

DLX-FPGA マイクロプロセッサにおける 浮動小数点パイプラインの実現

井上 弘士^{†1} 中垣 憲一^{†2} 大内 正英^{†3} 柗山 太郎^{†4}
久我 守弘^{†5} 末吉 敏則^{†6}

†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
〒820 福岡県飯塚市川津 680-4

E-mail : {¹kouji, ²wally, ³ouchi, ⁴kuri}@mickey.ai.kyutech.ac.jp
⁵kuga@cms.kyutech.ac.jp
⁶sueyoshi@ai.kyutech.ac.jp

再構成可能な FPGA を利用した上級コース向き教育用マイクロプロセッサ DLX-FPGA における浮動小数点演算機能を実装した。浮動小数点演算器はパイプライン化されており、整数型命令と浮動小数点命令のオーバーラップ実行が可能である。また、再構成可能な配線用デバイスである FPIC を利用した DLX-FPGA プロトタイプ・ボードを用いて、複数 FPGA へ実装し動作確認を行った。本稿では、DLX-FPGA における浮動小数点パイプラインの設計仕様を述べ、回路図入力ならびにハードウェア記述言語を用いた設計事例を示す。さらに、複数 FPGA を搭載した DLX-FPGA プロトタイプ・ボードおよび DLX-FPGA における浮動小数点パイプラインの実装結果について述べる。

Implementation of the Floating-Point Pipeline for the DLX-FPGA Microprocessor

Kouji Inoue^{†1} Kenichi Nakagaki^{†2} Masahide Oouchi^{†3} Taichirou Kuriyama^{†4}
Morihiro Kuga^{†5} and Toshinori Sueyoshi^{†6}

† Department of Artificial Intelligence, Kyushu Institute of Technology
‡ Center for Microelectronic Systems, Kyushu Institute of Technology
680-4 Kawazu, Iizuka City, Fukuoka, 820 Japan

E-mail : {¹kouji, ²wally, ³ouchi, ⁴kuri}@mickey.ai.kyutech.ac.jp
⁵kuga@cms.kyutech.ac.jp
⁶sueyoshi@ai.kyutech.ac.jp

This paper presents the implementation of the floating-point functional units into the educational microprocessor DLX-FPGA which is used for the advanced computer architecture and system design education. Each floating-point functional units are constructed with the pipeline fashion, so they are able to perform overlapped execution both the integer and floating-point instructions. The DLX-FPGA has been implemented using eight (or nine) FPGAs on the DLX-FPGA prototyping board which utilized three FPICs (Field Programmable Interconnect Component), and its design has been confirmed. In this paper, we describe the specification of the pipelined floating-point functional units on the DLX-FPGA, and the design examples using the schematic editor or the hardware description language. Furthermore, we also describe the DLX-FPGA prototyping board which have twelve FPGAs, and the implementation results of the floating-point functional units into the FPGAs.

1 はじめに

近年、大学等の教育機関ではLSI技術を活用した計算機工学教育が重要視されている。本学を含むいくつかの大学では、教育用マイクロプロセッサを利用した計算機ハードウェア教育を行っており、高い学習効果をあげている[5]。これらの教育用マイクロプロセッサは、学生自らが設計したマイクロプロセッサを実際にLSI化し、自らの手で動作検証を行うものである。ハードウェアからソフトウェアまで一貫したシステム設計教育を支援し、特に初級から中級コースの教育に利用できる。

一方、現在の計算機では、命令パイプライン、キャッシュ・メモリ、仮想記憶といった要素技術が当然のように用いられ、計算機のスループットを向上させる様々な工夫がなされている。したがって、今後、これらの要素技術をふまえた上級コースにおける計算機工学教育が重要である。しかし、現状では上級コース向けの教育を支援する有効な教材は少ない。そこで我々は、先進的な計算機アーキテクチャ技術の教育やそれらの技術を利用した実用レベルに近いシステム設計教育を支援する教材として、教育用マイクロプロセッサDLX-FPGAの開発を行った[2][3]。

DLX-FPGAは教育対象を学部上級生や大学院生のような、計算機工学の基礎的な教育を終えた学習者を対象とし、システム設計教育における上級コースでの利用を目的としている。実装デバイスは書換え可能なFPGA(Field Programmable Gate Array)を利用しており、プロセッサ・モデルとして文献[1]に紹介されているDLXアーキテクチャを採用している。しかし、先に述べたDLX-FPGAでは回路規模を考慮し、DLXで定義されている整数乗除算命令、ならびに浮動小数点命令の実装は行っていない。そこで、本研究では、書換え可能なFPGA(約10,000ゲート相当の論理回路を実現可能)を12個搭載したDLX-FPGAプロトタイプ・ボードを利用して、DLX-FPGAに浮動小数点パイプラインを実装し、その回路規模を調査した。

