動画像圧縮規格 H.264 の開発支援ツールの研究*

長谷川 辰雄**、菊池 貴**、菊池 清文**、
葛巻 清武***、木村 克久***、吉田 正雄****、田山 克也****

近年、車載カメラや DVD レコーダの画像の高精細化に伴い、その画像データ量が膨大に増え続けているため、画像の品質を落とさずデータ量を減らす圧縮技術が求められている。従来規格の MPEG2 では能力不足となってきており、MPEG2 の 2 倍以上の高圧縮率を可能とする新規格の H.264／AVC 対応製品の開発が各メーカの急務となっている。そこで本研究では、車載カメラ／情報家電メーカの要望が高い「動画像再生（デコーダ)」に的を絞った「H.264 用開発支援ツール」を試作開発した。
キーワード：H.264、デコーダ、開発支援ツール

Research of Development Tool Kit for H.264

HASEGAWA Tatsuo, KIKUCHI Takashi, KIKUCHI Kiyofumi,
KUZUMAKI Kiyotake, KIMURA Katsuhisa, YOSHIDA Masao and TAYAMA Katsuya

Recently, those amounts of image data keep increasing with development of mobile camera and high density DVD. The ability of the present MPEG2 technology is insufficient to deal with this enormous amount of data. The product makers require Software Developmet Kit for new specification H.264/AVC. This report is mentioned about the way of developing "the development support tool for H.264" with the present image compression of double and more.
key words : H.264, decoder, development tool kit

1 緒　言

試作開発した H.264 用開発支援ツールは、FPGA(Field Programmable Gate Array)開発支援ボード（ハードウェア）と、このボード上で稼働する専用ソフトウェアで構築される。FPGA 開発支援ボードは、小規模から大規模までの開発に柔軟に対応できるように、回路規模が異なる 2 種類の FPGA を搭載し、並行動作や協調動作の開発を可能とした。特徴は画像を取り扱うための大容量メモリ 512MB を搭載し、周辺デバイスに DVI 出力、USB、JTAG、SSRAM、FLASH を装備し柔軟な開発が可能となっている。これによって、小規模から大規模までのアプリケーション開発を、１つのボードで行えるハードウェア環境を実現した。また、専用ソフトウェアは、FPGA に H.264 データ供給するための CPU ソフトウェアと、FPGA 上で稼働する H.264 再生コア・ソフトウェアで構成され、H.264 コード解析が容易な開発環境を実現した。

2 実 験 方 法

2-1 開発条件

H.264 動画圧縮・再生用の組込み機器開発は PC で稼働するソフト開発とは異なり、省電力・省メモリという厳しい制約下で動画処理プログラムを動作させる必要がある。試作開発した FPGA 開発支援ボードは、組込み動画処理開発では一般的な動作クロック 100MHz、外部メモリが 512MB という条件で開発した。FPGA 回路プログラムの作成は、C 言語を Verilog 言語へ自動変換するソフトウェア「Codeveloper」を用いて記述したプログラムと、一部 Verilog 言語で作成したプログラムを組み合わせて開発した。表１に開発条件を示す。

表１　開発条件

FPGA 動作周波数	100MHz
外部メモリ(DDR2)	512MB
FPGA 回路プログラム	Codeveloper 用 C 言語及び Verilog 言語

2-2 FPGA 開発支援ボードの構成と特徴

FPGA 開発支援ボードの構成は、組込み動画機器の開発者が、様々な規模の回路ロジックに柔軟に対応できるように、大規模用、小規模用及び並列処理用の開発が可能

*　戦略的基盤技術高度化支援事業
**　電子情報技術部
***　有限会社エボテック
****　株式会社イーアールアイ

となる設計とした。具体的には回路規模が異なる２種類の FPGA と、それらと接続される周辺デバイスおよび、外部ホストと接続するためのホスト・インタフェースで構成されている（図１）。以下に本ボードの特徴を述べる。

