

次世代動画画像圧縮再生方式 H. 264/AVC の開発支援ツールの研究開発 -C 言語と HDL を併用したデブロッキング・フィルタの開発*-

菊池 貴**、長谷川 辰雄**、菊池 清文**、
中村 幸男***、三ヶ田 伸也***、吉田 正雄****

H. 264/AVC 形式の映像再生は複雑で大量の計算を行うため高速化が求められているが、開発コストの増加が問題となっている。そこで本研究では、複雑な設計が容易に記述できる C 言語を用いた FPGA 回路設計に取り組み、C 言語設計ツールである CoDeveloper (Impulse C) を用いて計算時間が長いデブロッキング・フィルタの専用回路を開発し検証を行った。

キーワード : H. 264/AVC、Impulse C、FPGA、デブロッキング・フィルタ

Research and Development of Support Tool for H.264/AVC Development of Deblocking Filter use of C and HDL

KIKUCHI Takashi, HASEGAWA Tatsuo, KIKUCHI Kiyofumi,
NAKAMURA Yukio, MIKATA Shinya and YOSHIDA Masao

Decoding of H.264/AVC video has high throughput but improvement of processing speed need high development cost. This report describes the development of processing circuits of Deblocking Filter with CoDeveloper which is C-based techniques for building FPGA-accelerated software application circuit.

key words : H.264/AVC, Impulse C, FPGA, Deblocking Filter

1 緒 言

デジタル映像機器の普及に伴い最新の動画画像圧縮規格である H. 264/AVC に対応した組込み機器の開発が進んでいる。H. 264/AVC 規格は従来規格の MPEG2 と比較して 2 倍の圧縮性能を持つが、処理の複雑化・計算量の増加により開発コストの増加が問題となっている。岩手県工業技術センターでは有限会社エボテックと株式会社イーアールアイと共同で FPGA (Field Programmable Gate Array) 開発支援ボード (図 1) と H. 264 デコード解析用ソフトウェア (図 2) の 2 つから構成される H. 264/AVC 開発支援ツールを開発した。このツールは、処理内容の解析・評価を容易に行うことが可能であるが、デコード処理の計算量が多く高速化が課題となっていた。

開発支援ツールでは処理の解析・評価を重視しているため内容が理解しやすく変更・拡張が容易なソフトウェア処理を FPGA 内のソフト CPU (NIOS) を用いて行っている。これらの処理を専用回路化 (ハード化) することで高速化が図れるが、H. 264/AVC の処理は規模が大きいいため全ての機能をハード化するためには莫大な開発コスト

がかかる。そのため、H. 264/AVC の IP (Intellectual Property : CPU コア) は符号・復号処理の一部についてのみ切り出したものの販売も行われている。当センターでは現在は用途の異なる 2 つのシステムを開発している。1 つは機能を限定した上で全体を HDL (Hardware Description Language : 回路記述言語) で開発する Verilog 版である。これは、デコーダ IP の商品化を意識したシステムであり可読性や拡張性に乏しいがデコーダ単体として必要十分な機能を有している。もう 1 つは可読性と拡張性を確保しつつ高速化を図っている Impulse C 版である。これは図 2 の H. 264 解析用ソフトウェアの構造を維持したまま各機能を段階的にハード化していくものである。実装に際しては C 言語設計ツールである Impulse 社の CoDeveloper (Impulse C) を用いており、C 言語のソースを HDL に変換してハード化したものと、NIOS 上のソフトウェアと連携して処理を行う。Impulse C を用いた場合開発効率が高いため HDL で開発するよりも開発コストを抑えることができるが、自動変換の性能が高くないため性能が得難いという問題もある。

* 盛岡市産学共同研究事業
** 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

本報告では復号処理の中でも計算時間が長いデブロッキング・フィルタ¹⁾について Impulse C を用いてハード化し、その処理のボトルネック部分について HDL を併用することで解決する手法について述べる。また試作したシステムについてのユーザー評価についても示す。