本稿では、教育用マイクロプロセッサDLX-FPGAにおける浮動小数点パイプラインの実現について述べる。以下、第2章では教育用マイクロプロセッサDLX-FPGAの特徴について述べ、第3章では本研究で実装したDLX-FPGA浮動小数点パイプラインの設計仕様について言及する。次に、第4章ではハードウェア記述言語およびスキマティック・エディタを用いた回路図入力による設計事例について報告し、第5章でDLX-FPGAプロトタイプ・ボードへの実装結果について述べる。最後に、第6章で簡単なまとめを述べる。

2 教育用マイクロプロセッサDLX-FPGAの特徴

DLX-FPGAは、実装デバイスとして書換え可能なFPGAを利用した上級コース向き教育用マイクロプロセッサであり、プロセッサ・モデルとしてDLXアーキテクチャを採用している。現在、回路図入力およびハードウェア記述言語を用いたDLX-FPGAの設計、FPGAへの実装、ならびに動作確認を完了している。以下、教育用マイクロプロセッサDLX-FPGAが持つ3つの特徴について述べる。

DLXアーキテクチャの採用：DLXアーキテクチャは、米国の教育機関において計算機アーキテクチャの教科書として広く利用されている文献[1]で紹介されるアーキテクチャである。DLXは、現在主流となっているRISC型のアーキテクチャであり、実用化された商用の計算機を平均化した構成となっている。DLX-FPGAは、本アーキテクチャを採用しているため、Load/Storeアーキテクチャや命令パイプラインなどの高速化手法、システム・ソフトウェア教育における最適化コンパイラ技術といった実用レベルに近いシステム設計教育を支援する教材として有効である。なお、本アーキテクチャは、九州大学が開発を行っている教育用計算機QP-DLX[4]にも採用されている。

書換え可能なFPGAの利用：大学などの教育機関においてLSIを利用したマイクロプロセッサの開発実験を行う場合、限られた予算・時間内で実験を遂行しなければならない。そこで、最近FPGAがハードウェア教育で有効なデバイスとして注目を集めており、いくつかの大学で教育への応用が試みられている[6]。DLX-FPGAでも、実装デバイスとしてSRAM方式のFPGAを利用している。そのため学習者は、設計完了後その場で短時間にLSI化でき、即座に動作確認を行うことができる。また、書換えの回数に制限がないので、デバッグや改良のために何度でも設計のやり直しができる。

ハードウェア記述言語を用いた設計教育：設計手法には、スキマティック・エディタを用いた回路図入力による方法とハードウェア記述言語を用いた方法がある。従来、スキマティック・エディタを用いた回路図入力が設計手法としては主流であったが、回路規模の増大ならびに複雑化が進むにつれて開発期間の短縮が必須となり、ハードウェア記述言語の利用が著しく増大している。したがって、今後さらなるハードウェア記述言語の普及が予測されるため、教育機関におけるハードウェア記述言語を用いた設計教育が重要となる。また、ハードウェア記述言語はRTL(Register Transfer

Level) に基づいた設計が可能であるため、計算機アーキテクチャ教育のようにシステム設計・機能設計に重きをおいた教育にも適している。そこで、ハードウェア記述言語も考慮した DLX-FPGA の設計を行っている。

3 DLX-FPGA 浮動小数点パイプラインの設計仕様

DLX-FPGA 浮動小数点パイプラインを実現するにあたり、DLX-FPGA の命令セットならびにレジスタセットの拡張を行った。また、例外処理についても新たに浮動小数点演算例外を追加し、命令パイプラインを改良することで正確な割込みを保証した。以下、これらの拡張部分について述べる (以降、浮動小数点パイプライン実装前の DLX-FPGA を DLX FPGA 整数パイプラインと記す)。

