側から下位側であるため,まず上位側を先にSDRAM へ転送し,次に下位側を SDRAM に転送する.

したがって,データは必ず2セットの16ビットデータが生成されるため,下位 側の16ビットデータ生成時にフレームエンド信号を付与する.

SDRAM の読み書きは通常,2次元アクセスで行われる.

本研究で用いた SDRAM コントローラの場合 , row アドレスが 1 度変化すると 少なくとも 10 サイクルはデータの読み書きができない時間が発生する .

したがって, row アドレスが頻繁に変化するようなデータの読み書きを行うと, SDRAMのアクセスに関わるオーバーヘッドが増大し,限られた時間内に読み 書きを終了することができなくなる.



図 6.25: データ分割

よって,データの読み出し順序を制御しつつ,SDRAMの読み書きのオーバー ヘッドを小さくするアクセス手法を考案する.

SDRAM の書き込み初期アドレスからのオフセットと対応するエンコーダ番号 rnum を表 6.6 のように予め決定する.

オフセット	rnum	オフセット	rnum
0,1	1	14,15	4
2,3	2	16,17	1
4,5	3	18,19	2
6,7	4	20,21	3
8,9	1	22,23	4
10,11	2	24,25	1
12,13	3	26,27	2

表 6.6: オフセットと rnum の関連

表 6.6 の関連を論理式で一般的に表す.ただし, InitAddr は書き込み初期ア ドレスであり, offset = address – InitAddr とする.

 $\{ \text{offset} \in \{0,1\} \lor (\text{offset} \ge 8 \land \text{offset} - 2^{[\log_2 \text{offset}]} \in \{0,1\}) \} \Rightarrow \text{rnum} = 1$

- $\{ \text{offset} \in \{2,3\} \lor (\text{offset} \ge 8 \land \text{offset} 2^{[\log_2 \text{offset}]} \in \{2,3\}) \} \Rightarrow \text{rnum} = 2$
- $\{\text{offset} \in \{4, 5\} \lor (\text{offset} \ge 8 \land \text{offset} 2^{[\log_2 \text{offset}]} \in \{4, 5\})\} \Rightarrow \text{rnum} = 3$

 $\{ \text{offset} \in \{6,7\} \lor (\text{offset} \ge 8 \land \text{offset} - 2^{[\log_2 \text{offset}]} \in \{6,7\}) \} \Rightarrow \text{rnum} = 4$

バッファから読み出したデータのエンコーダ番号と rnum が等しい場合はバッファから読み出したデータを書き込む.異なる場合は0を書き込んだ後アドレ

スを1ずつインクリメントする処理をエンコーダ番号と rnum が一致するまで行う.すなわち,書き込みはシーケンシャルアクセスとなる.

書き込みアルゴリズムのフローチャートを図 6.26 に示す.

SDRAM から読み出すときは,表 6.6 で与えられた関連にしたがって書き込み初期アドレスから順にアクセスすることにより,エンコーダ番号ごとにデー タを分類して読み出すことができる.

読み出し時は飛び飛びにアクセスされるが,データ間のアクセス距離は高々7で あるため,rowアドレスが頻繁に変化するということはなく,SDRAMの読み書 きオーバーヘッドは増大することはない.

読み出しアルゴリズムのフローチャートを図 6.27 に示す.







図 6.27: 読み出しアルゴリズム

図 6.26 と図 6.27 のアルゴリズムを用い, SDRAM の読み書きを高いスルー プットで実現するためにはデータの入力順が,エンコーダ番号1,エンコーダ番 号2,エンコーダ番号3,エンコーダ番号1,エンコーダ番号2,エンコーダ番号3... であればよい.

符号発生量が増大すると,バッファB1にはデータが常に存在するようになり, B2に入力されるデータの順番は,エンコーダ番号1,エンコーダ番号2,エンコー ダ番号3,エンコーダ番号1,エンコーダ番号2,エンコーダ番号3…に収束する. よって,符号量が増大するとスループットが向上し安定する.

バッファB1 からバッファB2 ヘデータを転送するハードウェア構成を図 6.28 に 示す.



