

動的部分再構成可能なFPGAを搭載したFPGA/DSPプラットフォームRICEのためのコンフィギュレーション・コントローラの実装

川原, 崇宏
九州工業大学情報工学部知能情報工学科

田中, 康一郎
九州工業大学マイクロ化総合技術センター

佐藤, 寿倫
九州工業大学情報工学部知能情報工学科

<https://hdl.handle.net/2324/6307>

出版情報 : SLRC 論文データベース, 2006-01
バージョン :
権利関係 :



動的部分再構成可能な FPGA を搭載した FPGA/DSP プラットフォーム RICE のためのコンフィギュレーション・コントローラの実装

川原 崇宏[†] 田中 康一郎[‡] 佐藤 寿倫[†]

[†]{t-kawahara, tsato}@mickey.ai.kyutech.ac.jp

[‡]tanaka@cms.kyutech.ac.jp

[†]九州工業大学 情報工学部 知能情報工学科

[‡]九州工業大学 マイクロ化総合技術センター

プログラマブルな LSI を動的に再構成することで、適応型ハードウェア・アルゴリズムやランタイム・リコンフィギュレーションなどの新しいアプリケーション分野が展開できると期待される。このようなシステムを具現化するために、我々は動的部分再構成可能な FPGA である Xilinx 社の Virtex-II を搭載した FPGA/DSP 混載システム「RICE」を開発した。それに伴い、今回は RICE 用のハードウェアライブラリとして必須である動的部分再構成に必要なコントローラ及び DSP を制御するためのコントローラを開発した。動的部分再構成コントローラは、外部からのダウンロードケーブル経由で実行する必要があった動的部分再構成をスタンドアロンで自動的に必要最低限の時間で実行可能にしている。このコントローラ開発には主に FPGA 構成データを保持するメモリから FPGA にデータをダウンロードする際の高スループットに重点を置いた。また DSP コントローラは、本システムの特徴であるハードウェア・ソフトウェア協調処理が可能となるように、DSP と RICE 上のメモリを繋ぐインタフェースにダイレクトマップ方式のキャッシュを実装している。これにより DSP とメモリ間に専用のロジックを挿入する事が可能となる。本稿では RICE とこのハードウェアライブラリの詳細を述べる。

1 はじめに

FPGA に代表されるプログラマブルな LSI も、使用中はアプリケーションの書き換えを行わない、いわゆる静的での再構成が主となっていた。だが、近年このような LSI を動的に再構成する研究が盛んとなっている[1][2]。Xilinx 社の Virtex シリーズを始めとする FPGA の一部には、動的部分再構成機能を備えているものがあるこの機能を活用することにより、ハードウェア自身のアップグレード、適応型ハードウェア・アルゴリズム、ランタイム・リコンフィギュレーションなどの新しいアプリケーション分野が有効となる。また従来のアプリケーション分野においても、デバイス数の削減、省電力化、基板面積の節約などの効果が期待できる。

本稿では、FPGA を用いて動的部分再構成を実現するシステムである RICE を紹介する。次節にて RICE の概要を述べる。3 節では動的再構成に要するデバイスおよびそのコントローラについて、4 節では動的再構成を実現するためのコントローラについて解説する。5 節では RICE に搭載されている DSP を扱うためのコントローラについて言及する。最後に評価結果として 6 節で動的再構成コントローラの評価、DSP コントローラの評価をそれぞれ示す。

2 RICE

図 1 に示す RICE[3] は、動的部分再構成を用いたアプリケーション開発を主な目的として設計したシステムで

ある。本システムは Xilinx 社の FPGA である Virtex-II の動的部分再構成機能でランタイム・リコンフィギュレーションを実現している。RICE の構成を図 2 に示す。このブロック図に示すように RICE には FPGA、DSP、SDRAM などのデバイスを搭載しており、ハードウェアとソフトウェアの協調処理を行うことができる。RICE の FPGA は 2 つ搭載されており、それぞれ異なる役割を持っている。1 つは各デバイスをコントロールするため (Fixed) に用い、もう 1 つは部分再構成機能を用いた演算処理のため (Reconfig.) に利用する。これら 2 つの役割を同一 FPGA にて行わせることも可能であるが、検証時に問題点を容易に解決できるようにするため RICE では 2 つの FPGA で分担させて実現している。SDRAM は DSP や FPGA のための命令およびデータ、または FPGA (Reconfig.) のビットストリームデータを格納するために用いる。その他に USB インタフェース、Ethernet インタフェース、CMOS カメラを備えることで多様なアプリケーション開発や研究にも利用できる。

3 主要デバイスおよびコントローラ

RICE にて動的部分再構成を行う際に必要となるデバイスは、再構成に必要な命令やデータを入力するための USB および再構成用ビットストリームデータを格納している SDRAM である。また FPGA (Fixed) が FPGA (Reconfig.) に対して構成・再構成を行うための Download コントローラが必要となる。これら 3 つのデ