3.1 命令セットの拡張

DLX-FPGA 整数パイプラインがサポートする 3 種類の命令セット (データ転送命令、算術論理演算命令、制御転送命令) に加え、表 1 に示す浮動小数点命令を追加した。DLX の命令セットでは単精度、倍精度の浮動小数点演算命令が定義されている。しかし、回路規模、FPGA の使用ピン数を考慮し、今回は単精度の浮動小数点演算命令に限定した。また、浮動小数点数は DLX と同様、IEEE 754 規格に準拠しており、命令形式は 32 ビットの固定長である。なお、オペレーションコードの割当てについては、カルフォルニア大学で開発された DLX のシミュレータである DLXsim のオペレーションコードを参考にして独自に決定した。以下、各命令の動作について説明する。

● 浮動小数点ロード/ストア命令

浮動小数点レジスタファイルとメモリ間でデータの転送を行う。実効アドレスはベースレジスタに符合付き 16 ビット変位を加算して得る。また、浮動小数点レジスタファイル間、浮動小数点レジスタファイルと整数レジスタファイル間でのデータ転送命令が用意されている。さらに、整数と浮動小数点数の相互変換命令を備える。

● 浮動小数点演算命令

浮動小数点数の加算、減算および乗算、除算命令が DLX で定義されている。浮動小数点加減算器および浮動小数点乗算器は設計を完了しているが、浮動小数点除算器の実装については、現在検討中である。また、浮動小数点比較命令では任意の浮動小数点レジスタの値を比較し、比較結果を浮動小数点状態レジスタの比較ビットに登録する。

● 制御転送命令

浮動小数点比較命令により登録された浮動小数点状態レジスタの比較ビットを分岐条件として参照する。分岐先アドレスは 16 ビットの符合付き変位とプログラム・カウンタを加算して得る。

表 1: DLX-FPGA に追加した命令

命令タイプ	命令の意味 (浮動小数点数は全て単精度)
データ転送命令	
LF	浮動小数点レジスタファイルへのロード
SF	浮動小数点レジスタファイルからのストア
MOVFP2I	浮動小数点レジスタファイルのデータを整数レジスタファイルへ転送
MOVI2FP	整数レジスタファイルのデータを浮動小数点レジスタファイルへ転送
浮動小数点演算命令	
ADDF	浮動小数点数の加算
SUBF	浮動小数点数の減算
MULTF	浮動小数点数の乗算
CVTF2I	浮動小数点数を整数へ変換
CVTI2F	整数を浮動小数点数へ変換
EQF, NEF LTF, GTF LEF, GEF	2つの浮動小数点数を比較し、その結果を浮動小数点状態レジスタの比較ビットに登録
制御転送命令	
BFPF	浮動小数点状態レジスタの比較ビットを
BFPF	分岐条件とする条件付き分岐命令

3.2 レジスタセットの拡張

DLX-FPGA に浮動小数点パイプラインを追加するため、以下のレジスタを追加した。

● 浮動小数点レジスタファイル (F0~F31)

32 ビットの浮動小数点レジスタを 32 個備える。

● 浮動小数点ストアメモリ・データレジスタ (FP-SMDR)

浮動小数点ストア命令を実行する際、メモリにアクセスする間ストアすべき浮動小数点数を保持する。

● 浮動小数点状態レジスタ (FP-SR)

浮動小数点演算に関する状態情報を管理する特殊レジスタである。文献 [1] では、浮動小数点状態レジスタの内部構成は定義されていないため、独自に決定を行った。浮動小数点状態レジスタは、図 1 に示すように 5 つのフィールドを持つ。C (比較) ビットは浮動小数点比較命令の実行時にセットまたはリセットされ、浮動小数点分岐命令により分岐の条件として参照される。Exceptions フィールド、TrapEnable フィールド、StickyBits フィールドは、浮動小数点例外の情報を保持するフィールド

であり、5つの例外要因に対して独自のビットを用意している。これらのフィールドの詳細については次節で述べる。RM フィールドは現在の丸めモードを表す。丸めモードとしては、IEEE 規格で示されている4種類のモード(最も近い数への丸め、0方向への丸め、 $+\infty$ 方向への丸め、 $-\infty$ 方向への丸め)を指定できる。

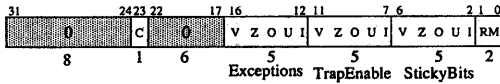


図 1: 浮動小数点状態レジスタ (FP-SR)

3.3 例外処理の拡張

DLX-FPGA 整数パイプラインでは、割込み処理において“正確な割込み”を保証している。“正確な割込み”とは、割込みを起こした命令の先行命令すべてがその実行を完了しており、割込み処理終了後、後続命令すべてがその実行を最初から正しく再開できるというものである。したがって、DLX-FPGA 浮動小数点パイプラインが追加されても、この“正確な割込み”は保証する必要がある。