①32bit／33MHz PCI、32bit／33MHz MPX の２種類の外部 I/F で、いずれか片方を外部ホストとして使用できる。

②StratixII、CycloneII２種類の FPGA を搭載しており、実装するデコーダの大きさ、速度などにより、何れか片方を選択して動作させる事ができる。また、２種類の FPGA 間にブリッジ I/F を持ち、２つの FPGA の並行動作または、協調動作を行うことも可能である。

③２種類の FPGA それぞれに DVI ドライバ／ポートが接続されており、何れか片方、または両方のポートから RGB 出力ができる。

④２種類の FPGA それぞれの演算に必要な様々なメモリーデバイスが接続されており、システムを自由に構築できる。

⑤ALTERA Daughter Card ポートを持ち、様々な外部接続を可能とするため、ALTERA Daughter Card の中から選択して接続することにより、本ボードの機能を拡張できる。

⑥ロジックアナライザを接続できるデバックポートを持ち、FPGA にロジックアナライザ I/F を実装することにより、ロジックアナライザでのデバッグが可能である。

図１　FPGA 開発支援ボード外観図

2-3 H.264 デコード解析用ソフトウェアの構成

H.264 デコードの機能は、エントロピー復号化、逆量子化・逆 DCT(Discreet Cosign Transform)、動き補償、画面内予測、重み付き予測、デブロッキングフィルタで構成され、その処理の流れは図２のように表される。消費電力、省回路規模が要求される組込み機器開発において、上記の H.264 デコード機能の実装は困難と言われている[1)]。そこで、本研究は H.264 の組込み機器開発を容易に解析・評価できることを目的に、デコード方法を３種類に分けて評価できるソフトウェアを開発した。第一は H.264 規格の参照ソフトウェアである JM(Joint Model)14.0 版（以下 JM）をベースにした解析ソフトであり、H.264 フル規格の評価を目的としたものである。第二は H.264 規格の最小構成であるベースライン・プロファイルをさらに機能限定し、小規模の開発を目的としたものである。第三はデコードの一部分の機能を FPGA 上に電子回路化し、高速化の評価を目的とするものである。下記に３種類の内容を説明する。

①フル規格版

H.264 規格の全てのデコード機能について、処理の流れや計算方法、データ構造を解析・評価することを目的とした組込みソフトウェアである。FPGA 上の NIOS Ⅱ CPU で稼働する。

②機能限定版

監視カメラや車載カメラシステム等、ある特定の小型アプリケーションの用途に合わせ、これに必要な機能に限定し解析・評価することを目的とした組込みソフトウェアである。FPGA 上の NIOSⅡ CPU で稼働する。

③高速化版

H.264 デコードの主要機能のうち、「動き補償」や「イントラ予測」、「デブロッキングフィルタ」を部分的に FPGA 回路化し高速化を図った組み込みソフトウェアである。動き補償では８～１０%の高速化を実現した。この FPGA 回路化に際しては、ソフトウェア開発で一般的な C 言語でアルゴリズムを開発し、それを FPGA 回路コード（Verilog）へ自動変換するソフト「Codeveloper」を使って実現した。Codeveloper を使用した H.264 のプログラム化は世界初である（2009 年３月時点）。NIOSⅡ CPU と FPGA 回路の協調で稼働する。

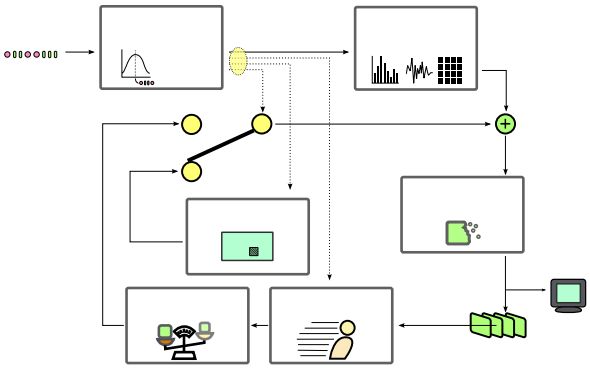


図２　H.264 解析用ソフトウェアの構造図

3 実 験 結 果

3-1 速度計測

効率よくデコード性能を向上させるためには、処理時間が長い部分からハードウェア化（FPGA 回路化）を進めた。ここでは、実際に FPGA ボード上で稼働する H.264 エンコードの各関数にかかる時間と、各処理時間の全処理時間に対する割合を求めた。計測方法は、FPGA にタイマーモジュールを搭載し、各関数の処理時間を 10nsec(1