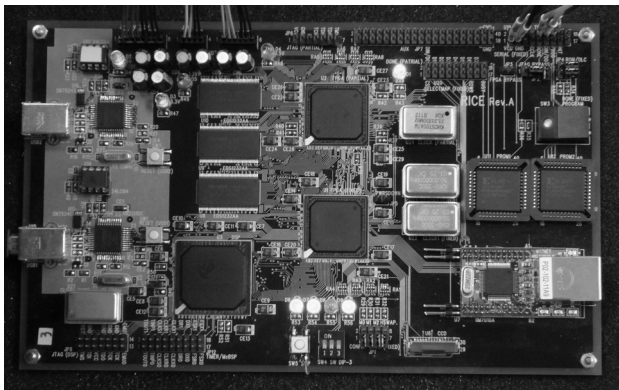


図 1: RICE

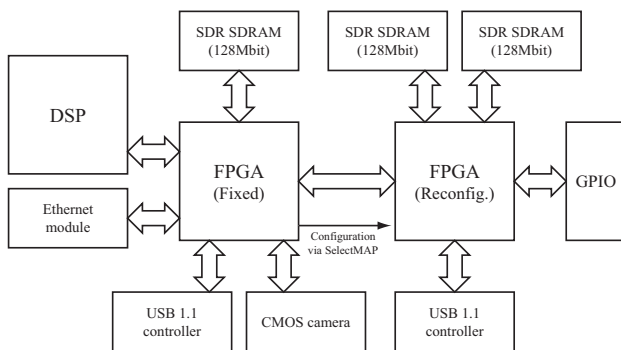


図 2: RICE の構成

バイスおよびコントローラにて動的部分再構成が実現される。

RICE に搭載されているデバイスおよびコントローラは、それぞれ最適な動作周波数が異なる。それらに対応できるように FPGA(Fixed) の内部回路はマルチクロックに対応できるように設計している。具体的には動作周波数が異なるロジック間で通信を行えるようにするために、FPGA の内部メモリである BRAM によるマルチクロック FIFO キューを各コントローラの境界に用意している。以下 FPGA(Fixed) に実装している各種コントローラの詳細について紹介する。

3.1 Download コントローラ

Download コントローラは、受信したビットストリームを FPGA(Reconfig.) にダウンロードするためのコントローラである。FPGA(Reconfig.) に対するビットストリームは、全体を再構成するデータと一部を再構成するデータに大別できるが、このコントローラでは両者のデータに対応することができる。

ランタイム・リコンフィギュレーションにおける要点の 1 つに、コンフィギュレーションのための時間がある。動的再構成を行うシステムでは、実際の処理時間はアプリケーションの実処理と再構成時間を加えた時間となるため、再構成に要する時間は非常に重要であ

る。本コントローラの開発では、この再構成時間を重視した。

Xilinx 社の Virtex-II シリーズには複数のコンフィギュレーションモードがあるが、その中でもパースナル・リコンフィギュレーションをサポートしているモードは SelectMAP およびバウンダリスキャンである。双方の再構成時間を比較すると、SelectMAP の方がバウンダリスキャンに比べて 10 倍以上高速である。このことからコンフィギュレーションモードには SelectMAP を採用している。

SelectMAP 用の Download コントローラの入出力端子を図 3 に示す。図の左側の端子は FPGA の内部からビットストリームデータを受信するためのインタフェースである。本コントローラに実装している FIFO キューは構成するビットストリームデータの一時保存用として使用しており、2KB の容量を確保している。ダウンロードの正常終了は DONEOUT ピンにより確認できる。また FULL ピンにより FIFO キューの状態が把握できる。一方右側の端子は FPGA(Reconfig.) のコンフィギュレーションピンに直接接続する端子である。PROG ピンは FPGA のリセットピンである、low にすることによってコンフィギュレーション・メモリをクリアする。コンフィギュレーションの状態は INIT ピンによって確認できる。また、ビットストリームデータ転送の成功は BUSY ピンにより確認できる。

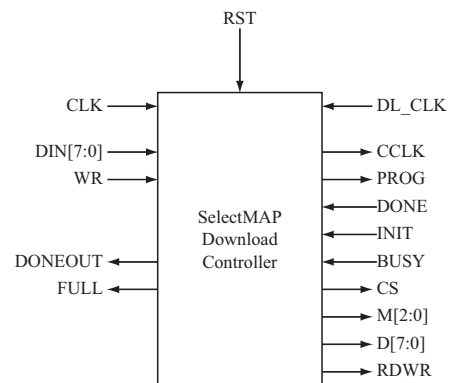


図 3: SelectMAP ダウンロード・コントローラの入出力端子

3.2 SDRAM コントローラ

SDRAM(Synchronous Dynamic Random Access Memory) は、従来の DRAM と比較して大域幅が大幅に改善されている。従来の DRAM はデータ、アドレス、コントロールを全て非同期制御で行っていたが、SDRAM は全て同期制御で行っている。そのため、アドレスとコントロールをデータの読み書きを行う以前にラッチすることができ、結果として現在のトランザクションを終了する前に次のトランザクションを平行し