図 6.28: B1 から B2 ヘデータ転送するハードウェア

6.2.4 コンピュータ側での出力データ識別

コンピュータ側でエンコーダ番号ごとにデータを識別できるような工夫を施 す必要がある.

SDRAM から読み出したデータにヘッダを付与してコンピュータにデータを転送すると, CamaraLinkのバス幅を最大限利用できないため,転送ビットレートが低下する.

よって, FPGA ボードから図 6.29 に示すような転送プロトコルにしたがってデー タを転送し, コンピュータでは送信側の転送プロトコルに基づいてデータを解 釈すれば CameraLink のバス幅を最大限に利用できるようになり, 転送ビット レートが向上する.

また,転送途中にデータ欠落が発生したか確認するチェックサムを無効データの

末尾に出力する.チェックサムは有効データと無効データのワード数の和である. ただし,1ワードは12ビットとする.



図 6.29: 転送プロトコル

6.3 設計の妥当性

符号化結果の妥当性は,符号化結果をソフトウェアエンコーダの符号化結果 と比較照合し一致しているか確認した.比較照合結果を表 6.7 に示す.

サンプル画像名	画素数	符号量 [words]	一致判定	
c0.pgm	1,200,000	8295	一致	
c1.pgm	1,200,000	8624	一致	
c3.pgm	1,200,000	7509	一致	
c4.pgm	1,200,000	5982	一致	
hikari_no.pgm	2,122,848	26611	一致	
pinama.pgm	1,268,520	10985	一致	
pinzure.pgm	1,268,520	21559	一致	
pinzure_f.pgm	2,122,848	74513	一致	
syouten.pgm	2,122,848	55400	一致	
tst_med.pgm	2,122,848	52460	一致	

表 6.7: 比較照合結果

すべてのサンプル画像の出力結果はソフトウェアエンコーダの符号化結果と 等しいため,ハードウェアエンコーダの出力は妥当である.

7 評価実験と考察

7.1 実験の条件と評価内容

実装したハードウェアエンコーダに画像データを入力し,LE数・メモリ量・ 消費電力を試算する実験を行い,5,6節で実装したハードウェアエンコーダに対 しての相対量で評価を行った.ハードウェアエンコーダにはS/P変換を介して, 符号発生量の最も多いサンプル画像を入力した.

LE 数とメモリ量は quartus2 でフルコンパイルした際に出力されるコンパイ ルレポートのデータを用い,消費電力は quartus2 の電力解析ツール PowerPlay Power Analyzer を用いて算出した.

設計データを ModelSim でシミュレーションすると,信号アクティビティと静 的確率情報が Value Change Dump File に出力され,これを PowerPlay Power Analyzer に設定することにより,ダイナミック電力を得ることができる[3].

7.2 バッファの容量決定

ハードウェアエンコーダに符号量の最も多いサンプル画像を入力し,バッファ の使用状況をモニタリングした.

バッファB3のモニタリング状況を図 7.30 に示す.



図 7.30: バッファB3の使用状況

バッファは空フレームの格納時に非常に多く使用されており,圧縮データを 扱う時のバッファ使用量は少なくなっている.

空フレームの発生原因は,圧縮データがSDRAMに存在していないのにもかかわらずフレームエンド信号が生成されているからである.

SDRAM に3フレーム分の圧縮データが存在しない場合はフレームエンド信号 を無視する機能を追加することによって,空フレームの発生を防ぐように改良 を施した.

次に,バッファの容量を最適化する.

手法1改良前のバッファの最大使用率は20%未満であったため,バッファB2と B3は60~80%の最大使用率になるように容量を設定した.

バッファB1の容量は,突発的な符号量増大に対応するため,最大使用量の6倍に設定した.

また,バッファ容量は IP コアの仕様上,2のべき乗単位でしか容量を設定できない.

バッファの使用状況をモニタリングし最適化した結果を表 7.8 に示し,最適化を 施した状態を手法1改良後(図 7.31)とする.