図1 FPGA 開発支援ボード

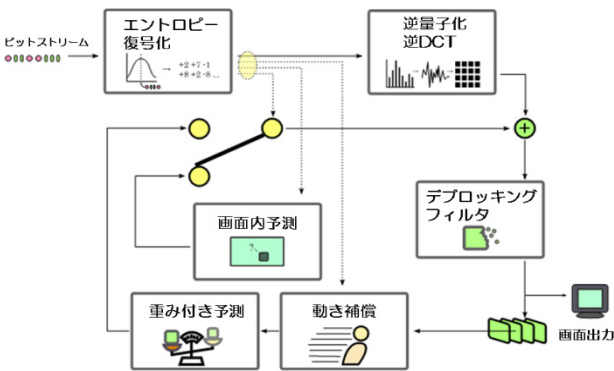


図2 H.264 解析用ソフトウェアの構造図

2 デブロッキング・フィルタ

2-1 概要

デジタル映像圧縮では、各画像を小領域 (16×16 画素のマクロブロック) に分割して個別に圧縮を行うため、その境界付近でブロックノイズと呼ばれるブロック間の歪みが発生する。映像の場合は復号済みの画像を使って次の画像を復号するフレーム間予測が用いられているため、このブロックノイズが以降の画像に伝播してしまう。そのため、一般にデジタル映像圧縮では圧縮率を高めると画質が劣化するという問題がある。これに対して H.264/AVC では、ブロックノイズの発生度合いに応じて適応的に平滑化を行うデブロッキング・フィルタが採用されている。しかし、この処理量は膨大となり復号処理全体の 20～40% を占める (表 1)。そのため、アルゴリズムの改善²⁾ やデータ転送の効率化³⁾、専用処理回路の開発^{4) 5)} など高速化に関する研究がなされている。

2-2 フィルタリング On/Off の決定

デブロッキング・フィルタは水平と垂直の 2 方向に対して行われ、図 3 に示す p, q により構成される 8 画素の

表 1 復号処理における各計算時間の比率

	テスト1	テスト2
入力 (%)	11.5	6.7
エントロピー復号 (%)	15.9	25.5
動き補償 (%)	22.4	31.1
デブロッキング・フィルタ (%)	30.7	25.0
出力 (%)	5.9	4.3
その他 (%)	13.6	7.4

ブロックを 1 つの処理単位として扱う。処理に先立ち対象の領域が画面内符号化であるかどうか、直行変換係数があるかどうか等の条件に応じてブロック境界強度 (B_s) が決定され、この値に応じてフィルタ処理を行う。フィルタリング処理は以下の条件が成立する場合のみ適用される。

- ① $B_s > 0$
- ② $|p_0 - q_0| < \alpha \wedge |p_1 - p_0| < \beta \wedge |q_1 - q_0| < \beta$
 α, β は量子化パラメータにより予め設定されている。

2-3 フィルタリング処理

$0 < B_s < 4$ の場合、以下の 4 タップの FIR (Finite Impulse Response Filter) フィルタを用いる。

$$\Delta = \text{Clip3}[-tc, tc, \{(q_0 - p_0)(2 + (p_1 - q_1) + 4)\} / 8]$$

$$p'_0 = p_0 + \Delta$$

$$q'_0 = q_0 - \Delta$$

ここで $\text{Clip3}(a, b, c)$ は c の範囲が $a \leq c \leq b$ となる

ようにクリップ処理を行う関数である。

$B_s=4$ の場合、 p の領域についてのフィルタリングは

$$|p_2 - p_0| < \beta \wedge |p_0 - q_0| < \alpha / 4 + 2 \text{ が満たされる時}$$

$$p'_0 = (p_2 + 2 * p_1 + 2 * p_0 + 2 * q_0 + q_1) / 8$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0) / 4$$