て開始できる．

RICE に搭載している SDRAM は，Elpida 社の EDS1232AATA-75 である．この SDRAM は S-DR(Single Data Rate) で動作できる．容量は 128Mb であり， $1,048,576\text{words} \times 32\text{bits} \times 4\text{banks}$ で構成される．CAS Latency(CL) およびバースト長(BL) はプログラマブルである．CL に関しては CL=3 の時 133MHz，CL=2 の時 100MHz で動作させることができる．また BL に関しては 1，2，4，8，Full Page が選択可能である．リフレッシュサイクルは 64ms あたり 4,096 サイクル必要である．

SDRAM を制御するためのコントローラの構成を図 4 に示す．このコントローラは Xilinx 社の SDRAM コントローラ IP[4] をコアとして用いている．ただ，SDRAM IP のアドレス幅は 32Mb の容量までの対応となっているので，128Mb の容量に対応させるため若干の修正を行っている．FIFO キューは，入力用のデータバッファおよびアドレスバッファ，そして出力用データバッファに用いている．出力用のアドレスバッファを省略した理由は，回路動作の簡素化のためである．

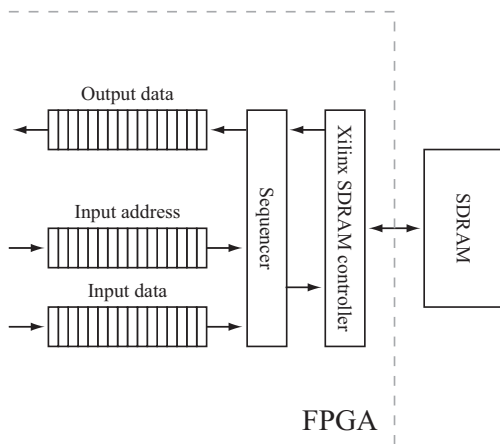


図 4: SDRAM コントローラの概略

コンフィギュレーション時間にはデータをメモリから読み込む速度にも依存する．そのため SDRAM の動作モードにおいてバンド幅を最大限に得るために，BL を SDRAM IP がサポートしている中で最大の 8 としている．また SDRAM の動作周波数を 100MHz としているため，CL=2 としている．

3.3 USB コントローラ

RICE には，搭載している USB を制御するために Cypress Semiconductor 社の AN2131SC という USB コントローラチップを実装している．このコントローラチップは内部に 8051 を拡張したプロセッサを搭載しているため，プログラムによって動作を定義することができる．また，動作周波数は 24MHz となっている．転送モードはバルク転送，インタラプト転送，および

アイソクロナス転送をサポートしている．

AN2131SC と FPGA を接続するための USB コントローラ制御回路の内部構成を図 5 に示す．この制御回路は，バルク転送用のバッファである FIFO キューとインタラプト転送のためのデュアルポートレジスタで構成される．FIFO キュー，デュアルポートレジスタ共に FPGA の BRAM を利用している．このコントローラでは，AN2131SC がサポートする転送モードのうち，バルク転送およびインタラプト転送を使用している．エンドポイント数は，各出力に対してそれぞれ 1 つずつを使用できる．

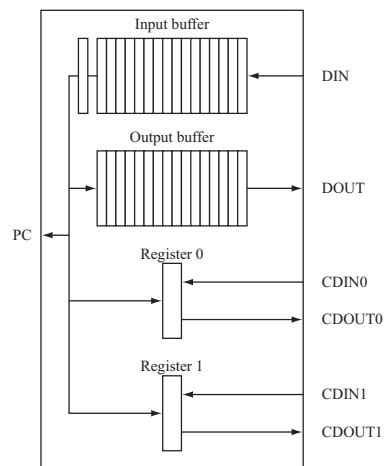


図 5: USB コントローラの概略

AN2131C のバルク転送時におけるバケットサイズは，1 バイトから 64 バイトまで変更できる．様々なアプリケーションに柔軟に対応できるように，この USB コントローラではバルク転送時のバケットサイズを PC から変更できるように設計している．デフォルトでは，送受信とも 64 バイトである．

4 コンフィギュレーション・コントローラ

コンフィギュレーション・コントローラは，前節で述べた 3 つのコントローラおよびこれらを統括するアービタによって構成される．このコントローラの概略を図 6 に示す．