浮動小数点演算の割込み要因としては、IEEE 規格で定義されている以下の5つの算術例外を追加する。

- V: 無効演算例外
- Z: ゼロ除算例外 (現在は除算をサポートしていないため未定義)
- O: オーバフロー例外
- U: アンダフロー例外
- I: 精度異常例外

浮動小数点演算命令の実行時において、浮動小数点例外が発生した場合、浮動小数点状態レジスタの Exceptions フィールドおよび StickyBits フィールドの対応するビットがセットされる。そして、予め設定しておいた TrapEnable フィールドを調べ、その例外要因に対応するビットがセットされていれば例外発生として割込み処理ルーチンに制御を移す。対応するビットがリセットされていれば例外発生は受理されず、そのままプログラムの実行を続行する。StickyBits フィールドは過去に1回以上の当該割込みがあったことを示すビットであり、1度セットされたら浮動小数点状態レジスタに新しい値を書き込まない限り変更されない。

3.4 命令パイプライン

DLX-FPGA 整数パイプラインは図2に示すように IF, ID, EX, MEM, WB の5ステージで構成されている。これに浮動小数点加算器と浮動小数点乗算器を

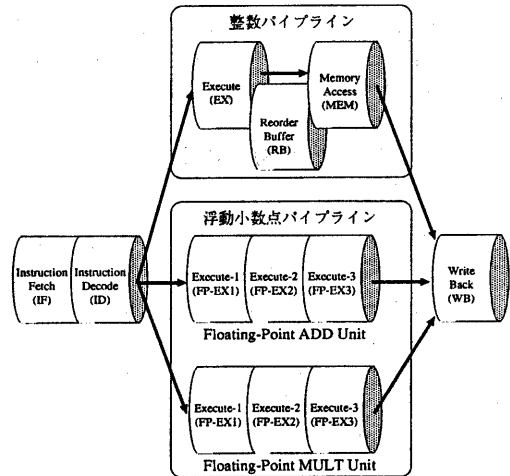


図 2: 浮動小数点パイプラインを実装した DLX-FPGA 命令パイプライン

追加し、整数型命令と浮動小数点命令のオーバーラップ実行、さらに浮動小数点加減算命令と浮動小数点乗算命令のオーバーラップ実行を可能にした。また、各浮動小数点演算器はパイプライン化されているため、同一演算器を使用する浮動小数点命令同士の時間的オーバーラップ実行も可能である。各浮動小数点演算器は、各ステージの実行遅延時間やオーバーラップ実行における正確な割込みの保証(詳細は後述)などを考慮して、図3で示すように3つのパイプライン・ステージ (FP-EX1, FP-EX2, FP-EX3 ステージ) で構成した。

以下、DLX-FPGA 浮動小数点パイプラインの追加によって発生する問題点とその解決策について説明する。

1. 浮動小数点演算命令の実行による、WB ステージでのレジスタファイルに対するアクセス競合

浮動小数点加算器と浮動小数点乗算器のパイプライン・ステージ数が異なる場合、WB ステージにおいて演算結果を同時にレジスタファイルへ書き込む状況が発生する。そこで、本浮動小数点パイプラインでは図2に示すように、浮動小数点加算器と浮動小数点乗算器のパイプライン・ステージ数を同じにすることで、浮動小数点レジスタファイルに対する書き込みの競合を回避した。

2. RAW(Read After Write) ハザードの発生

各浮動小数点演算器で実行中の命令の演算結果を当該命令が (ID ステージにおいて) 必要とする場合データハザード (RAW ハザード) が発生する。このデータハザードは、フォワーディングでは解決できない。なぜなら、演算結果は現在計算中なので確定していない