$$p'_2 = (2 * p_3 + 3 * p_2 + p_1 + p_0 + q_0) / 8$$

となり、満たさない場合は

$$p'_0 = (2 * p_1 + p_0 + q_0) / 4$$

となる。同様に q の領域に対しては

$$|q_2 - q_0| < \beta \wedge |p_0 - q_0| < \alpha / 4 + 2 \text{ が満たされる時}$$

$$q'_0 = (q_2 + 2 * q_1 + 2 * q_0 + 2 * p_0 + p_1) / 8$$

$$q'_1 = (q_2 + q_1 + q_0 + p_0) / 4$$

$$q'_2 = (2 * q_3 + 3 * q_2 + q_1 + q_0 + p_0) / 8$$

となり、満たさない場合は

$$q'_0 = (2 * q_1 + q_0 + p_1) / 4$$

となる。

このようにブロックノイズの発生度合いに応じてフィルタ強度を変えることで解像度を損ねることなくブロックノイズを除去している。フィルタリングの On/Off による画質の違いを図 4 に示す。

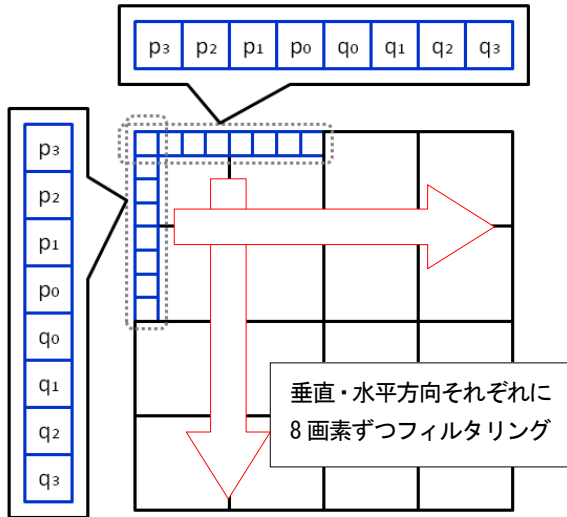


図 3 デブロッキング・フィルタの処理対象画素



図 4 デブロッキング・フィルタの有無による画質の違い

3 専用処理回路開発

3-1 Impulse C によるデブロッキング・フィルタの開発

Impulse C を用いて復号処理のデブロッキング・フィルタ処理の部分をハード化する。Impulse C は ANSI-C に準拠したプログラミング言語でありハードウェア開発用の専用ライブラリが準備されている。特に注目すべきは

メモリアクセス用の関数が用意されており、これを用いることで Avalon Bus ブリッジが自動で生成されメモリアクセスが可能となる。これにより NIOS II 上で動作している C 言語プログラムと専用処理回路は外部メモリのデータを共有することができハードとソフトの協調動作を行うことが可能である。システムの構成を図 5 に示す。しかし、Impulse C はポインタや構造体の記述については制限があり、多重ループの展開や配列を用いた処理の記述については自動変換により意図しない処理が行われるなどの問題がある。また配列で宣言したデータ領域は外部メモリに自動的に配置されるため、非常に大きなデータを扱えるが、処理のたびに外部メモリアクセスが発生し時間がかかってしまう。このように、Impulse C による開発は ANSI-C のソースコードをそのまま自動変換するのではなく、ハード用の記述を行う必要がある。

そこで、ソフトウェア用のデブロッキング・フィルタのアルゴリズムとデータフローを以下の条件を満たすように改良した。

- ▽ 必要なデータを外部メモリから一括で読み込む
- ▽ 画素データを外部メモリに配置せずにレジスタで保持する
- ▽ フィルタリングを行うモジュールを分離し、データをストリームで転送することで DSP を節約する
- ▽ 計算結果のデータを一括で外部メモリに書き込む

このシステムでは従来の配列を利用したプログラムと比較して外部メモリに対するランダムアクセスを大幅に低減しており、かつ回路規模も小さく抑えている。しかし、ModelSim を用いたシミュレーションの結果ではソフトウェアで処理した場合と比較して速度にほとんど変化がなくテストケースによっては速度が低下した。解析の結果、DDR2 メモリとの通信を行う際に 1 回のリードリクエストに毎回 20 クロック必要なため通信速度が低く全体の処理のボトルネックとなっていることが明らかに