アービタは USB コントローラから送られてきた命令の解釈および実行および各コントローラ間のデータの受け渡しを行う．また，SDRAM はデータバス幅が 32 ビットなのに対し USB および Download で扱うデータバス幅は 8 ビットである．これらの違いに対応する機能も実装している．

FPGA(Reconfig.) に対するコンフィギュレーションの流れを図 7 に示す．まず PC に用意されたビットストリームデータを FPGA(Reconfig.) にダウンロードするため，FPGA(Fixed) 内の Sequencer に対して命令を発

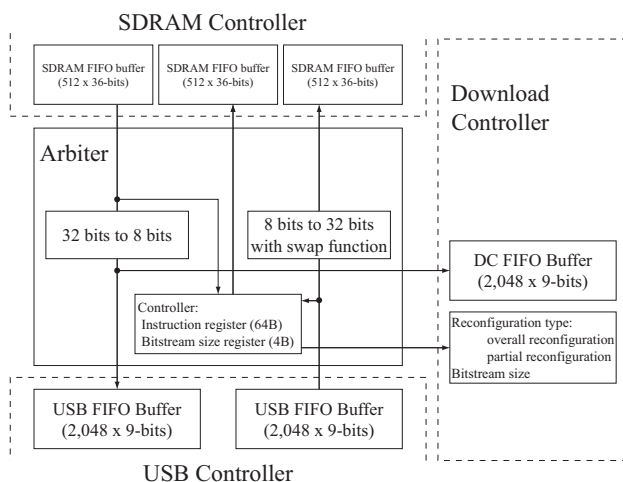


図 6: コンフィギュレーション・コントローラの概略

行する．なお，実際は Sequencer の機能はアービタが担っている．命令の内容には，PC から SDRAM ヘビットストリームデータを転送することと SDRAM 内のビットストリームデータを FPGA(Reconfig.) ヘダウンロードすることを明示する．次に，USB 経由でビットストリームデータを SDRAM に格納する．最後に，SDRAM からアービタを経由して Download コントローラにビットストリームデータを送る．Download コントローラは最初のデータが到着すると，すぐに FPGA(Reconfig.) に対してダウンロードを開始する．なお，ダウンロードを行う前にビットストリームデータを SDRAM ヘ転送する必要は必ずしも無く，いくつかのビットストリームデータを先に SDRAM に転送しておき，その後ビットストリームデータを選択しながらダウンロードすることも可能である．

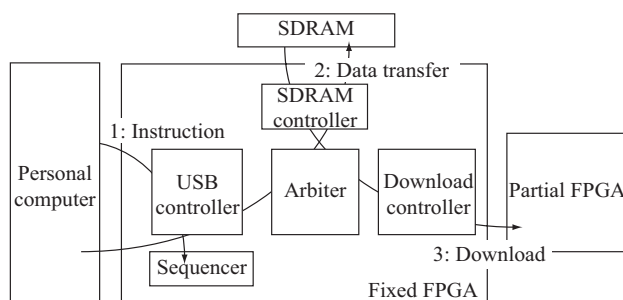


図 7: コンフィギュレーションの動作

部分再構成の動作も，上で述べたコンフィギュレーションとほぼ同様であるが，相違点は 2 つある．1 つは PROG ピンによる初期化は FPGA 全体を初期化するために使用できないことである．もう 1 つは再構成の成功を確認するための DONE ピンが利用できないために，確認用ロジックを組み込まなければならないこ

とである．

最後にビットストリームデータのファイル・フォーマットについて述べる．通常 Xilinx 社の論理合成ツール ISE にて生成される bit フォーマット・ファイルには，コンフィギュレーション・メモリ用のデータ以外にタイムスタンプやターゲットデバイスなどのヘッダ情報が格納されている．また，bin フォーマット・ファイルは bit フォーマット・ファイルからヘッダ情報を削除したものである．現状では，この bin フォーマット・ファイルを用いてコンフィギュレーションを行っている．なお，bin フォーマット・ファイルを ISE で生成するには，オプションを指定する必要がある．

5 DSP

RICE に搭載されている DSP(Digital Signal Processor) は Texas Instruments 社 (以下 TI 社) の TM-S320C6711C(以下 C6711C)[5] である．この C6711C は TI 社が提供している数種のプラットフォームの中の C6000 [6] と呼ばれるプラットフォームで，Very Long Instruction Word(VLIW) アーキテクチャが用いられている．また，浮動小数点演算命令にも対応している．外部メモリとして，SRAM(非同期)，SBSRAM，SDRAM に対応しており，RICE では搭載している SDRAM を使用している．最大動作周波数は 200MHz である．以下では，この DSP を制御するためのコントローラについて述べる．