億分の１秒)単位で計測した。試験条件及び結果を以下に述べる。

①テストデータ

動画像圧縮の国際標準化機関の JVT(Joint Video Team)で公開されているテストシーケンスを使用した。

表２のテスト１は高圧縮エントロピーの CABAC(Context-based Adaptive Binary Arithmetic Code)で圧縮したもので、テスト２は従来方式を拡張した CAVLC(Context-based Adaptive Variable Length Code)で圧縮したデータである。

②NIOSⅡ用ソフトウェア

速度計測のために、JM を改造し NIOS II IDE によりビルドしたものを使用した。

③NALU_t 構造体を削除し、NALU 切り出しモジュールをハードウェア化及び DPB 処理部分の効率化

表２に示す実験結果から、ReadMB(エントロピー復号化)、DecodeMB(動き補償)、Deblock(デブロッキングフィルタ)の機能に処理負荷が大きいことが分かった。

表２　速度計測結果

	テスト１	テスト2
FILE SIZE	194	395
ENTOROPY	CABAC	CAVLC
VIDEO FRAMES	30	30
REF-FRAME	IBBP	IBBP
FIELD/FRAME	FIELD	FIELD
RESOLUTION	720x480	720x480
StartMB (%)	1.904	2.201
ReadMB (%)	29.650	23.415
DecodeMB (%)	28.215	30.409
compute_colocated(%)	4.324	5.454
Deblock (%)	22.963	26.051
store_picture_in_dpb(%)	7.966	8.604

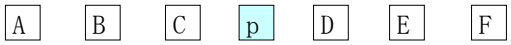
3-2　動き補償の FPGA 回路開発

H.264 の処理機能のうち、負荷の大きい「動き補償」を FPGA 回路化する方法及びその実験結果について述べる。

3-2-1　輝度信号の動き補償

(1)輝度信号の動き補償設計方法

動き補償処理は、輝度信号の動き補償と色差信号の動き補償の２つに分けられる。輝度信号の動き補償処理は、1/2 画素精度の予測画像を作成する際に 6 タップ・フィルタ処理を行う。このとき、マクロブロックサイズが 16×16 の場合、16×16～21×21 画素のデータが必要になる。6 タップ･フィルタ処理は、図３に示す整数画素 A, B, C, D, E, F を用いて、p=(A-5B+20C+20D-5E+F)/32 で表される積和演算であり、この処理を１マクロブロックにつき 256～529 回の演算を行うため処理量が非常に多い。また、1/4 画素精度の予測画像を作成する際にはさらに 256 回の平均処理が必要となる。このように、輝度信号の動き補償処理はメモリへの頻繁なランダムアクセスと、大量の計算処理が必要になる。これを高速化するため、高速メモリアクセスの「CDC(CoDeveloper Control)モジュール」及び「輝度信号の動き補償ハード」を開発した（図４)。「輝度信号の動き補償ハード」の開発には C 言語を VerilogHDL 言語に変換するツール（Codeveloper）を使用した。