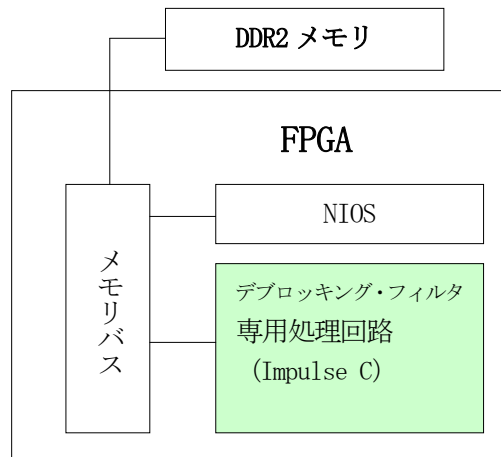


図 5 Impulse C を用いてデブロッキング・フィルタのハード化を行ったシステムの構成

なった。Impulse C で準備されているメモリアクセス関数を用いるとAvalon Busブリッジが自動的に生成されメモリアクセスが可能になるが、このバスブリッジの性能不足が原因で Impulse C のみで開発した回路では十分な性能が得られないことがわかった。

3-2 Impulse C と Verilog HDL を併用したデブロッキング・フィルタの開発

自動生成したAvalon Busブリッジの性能不足により処理速度が得られないことが明らかになった。これを解決するために、メモリアクセス部分を Verilog HDL で開発し、それを Impulse C で開発した専用処理回路に接続する手法を開発した。

専用処理回路は Avalon Bus からは切り離されるためバスブリッジの自動生成は不要となる。その代わりに Avalon Bus に接続されメモリアクセスと専用処理回路の起動と終了の制御を行う CDC (CoDeveloper Device Controller) モジュールを開発した。CDC は Verilog HDL で開発を行い、それを Impulse C から変換された Verilog HDL ソースコードと HDL レベルで接続する。図6にシステムの構成を示す。これにより Impulse C で開発された専用回路モジュールを隠蔽することが可能であり、ボトルネックとなっていた自動生成されるバスブリッジの存在を解消できる。

ModelSim を用いて1マクロブロック分のデブロッキング・フィルタ処理をそれぞれソフトウェアと専用処理

表2 マクロブロック1個をフィルタリングする際に必要な処理時間の比較

	試作版	改良版
処理時間 (clock)	213, 576	8, 183

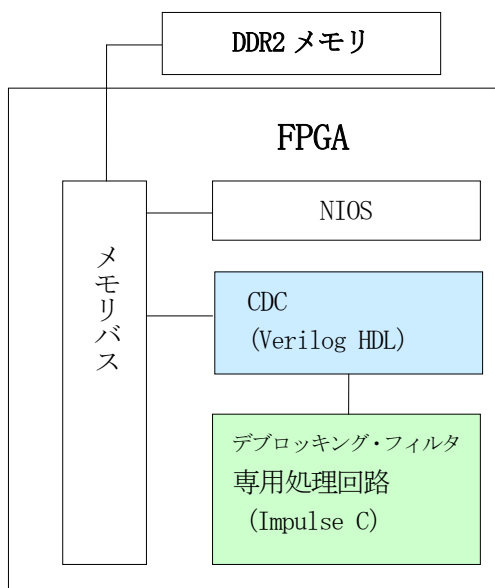


図6 Impulse C と Verilog HDL を用いたシステム構成

回路で行った場合に要するクロック数の比較について表2に示す。CDC を利用した改良版では処理に要するクロック数が約 1/26 と大幅に減少している。これにより、Impulse C による開発でボトルネックとなっていたメモリアクセスの部分が解決できた。

4 ユーザー評価

本研究で開発したシステムについて Embedded Technology 2009 (パシフィコ横浜、11/18~20) にて展示を行い、来場者から質問、要望、意見を頂いた。

主なものについて以下に示す。

- C 言語から HDL への自動変換技術が近年注目されているが実際にきちんと変換できるのか疑問である。
- 近年のデジタル回路は開発規模が大幅に増加しており HDL での開発が追いつかなくなっている。そのため、C 言語で開発ができると助かる。
- H. 264/AVC のような大規模なシステムを開発する上では C 言語設計は有効だと思うが、現場レベルではなかなか移行できない。
- C 言語ベース LSI 設計についてのプレゼンテーションも展示会中 3 回行ったが、いずれも 20 人以上の聴講者がおり、関係者の関心が高いことが伺えた。