5.1 DSP コントローラ

DSP コントローラは，C6711C 全体を制御するための 2 つのインタフェースコントローラによって成り立っている．インタフェースの 1 つは Host Port Interface(HPI)[7] であり，もう 1 つは External Memory Interface(EMIF)[8] である．HPI は DSP を制御するためのインタフェースであり，DSP 内部の制御レジスタ，内部メモリ，外部メモリに対して自由にアクセスすることができる．一方 EMIF は外部メモリのためのインタフェースである．外部メモリは 4 つの空間に分かれており，それぞれ異なるメモリ向けに使うことができる．

5.2 HPI コントローラ

前述した通り，HPI は DSP を外部から制御するためのインタフェースである．今回設計した HPI コントローラで求められる機能を次に示す．

1. DSP のブート
2. DSP 内の制御レジスタの設定
3. DSP 内部メモリ内へのプログラム書き込み
4. DSP 内部および外部メモリのデータ読み込み

HPI コントローラにも，前節で述べた主要デバイスコントローラと同様に BRAM で構成された FIFO キュー

を組み込んでいる．これは USB を介して PC から DSP に対して送られてきた命令やプログラム，データを一時保存しておくために用いている．

5.3 EMIF コントローラ

前述したように，DSP における EMIF インタフェースは SRAM，SBSRAM，SDRAM に対応している．ここでは，RICE に搭載されている SDRAM と接続するための EMIF コントローラおよび，SRAM をキャッシュとして接続する EMIF-DCACHE コントローラについて述べる．

SDRAM との接続 RICE では EMIF バス上にメモリを用意しておらず，FPGA が EMIF と直接接続されている．そのため FPGA によって DSP のメモリを構成する必要がある．DSP，FPGA，SDRAM の接続を図 8 に示す．単方向の信号線は，DSP の信号を SDRAM に FPGA を介して接続する．一方データバスのような双方向の信号線は，FPGA の入出力端子の 3 ステートバッファを使用する必要がある．データバスの調停には，SDWE ピンを使用する．

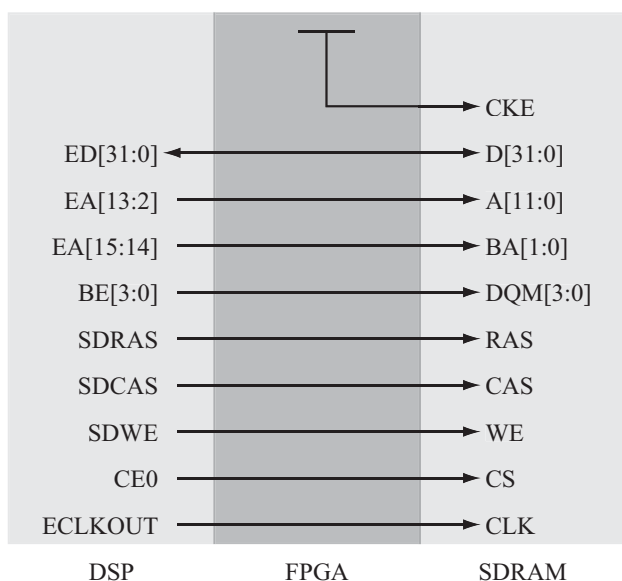


図 8: FPGA を介した EMIF と SDRAM の接続

Asynchronous SRAM との接続 続いて，EMIF コントローラにキャッシュを搭載したモデル (以下 EMIF-DCACHE コントローラ) について述べる．このコントローラは DSP の EMIF と SDRAM を直接接続せず，図 9 に示すように FPGA 内にキャッシュメモリを搭載し，それを介して EMIF と SDRAM の通信を行う．このような構成をとることで，図 10 に示すように EMIF と SDRAM の間に自由にユーザの意図したロジックを挿入することができる．