バッファ	現在容量 [words]	最大使用量 [words]	最適化容量 [words]
B1	256	5	32
B2	8192	323	512
B3	8192	1524	2048

表 7.8: 手法1改良前のバッファの最大使用量と最適化



図 7.31: 手法1改良後

一方,手法1改良前においてバッファB2を設けない手法(図7.32)も考えられる.これを手法2としてバッファ容量を最適化た.その結果を表7.9に示す.



図 7.32: 手法 2

表 7.9: 手法2のバッファの最大使用量と最適化

バッファ	現在容量 [words]	最大使用量 [words]	最適化容量 [words]
B1	2048	508	1024
B3	8192	213	512

7.3 評価結果

手法1改良前と手法1改良後と手法2の評価結果を表7.10・表7.11と図7.33 に示す.

手法1改良前に対し,手法1改良後と手法2のLE・メモリ・消費電力の削減量 について考察を行った.

手法1のメモリ量は376784 ビットに対し20%まで低減され71216 ビットにな リ,消費電力は303.16[mW] に対し86%まで削減され260.40[mW] になった. 一方,手法2のメモリ量は376784 ビットから24%まで低減され90216 ビットに なり,消費電力は303.16[mW] から14%まで削減され260.71[mW] になった.

LE に関しては,手法1改良前に対し,手法1改良後・手法2共に大幅な削減 を確認することはできなかったが,手法1改良後の方が手法2よりわずかにLE 数が削減されている. バッファは quartus2 に付属の非同期 FIFO を用いている.この FIFO を構成 する LE は少ないため,メモリ量を削減しただけでは大幅な LE 数削減を実現す ることはできない.

消費電力に着目すると,メモリサイズが大幅に低減できていることにより,回路の動作時に消費する電力 Dynamic Power が削減できている.

しかし,回路の規模に関わらないstatic 電力はほとんど変化せず,I/Oの条件を 一定にして実験を行っているためI/O Power もほとんど変化していない.

以上により,LE・メモリ・消費電力の観点から手法1改良が優れていること が明らかになった.

手法	LE	Memory[bit]	Total Power[mW]
手法1改良前	4384	376784	303.16
手法1改良後	4246	71216	260.40
手法 2	4319	90320	260.71

表 7.10: 評価結果

表 7.11: 消費電力内訳

手法	Dynamic Power[mW]	Static Power[mW]	I/O Power[mW]
手法1改良前	124.38	89.30	89.48
手法1改良後	84.37	89.01	87.03
手法 2	84.65	89.01	87.05



図 7.33: 手法1改良前に対する相対割合

8 エンコーダ部搭載数の増加

本節では,1台の FPGA ボードに搭載できるエンコーダ部の数について議論 する.

6 節までは FPGA ボードの制約条件緩和のために, FPGA ボード1台に接続す るセンサの数は3つと決定したが,もしピン数やエンベデッドメモリ量が多い FPGA ボードを使用することができる場合,1台の FPGA ボードにセンサ数を 3つ以上接続することが可能になる.

1 台の FPGA ボードに接続するセンサの数をできるだけ少なくする方が, 消費 電力や設備投資費用及び保守管理の手間の削減につながる.

センサ3つを接続したとき,DE0NANOのピン154個のうち99個を使用し, 33個はDE0NANOのDIPスイッチやLEDなどに使用される.したがって,空 いているピン数は

$$154 - 99 - 33 = 22[\texttt{I}] \tag{7}$$

である.

また,センサ1つ接続するごとにピン数は15個増加するため,DE0NANOに追 加接続できるセンサ数は1個である.

18 個のセンサを用いてレーザーの照射状態をキャプチャするため,3の倍数の 個数を FPGA ボードに接続することにより例外設計を排除する.

本研究では,別の FPGA ボードに 6 個のセンサを接続したと仮定して,7 節と 同様の実験を行い評価した.

ただし、電力の算出はすべてのピンアサインが適切に設定されてる必要があるが、DE0NANOに6つのエンコーダ部を搭載できないためピンアサインの設定

ができない.したがって,本研究では電力比較を行わないこととする. センサ1つにはエンコーダ部1つが接続されているから,ハードウェアエンコー ダに搭載されるエンコーダ部は6つである,エンコーダ部を6つ搭載したハー ドウェアエンコーダを手法3とする.