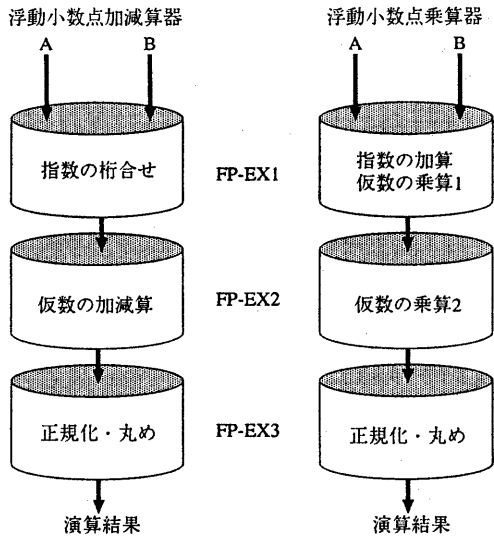


図 3: パイプライン化浮動小数点演算器

からである。このような場合、先行命令の演算が終了するまで当該命令の発行を停止する。そして、先行命令の演算終了後、当該命令は先行命令の演算結果を得て発行する。ただし、図4のブロック図で示すように、先行3命令目の浮動小数点演算命令に対しては、フォワーディングによりその演算結果を得ることが可能である。これにより全てのRAWハザードが解消される。また、レジスタファイルの読み出しは全命令とも同一ステージ(IDステージ)で行われるためWARハザードは発生しない。

3. 正確な割込みの保証

DLX-FPGA 整数パイプラインでは、パイプラインの各ステージで割込みが発生した時、命令と共にパイプライン・ステージを流れる割込みステータスペクタ(ISV)にその旨を記録する。そして、そのまま命令の実行を続け、最後のWBステージで割込みを検出する。この方法によりDLX-FPGA 整数パイプラインでは正確な割込みが保証されている。

DLX-FPGA 整数パイプラインに本浮動小数点パイプラインを追加した場合、整数型命令同士または浮動小数点命令同士のオーバーラップ実行では順発行(in-order issue)の乱完了(out-of-order execution)となる。しかし、整数型命令のEXステージは1クロックサイクルで実行を完了するのに対し、浮動小数点演算命令のEXステージは3クロックサイクル必要である。つまり、整数型命令と浮動小数点命令のオーバーラップ実行では順発行

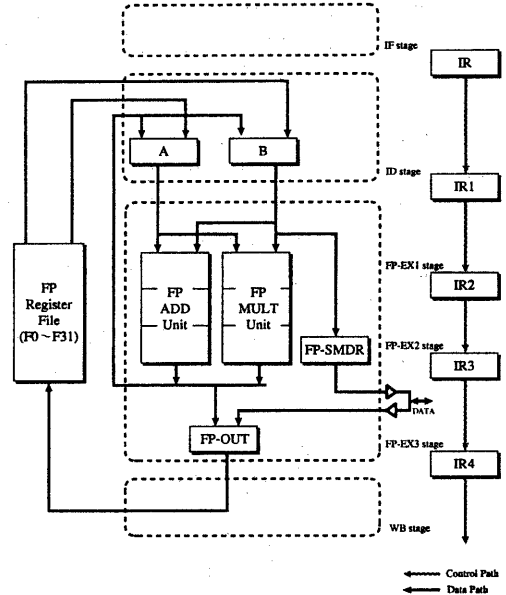


図 4: DLX-FPGA 浮動小数点パイプラインの構成

の乱完了となるため、正確な割込みを保証する何らかの方法を考慮しなければならない。そこで、図2に示すように、DLX-FPGA 整数パイプラインではEXステージの後にReorder-Bufferを追加した。浮動小数点命令が発行されると、後続の整数型命令に対してReorder-Bufferが有効となる。そして、浮動小数点命令の実行完了後、整数型命令のうちEXステージ以降を必要としない命令(分岐命令, NOP命令など)が発行されると、その後続命令ではReorder-Bufferが無効となり、通常どおり5ステージの命令パイプライン処理を行う。これにより、整数型命令と浮動小数点命令とのオーバーラップ実行においても、レジスタ等の資源は命令の発行順に更新を行うことが可能となるので、正確な割込みを保証することができる。

4 設計事例

4.1 設計工程

DLX-FPGA 浮動小数点パイプラインの設計工程を図5に示す。回路規模の調査を行うため、ハードウェア記述言語による設計、ならびにスキーマティック・エディタを利用した回路図入力による設計を行った。

ハードウェア記述言語による設計では、IEEEにより標準化されているVHDL(VHSIC Hardware Description Language)を用いた[8]。まず、設計仕様に従いRTL(Register Transfer Level)で回路を記述する。そして、VHDLシミュレータによる動作確認完了後、論

理合成ツールを用いて論理回路への変換・最適化を行う。その後、ネットリストへ変換し、構成データを生成する。この構成データをFPGAへダウンロードすることで、自ら設計した論理回路がFPGAに実装される。次に、LSI化された論理回路の動作確認を行い、誤りがあった場合は、以上の工程を繰り返す。