p　　:1/2 画素精度

A, B, C, D, E, F：整数画素

図３　６タップフィルタの計算方法

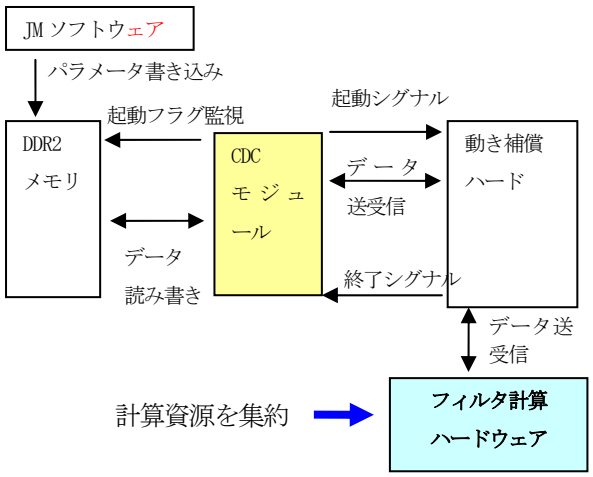


図４　輝度信号の動き補償のソフトウェア構造図

(2)輝度信号の動き補償実験結果

上記の CDC モジュールを使用して、輝度信号の動き補償の一部分の FPGA ハード化を行い、速度計測の比較実験を行った。この実験条件と結果を表３、４に示す。このハード処理によりソフト処理に対して約 160 倍以上の高速化を実現した。これによって、メモリアクセスの高速化及びデータの流れをストリームとレジスタ・インタフェースで処理する CDC モジュールと、Codeveloper で作成した C 言語によるハード化によって、FPGA 開発手法を実現した。

表３　輝度信号の動き補償実験条件

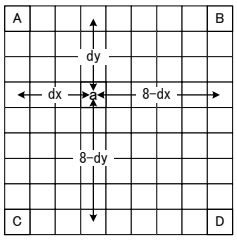
画像サイズ	FPGA 動作周波数	測定回数	NIOSⅡデータキャッシュ
320×240	100MHｚ	260 回	無し

表４　速度計測の実験結果

試験条件	ソフト処理 平均(μｓ)	ハード処理 平均(μｓ)	ソフト/ハード
動き方向：横	984.5	5.8	168.7

3-2-2　色差信号の動き補償

(1)色差信号の動き補償設計方法



・A,B,C,C ：整数画素
・dx,dy ：A から 1／8 画素の距離
・予測補間画素計算：
a=[(8-x)(8-y)A+dx(8-dy)B
+(8-dx)dyC+dxdyD]/64　･･･式 1

図５　色差信号の動き補償計算方法

色差信号の動き補償は、予測画素値を図５のように、整数画素値（A、B、C、D）を１／８画素精度で線形補間して生成する。線形補間とは、式１に示すように、元の信号からの距離に比例した係数を掛けて補完を行うことである。

(2)色差信号の動き補償実験結果

動き補償（色差）の FPGA 回路化の実行結果を表５に示す。ソフトウェアのみのデコード（再生）結果と、動き補償（色差）＋逆 DCT のハードウェアをソフトウェアに組み込んだ結果である。この結果から、6%程度の高速化ができた。ただし、この色差信号の動き補償ハードウェアのメモリアクセスは Codeveloper のメモリ関数を使用しているためアクセス速度は遅い。今後は CDC モジュールを組み込むことでさらに高速化を図る予定である。

表５　実験結果

実験構成	sec	fps
ソフトウェアのみ	17.954	1.671
ソフトウェア ＋動き補償（色差）＋逆 DCT	17.848	1.681

3-3　イントラ予測の開発

(1)イントラ予測の設計方法

符号化されたデータから画像を生成する際、同一画面内にて予測対象となるマクロブロック(縦 16x 横 16 画素を 1 マクロブロックとする)の周辺にある予測済画素値を参照して新たに画像を生成する方法をイントラ予測という。予測画素算出の後、変換係数を参照して逆離散コサイン変換(IDCT)を行い最終的な画素値を得る。輝度値のイントラ予測では、1 回の処理で予測を行う画素の単位が 4x4、16x16、8x8 と 3 通り存在する。4x4、8x8 については９つのモードのいずれかで予測が行われ、16x16 では Vertical、Horizontal、DC の 3 モードに Plane を加えた全 4 モードで予測を行う。色差については、8x8 単位で予測を行い、予測モードは輝度 16x16 と同様の 4 モードで予測を行う。これらの各モードについて、Codeveloper 用 C 言語でコードを作成し、FPGA 上で動作を確認した。