データ転送方法は,7節の手法1改良後と同等であるとする.

バッファの最適化容量を表 8.12 に示し,7節の手法1改良後2台と比較考察 を行う.

バッファ	現在容量 [words]	最大使用量 [words]	最適化容量 [words]
B1	256	28	64
B2	8192	2111	4096
B3	8192	159	512

表 8.12: 手法3のバッファの最大使用量と最適化

表 8.13: 手法1 改良後と手法3の比較

手法	LE	バッファ[bit]
手法1改良後2台	8492	142432
手法 3	7788	143136

LE・メモリ量について,2台のFPGAボードで実現した手法1改良後と1台のFPGAボードで実現した手法3の結果を表8.13に示す.

総 LE 数は手法 1 改良後 2 台より 704 減少,メモリ数は 704 増加した.

LE 数・メモリ使用量の観点から, DEONANO と同等の LE 数・メモリ量で空き ピン数が 23 個以上多い FPGA ボードを用いれば, システムを実現できること が明らかになった.

手法1改良後2台より手法3のLE数・メモリが小さくなっていることから,1 台のFPGAボードに搭載するエンコーダ数が多くなると,小さな回路規模でシ ステムを実現できる.したがって,Static電力を半分に削減でき,Dynamic電 力も削減できることが予想される.

9 あとがき

本研究では,レーザープリンタの印刷精度を決定するレーザー光の照射状 況を解析するためのシステムの一部を実装した.

18 個のセンサから入力される 540fps の画像を高速に圧縮伸張するという要求 に対して,ランレングス符号化と差分符号化を用いたハードウェアエンコーダ に画像を入力し,本研究で提案した SDRAM 読み書き手法を用いてデータ転送 を行うことにより 33ms の遅延を経てコンピュータに圧縮画像が転送されること をシミュレーション上で確認した.

実装したシステムは,1フレームあたりの許容転送時間を考慮して実装されているが,複数のFPGAボードが関わる部分は設計・実装されていない.また, 実機で動作確認を行うことができなかった.今後は複数のFPGAボードのタイ ミング設計及び6台のFPGAボードを接続し,実動作確認を行わなければならない.

回路構成については,2つの手法が考えられるデータ入出力部のうち,どちら が優れているかを回路規模・消費電力の観点から考察し,手法1の方が優れてい ることを明らかにした.しかし,メモリ容量の削減で実現できなかった LE 数の 低減には,回路構成をさらに見直すことが必要である.

設計したハードウェアエンコーダは FPGA ボードの制約により,3 つのセンサ が接続されている.しかし,ピン数やエンベデッドメモリ量の多い FPGA ボー ドを使用できるならば,1台の FPGA ボードに接続できるセンサ数を増やすこ とが可能になる.本研究では,ハードウェアエンコーダに6つのエンコーダ部 を搭載し評価実験を行った.その結果,LE・メモリ数は DEONANO と同等以上 で空きピン数が23多い FPGA ボードを用いれば,6つのセンサを1台の FPGA ボードに接続してシステムを実現できることを明らかにした.1台の FPGA ボー ドに多数のセンサを接続してシステムを運用することにより,ハードウェア規 模の低減や消費電力の削減,システムの維持管理や設備投資費用の低減が期待 できる.

謝辞

本論文を作成するにあたり,指導教員の近藤利夫教授,佐々木敬泰 助教に感謝いたします.

また,研究を通じて活発な議論にお付き合い頂いた動画班の同期の水野

君をはじめ,計算機アーキテクチャ研究室の同期・先輩の皆さまに感謝 いたします.

参考文献

- [1] リコープロダクションプリントソリューションズ・ジャパン株式会 社「レーザープリンタの動作原理について」
 http://www.rppsj.ricoh.co.jp/support/techinfo/laser_dousa1_jp.html
 (参照: 2014/1/21).
- [2] Terasic(2012) 「DE0NANO User Manual」 pp.4-24
- [3] ALTERA 「PowerPlay による電力解析」
 http://www.altera.co.jp/literature/hb/qts/qts_qii53013_j.pdf
 (参照: 2014/1/31).