回路図入力による設計では、設計仕様に従い、スキーマティック・エディタを用いて論理回路を入力していく。回路図入力完了後、論理回路シミュレータにより動作確認を行い、図面データをネットリストに変換する。その後は、ハードウェア記述言語の場合と同様に構成データを生成し、FPGAへのダウンロード完了後、動作確認を行う。

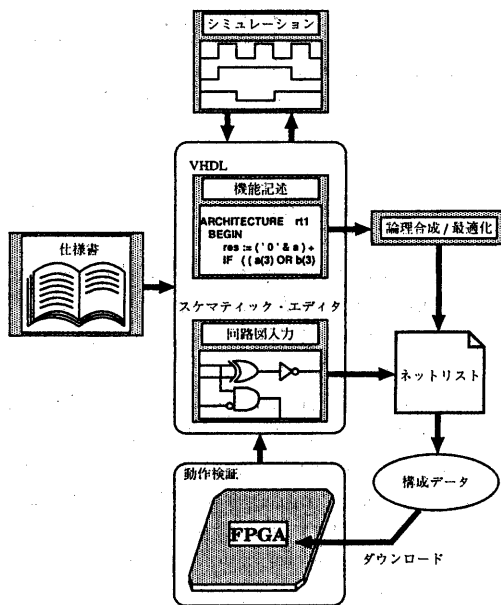


図 5: 設計工程図

4.2 複数FPGAに対する分割設計

DLX-FPGA 整数パイプラインは、Register Unit, ALU Unit, Address Unit, Control Unit の4つの機能ユニットに分割され、これらが約 10,000 ゲート相当の論理回路を実現できるFPGA(米国 Xilinx 社のXC4010[7])4個に実装される。この分割実装においては、DLX-FPGAが教育に使用されることを考慮して、無理のない自然な分割(機能分割)で実装した[2]。

DLX-FPGA 浮動小数点パイプラインの実装においても、回路規模の関係から複数個のFPGAが必要であり、DLX-FPGA 浮動小数点パイプラインを複数個のFPGAへ分割実装しなければならない。また、論理回

路の分割においてもDLX-FPGA 整数パイプラインと同様、無理のない自然な機能分割を実現する必要がある。そこで、図6に示すように、DLX-FPGA 浮動小数点パイプラインを3つの機能ユニットに分割する。各機能ユニットの詳細を以下に示す。

1. **FP Register Unit:** 主に浮動小数点レジスタファイルで構成される。また、先行3命令目の演算結果に対するフォワーディング機構を備えている。
2. **FP Add Unit:** 浮動小数点加算器で構成される。また、浮動小数点減算命令、整数と浮動小数点数の相互変換命令、浮動小数点比較命令もこのユニットで実行する。
3. **FP Mult Unit:** 浮動小数点乗算器で構成される。

各機能ユニットはバスにより接続されており、Control Unitにより整数パイプライン、浮動小数点パイプラインの制御を行う。ただし、FP Add Unit, FP Mult Unitをそれぞれ1個のFPGA(XC4010)に実装することは、回路規模的に困難である。そこで、VHDLによる設計では、FP Add Unitに2個、FP Mult Unitに2個のFPGAを利用し、回路図入力による設計では、FP Add Unitに1個、FP Mult Unitに2個のFPGAを利用した。