また、HDL のみで設計した Verilog 版についての展示も同時に行った。システムの仕様は以下のとおりである。

- ▽ 動作クロック : 9.5MHz
- ▽ 画像サイズ : 352x288
- ▽ フレームレート : 30fps
- ▽ プロファイル : ベースライン

これについての質問、要望、意見も次に示す。

- 10MHz で再生できているのは驚きだ。
- IP コアを購入せずに独自に開発しているのは凄い。
- 既に販売されている IP コアとはどこが違うのか。
- 画像サイズがもっと大きければ購入を検討する。
- 今の仕様でも機能としては十分である。
- 価格はいくらなのか。
- エンコーダとセットでほしい。
- ドライブレコーダとしては価格の制限が厳しいため、H. 264/AVC の製品は出せない。

Impulse C 版と Verilog 版では、Verilog 版の方がより製品に近い具体的な質問や要望が多かった。一方で、Impulse C 版は発展途上の技術であるが、将来性が高い技術であると認識されていた。そのため、自動変換の精度や生成される回路性能、開発効率などに関する質問が多く、開発に携わる技術者が関心を示していた。

5 結 言

H. 264/AVC デコーダの高速化を図るために計算時間の長いデブロッキング・フィルタのハード化を行った。ハード化の際には開発効率の高い C 言語設計ツールである Impulse C を用いて実装を行ったが、Impulse C のみでハード化を行った場合には、自動生成されるバスブリッジの性能不足から通信速度が低くなり良好な性能が得られないことが明らかになった。そこで、メモリアクセス部分を担う CDC モジュールを Verilog HDL で開発し Impulse C で開発した計算処理回路と接続する手法について提案した。この手法を用いることでデブロッキング・フィルタの処理について従来の約 26 倍の高速化を実現した。この手法はデブロッキング・フィルタ以外の処理にも適用が可能な技術であるため、他の処理についても同様に適用することで高速化が期待でき、デコーダ全体の高速化を図ることが可能であるため、今後、開発する予定である。

また、ユーザー評価において、Impulse C 版については、技術力や将来性が評価されており、今後開発事例を増やしていくことで受託開発などの事業化へつなげる予定である。Verilog 版については、スペックの向上に関する要望が多かったため、今後画面サイズの拡大を行い製品化していく予定である。

文 献

- 1) Peter List, Anthony Joch, Jani Lainema, Gisle Bjontegaard and Marta Karczewicz, Adaptive Deblocking Filter, IEEE Transaction on Circuits and System for video Technology, VOL. 13, NO. 7, pp. 614-619 (2003)
- 2) Tsu-Ming Liu, Wen-Ping Lee and Chen-Yi Lee, An In/Post-Loop Deblocking Filter With Hybrid Filtering Schedule, IEEE Transaction on Circuits and System for video Technology, vol. 17, No. 7, pp. 937-943 (2007)
- 3) Tsu-Ming Liu, Wen-Ping Lee, Ting-An Lin and Chen-Yi Lee, A Memory-Efficient Deblocking Filter for H. 264/AVC Video Coding, IEEE Int. Symp. Circuits Syst. vol. 3, pp. 2140-2143 (2005)
- 4) Yu-Wen Huang, To-Wei Chen, Bing-Yu Hsieh, TuChih Wang, Te-Hao Chang and Liang-Gee Chen, Architecture Design for Deblocking Filter in H. 264/JVT/AVC, Proc. IEEE Intl. Conf. On Multimedia and Expo, vol. 1, pp. 693-696 (2003)
- 5) Chung-Ming Chen and Chung-Ho Chen, An Efficient Architecture for Deblocking Filter in H. 264/AVC Video Coding, IASTED Int. Conf. on Computer Graphics and Imaging, pp. 177-181 (2005)