キャッシュメモリの構成方法として，最も単純なダイ

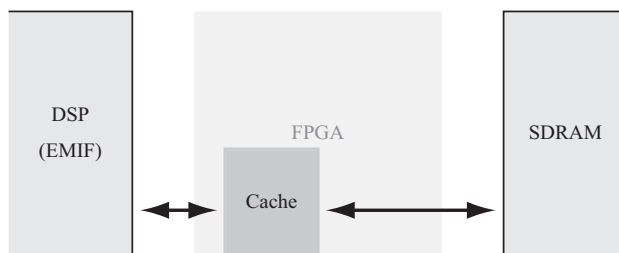


図 9: キャッシュを用いた EMIF と SDRAM の接続

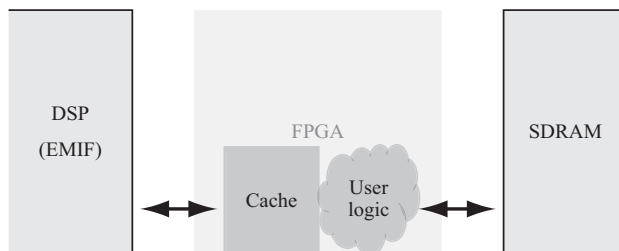


図 10: EMIF-DCACHE コントローラに対するユーザロジックの挿入

レクトマップを採用した．図 11 に示すようにダイレクトマップではデータが保管される Cache memory とタグと呼ばれるアドレスの一部が保管される Tag memory によって構成される．Tag memory は，DSP から EMIF 経由でデータを読み込みまたは書き込みを行う際，Cache memory に保管されているデータが DSP の期待するデータであるかどうかを確認するために使用される．もし Tag memory のデータとアドレスのデータが同一であれば Cache memory のデータは有効 (Hit) であり，同一でなければ無効 (Miss) である．なお Tag memory にはアドレス以外に Cache memory のデータ自身の有効を示す Valid ビットと Cache memory 内のデータと SDRAM のデータの相違を示す Dirty ビットの 2 ビットも一緒に保管されている．Cache memory のエントリ数は 64 である．ラインサイズは 32 バイトである．このサイズは RICE に搭載されている FPGA の BRAM のサイズから決定した．なお Tag memory に関しては BRAM を使用すると効率が悪いので，DFF で構成した．このように EMIF-DCACHE コントローラは非常に簡素に構成されている．

次に EMIF-DCACHE コントローラの動作について説明する．このコントローラでは図 12 に示すようにステートとして 4 つの状態がある．初期状態は IDLE ステートであり，この状態時に DSP からのリクエストを受信するとコントローラ内の状況に応じて READ/WRITE ステート，PREREAD ステート，WRITEBACK ステートのいずれかに状態を遷移する．READ/WRITE ステートへは，Tag memory 内のデータとリクエストされたアドレスとが同一であり (Hit)，かつ Cache memory 内

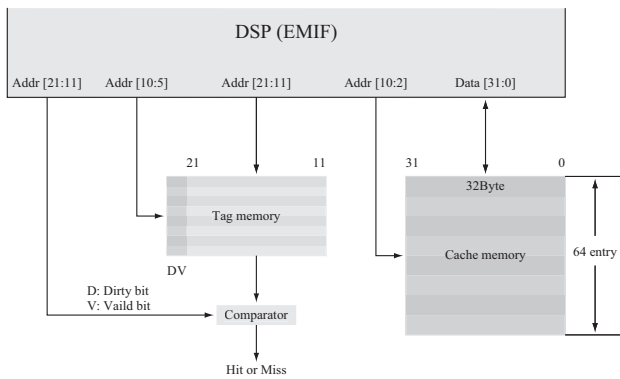


図 11: EMIF-DCACHE コントローラの構成

のデータが有効 (Valid) である場合に遷移する。一方 WRITEBACK ステートへは、Tag memory 内のデータとリクエストされたアドレスとが異なり (Miss)、かつ Cache memory 内のデータが有効 (Valid) であり、かつその Cache memory 内のデータに変更があった (Dirty) 場合に遷移する。そのほかの場合は PREREAD に遷移する。READ/WRITE ステートでは Cache memory を使って DSP とデータの読み書きを行う。また PREREAD ステートでは SDRAM から Cache memory へ必要なデータを読み込む。最後に WRITEBACK ステートでは Cache memory から SDRAM へ必要なデータを書き込む。これらのステージを遷移することで DSP と SDRAM はデータの読み書きを行うことができる。

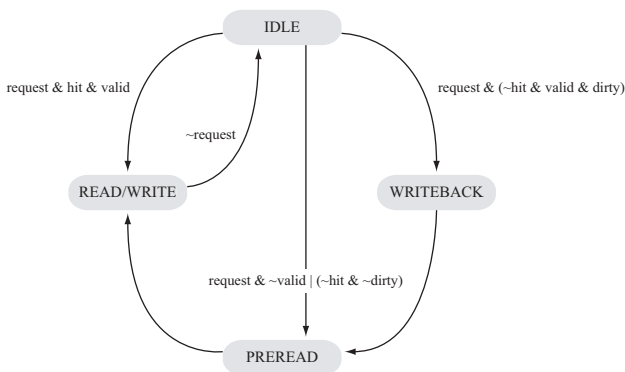


図 12: EMIF-DCACHE コントローラのフロー

5.4 実装時の全体図

DSP コントローラを使用するためのロジックの概要を図 13に示す。これは、DSP コントローラ、USB コントローラと、両者をつなぐアービタによって構成される。なお、アービタはコンフィギュレーション・コントローラのそれを拡張したものとなっている。