(2)イントラ予測の実験結果

イントラ予測をハード化（FPGA 回路化）した結果を表６に示す。この結果からハード処理ではソフト処理に対して部分的に 8～9 倍の高速化を実現した。試験条件はブロックサイズを 4×4 から 16×16 に変化させたものと色差信号とで比較した。

表６　イントラ予測の実験結果

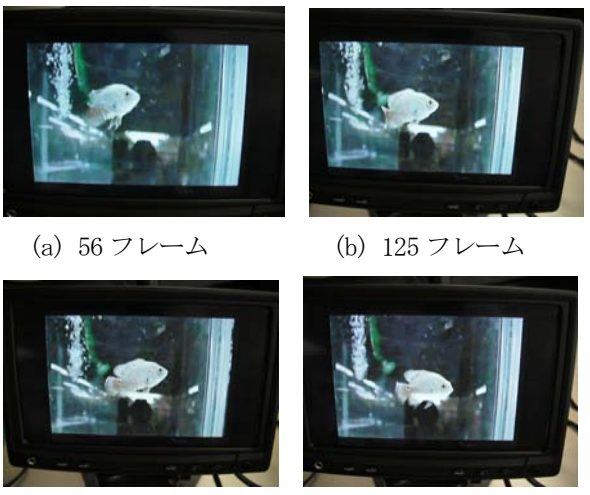
試験条件 (ブロックサイズ)	ソフト処理 (ms)	ハード処理 (ms)	比較 (ms)
イントラ 4x4	228.186	27.565	-200.621 (8.33 倍)
イントラ 8x8	176.129	19.422	-156.707 (9.09 倍)
イントラ 16x16	9.700	9.187	-0.513 (1.11 倍)
イントラ 色差信号	142.485	16.171	-126.314 (9.09 倍)

3-4　全体デコード結果

H.264 映像の全体的なデコードの実験では、機能限定版（NIOSⅡCPU のソフトウェア処理）と輝度信号の動き補償の一部を FPGA 回路化した高速化版の比較実験を行った。実験に使用した H.264 ファイルは独自に撮影しエンコードしたものである。表示速度の結果を表７に示す。

表７　表示実行結果

H.264 種類	フレーム数	ソフト＋ハード (FPS)	ソフトのみ (FPS)	高速化率
fall31	281	12.1	11.1	9.0%
ETCrayfish	199	10.0	9.3	7.5%
fish38	315	10.2	9.0	13.0%
Monkey	89	9.4	8.7	8.0%



(a) 56 フレーム　(b) 125 フレーム
(c) 238 フレーム　(d) 300 フレーム

図６　fish38 のデコード画像（抜粋）

また、この実験における画像表示の結果を図６に示す。

この結果からハード化によって表示速度(FPS ： Flame Per Second)の高速化を実現した。

4 結　言

開発した「H.264 開発支援ツール」は、小規模から大規模までの開発に柔軟に対応できるように、回路規模が異なる 2 種類の FPGA を搭載し、並行動作や協調動作の開発を可能とした。512MB の大容量メモリや周辺インタフェースも充実し開発自由度の高いツールとなっている。H.264 デコードソフトウェアは、フル規格版、機能限定版、高速処理版の３つの解析ツールとして実現した。その特徴はC 言語からハード言語への自動変換技術であり、本ツールの強みとなっている。ただし、ハード化は部分的にしか実装されておらず、全体的な性能の向上には至っていない。今後はハード化の範囲を広げ、全体的な性能向上を図っていく予定である。

文　献

１) 小野, 村上, 浅井, 動画像の高能率符号化、オーム社、pp. 144-149(2005)。