DSP の実行までの起動の流れを次に述べる。C6711C ではリセット後、まずエンディアンを設定を行う必要がある。次に外部メモリに対する設定を行い、プログ

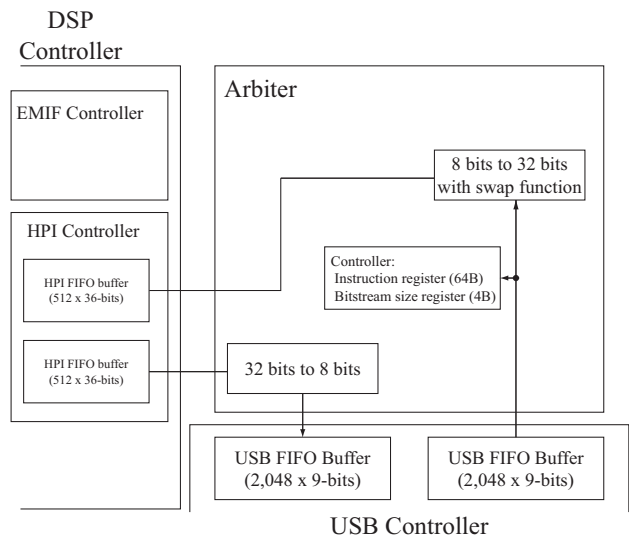


図 13: DSP コントローラの概略

ラムコートおよびデータをメモリに書き込む。最後に DSP を起動することで、書き込まれたプログラムの処理が開始される。

6 性能および評価

本節ではコンフィギュレーション・コントローラおよび DSP コントローラの評価について述べる。

6.1 コンフィギュレーション・コントローラの評価

コンフィギュレーション・コントローラおよび各コントローラ・ライブラリの回路規模を表 1に示す。表 1における Overall はコンフィギュレーション・コントローラ全体の回路規模を表しており、他の 4 つは各コントローラとそれらを接続するアービタのライブラリ単体の回路規模を表している。表 1で示すように、Flip Flop や Look Up Table(LUT) は 8 割以上余っている。また Block RAM(BRAM) においても、各々のコントローラで用意した FIFO キューの数だけ使用されている。アービタにおいては、各コントローラの FIFO キュー中のデータを受け渡しているため、内部に BRAM は必要ないので 0 となっている。

次に、表 1下部に示す動作速度について述べる。ダウンロード・コントローラは Virtex-II に SelectMAP モードでコンフィギュレーションを行うことができる最大速度である 50MHz[9] の周波数を分周により内部で生成するため、倍の 100MHz で動作させている。SDRAM コントローラは、使用している Xilinx IP での最大動作周波数と同じ 100MHz で動作させている。また USB に関しては AN2131SC の動作周波数である 24MHz で動作させている。動作速度は USB コントローラ・ライブラリ単体に比べてコンフィギュレーション・コントローラ

表 1: コンフィギュレーション・コントローラ回路規模

	Overall	Download	SDRAM	USB	Arbiter	Available
Flip flops	1,049	140	465	136	321	10,240
4-input LUTs	1,889	353	585	124	1,013	10,240
Block RAMs	8	1	3	4	0	40
Minimum period (ns)	19.804/20 (Core) 9.892/10 (Download) 9.870/10 (SDRAM) 9.508/42 (USB)	11.656/20 9.967/10	10.486/20 9.920/10	12.082/20 11.428/42	10.955/20	—

内部のほうが速くなっている．この理由は，USB コントローラのファンアウト数がコンフィギュレーション・コントローラに合成されることによって使用していないピンが削られ，減少したためであった．

続いて，性能の評価について述べる．今回の性能評価では，18,948 バイトのビットストリームを動的部分再構成することによって，実際に再構成が行われているか調査し，また再構成にかかる時間の計測を行った．評価を行うために，再構成前のクロックを分周した信号と最高生後のクロックを分周した信号は，それぞれ違うピンでプローブしている．また再構成中であることを示す信号線もプローブし，それぞれをロジックアナライザで観測した．

実際に動的部分再構成を実行した際に観測したロジックアナライザの波形を図 14 に示す．A2[5] ピンは再構成前のクロック分周信号で，A2[6] ピンは再構成後のクロック分周信号を表している．また，A2[1] ピンは再構成中であることを示す信号線である．これらの波形において，A2[5] ピンと A2[6] ピンをみると，動的再構成が行われたことが確認できる．また，A2[1] ピンが high のときに実際の再構成が行われており，その時間を計測したところ図 14 に示されているように 376usec かかっていることがわかる．FPGA(Reconfig.) に対し SelectMAP モードにてデータを構成する際，1 クロックで 1 バイトずつデータが送られている．すなわち，ダウンロードにかかる理論的な時間は， $18,948(\text{bytes})/50\text{MHz}$ 380(usec) となる．この理論値は，実測値とほぼ同値であるため，動的部分再構成が最速時間で完了していることがわかる．

6.2 DSP コントローラの評価

DSP コントローラの回路規模を表 2 に示す．EMIF における回路規模について，内部で構成しているキャッシュのタグメモリ領域は BRAM を使用するとリソースが無駄になる．そのため Flip flop を用いているので使用数が高めとなっている．またキャッシュメモリ領域は BRAM を使用している．それ以外のコントローラはコンフィギュレーション・コントローラと同様である．全体規模はコンフィギュレーション・コントローラよりも

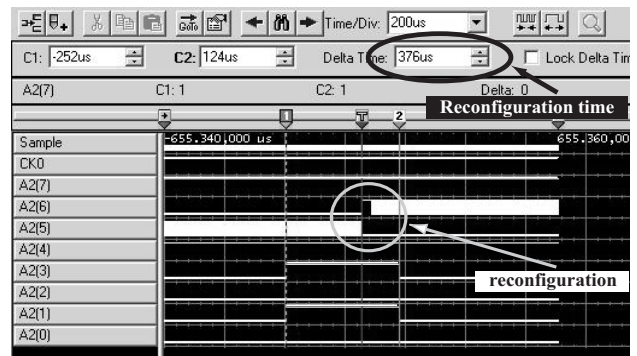


図 14: 動的再構成時の波形

大きくなっているが，6 割以上余っていることがわかる．

動作速度において，DSP コントローラは現状ではコンフィギュレーション・コントローラと同速度で動作させている．EMIF コントローラは SDRAM にアクセスするため，100MHz で動作させている．その他はコンフィギュレーション・コントローラと同様である．

DSP が正しく動作しているか評価するため，実際にプログラムを実行し動作結果の確認を行った．今回用いたプログラムは，外部メモリ上にある画像データをネガ反転し，結果を元画像データの後に書き込むというものである．DSP プログラムは DSP 内部メモリに，変換対象となる画像データは外部メモリである SDRAM にそれぞれ HPI 経由で PC からダウンロードした．

元画像を図 15 に，実行結果を図 16 にそれぞれ示す．実行結果から，元画像が正常にネガ反転されて出力されていることがわかる．

7 おわりに

本稿では，主に RICE における動的部分再構成と DSP コントローラについて述べた．現状では，RICE 用に開発した DSP コントローラでは HPI によるデータリードには一部問題があるものの，それ以外は問題なく動作することが確認できた．また，コンフィギュレーション・コントローラを用いて動的部分再構成を行うと，最大速度で実行できることも確認できた．

表 2: DSP コントローラの実装結果

	Overall	EMIF	HPI	USB	Arbiter	Available
Flip flops	2,593	2,047	190	136	383	10,240
4-input LUTs	3,453	2,257	143	124	1,492	10,240
Block RAMs	10	4	2	4	0	40
DCMs	2	2	0	0	0	8
Minimum period (ns)	19.572/20 (Core)	19.360/20	14.946/20	12.082/20	13.239/20	—
	9.972/10 (EMIF)	9.970/10				
	9.590/42 (USB)			11.462/42		



図 15: 元画像



図 16: 実行結果

今後は、DSP コントローラとコンフィギュレーション・コントローラの一体化を進めることで、ハードウェア・ソフトウェア協調処理を実現する予定である。

参考文献

- [1] Steiger, C., Walder, H. and Platzner, M.: Operating Systems for Reconfigurable Embedded Platforms: Online Scheduling of Real-Time Tasks, *IEEE Transaction on Computers*, Vol. 53, No. 11, pp. 1393 – 1407 (2004).
- [2] 邊輪一人, 岡嶋知宏, 柴村英智, 飯田全広, 久我守弘, 末吉敏則: リコンフィギャラブルクラスタコンピューティングに向けた FPGA ボードの開発, 信学技報 RECONF2005, pp. 25 – 30 (2005).
- [3] Tanaka, K.: Development of a Dynamically Reconfigurable Hardware Platform Using General-Purpose FPGAs, in *Proceedings of the 20th Commemorative International Technical Conference on Circuit/Systems, Computers and Communication*, Vol. 2, pp. 565 – 566 (2005).
- [4] 川原崇宏, 田中康一郎, 佐藤寿倫: リコンフィギュラブル・システム RICE における再構成時間の短縮, 第 13 回電子情報通信学会九州支部学生会講演会, p. 96 (2005).
- [5] Texas Instruments, Inc., : *TMS320C6711, TMS320C6711B, TMS320C6711C FLOATING-POINT DIGITAL SIGNAL PROCESSORS* (2004).
- [6] Texas Instruments, Inc., : *TMS320C6000 Peripherals Reference Guide* (2001).
- [7] Nikolic, Z.: *TMS320C6201/6701 DSP Host Port Interface (HPI) Performance*, Texas Instruments, Inc. (1998).
- [8] Texas Instruments, Inc., : *TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide* (2005).
- [9] Xilinx, Inc., : *Change to the Maximum CCLK Frequency for Virtex-II SelectMAP mode* (2004).