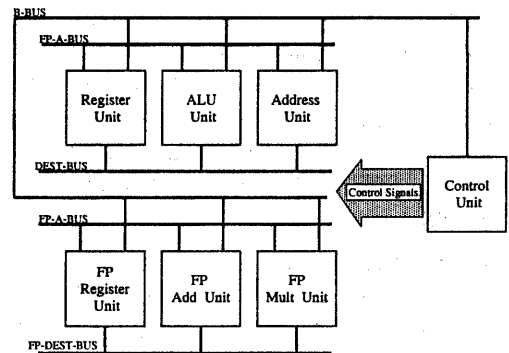


図 6: DLX-FPGA における分割設計

5 DLX-FPGA プロトタイプ・ボードへの実装

DLX-FPGA プロトタイプ・ボードを利用して、DLX-FPGA 浮動小数点演算パイプラインをFPGAへ実装し、動作確認を行った。

5.1 DLX-FPGA プロトタイプ・ボード

本研究で設計したDLX-FPGAを実際にFPGAへ実装し、動作検証を行うための環境として、DLX-FPGA

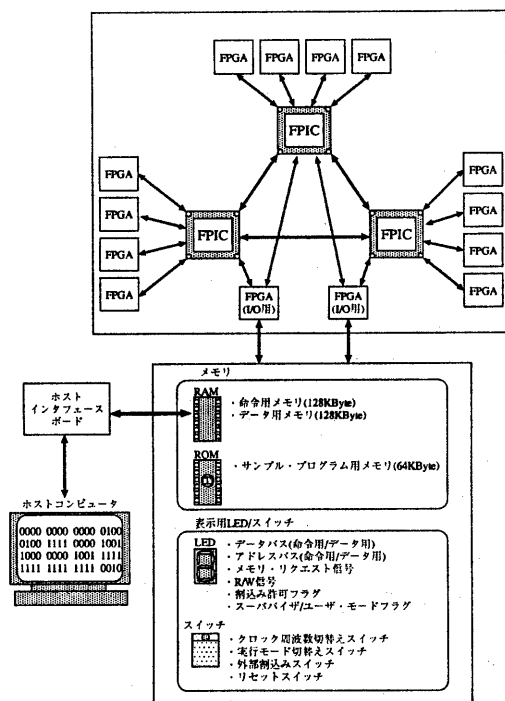


図7: DLX-FPGA プロトタイプ・ボードの概要

プロトタイプ・ボードを作成した。図7にDLX-FPGAプロトタイプ・ボードの構成を示す。DLX-FPGAプロトタイプ・ボードの作成には、複数のFPGAが搭載可能なAptix社のFPCB(Field Programmable Circuit Board, AXB-MP3)を利用した。本プロトタイプ・ボードは14個のFPGA(2個は入出力専用)を搭載しており、約120,000ゲート相当の論理回路を実現することが可能である。そして、これらのFPGAは配線用デバイスである3個のFPIC(Field Programmable Interconnect Component, AX1024)を介して接続される[9]。また、構成データをFPICにダウンロードすることで、利用者がFPGA間の配線を自由に変更できる。そのため、様々な分割実装を試みたり、現在、未使用のFPGAを利用して整数乗除算命令や浮動小数点除算命令を実装することも可能である。

DLX-FPGAプロトタイプ・ボードは、メモリ/表示用ボードを備えており、2個の入出力用FPGAを介してデータの転送が行われる。DLX-FPGAはハーバード・アーキテクチャを採用しているため、命令用メモリとデータ用メモリを用意しており、それぞれに対し同時にアクセス可能である。さらに、初期化用ROMを搭載しており、IPL(Initial Program Loader)スイッチを押すことによりROMの内容を命令用メモリに書き

込むことができる。

また、ホストインタフェース・ボードを介して、ホスト・コンピュータから命令用メモリ、データ用メモリへの読み書きが可能である。これにより、ホスト・コンピュータからプログラムのロードやデータの転送を行うことができる。

表示機能に関しては、命令用、データ用メモリに対するデータバス、アドレスバスが観測可能である。また、DLX-FPGAから出力されるメモリ・リクエスト信号、R/W信号、DLX-FPGAの状態レジスタが持つ割り込み許可フラグ、およびスーパーバイザ/ユーザー・モードフラグを表示できる。さらに、DLX-FPGAの実行動作モードを決定する動作モードスイッチ、クロック周波数切替えスイッチ、リセットスイッチ、外部割り込みスイッチを備えている。

5.2 実装結果

第4章2節で述べた分割案に従い、DLX-FPGA浮動小数点パイプラインの分割設計を行った。その結果を表2に示す。表中のCLB(Configurable Logic Block)は実装デバイスの論理セルであり、()内の値は内部資源の利用率(%)を示している。表2より、DLX-FPGA浮動小数点パイプラインを自然な機能分割で設計でき、4または5個のFPGA(XC4010)を用いることで実装が可能であることを確認した。ここで、FP Register Unitに含まれる浮動小数点レジスタファイルは、回路規模の増大を防ぐためにRAM機能(RAMマクロ)を用いて実現している。

また、表3、表4に浮動小数点パイプライン実装前後のDLX-FPGA整数パイプラインの実装結果を示す。これらの表より、DLX-FPGA整数パイプラインに浮動小数点パイプラインを追加した場合、命令デコードの拡張、およびReorder-Bufferの追加を行う必要があるため、若干回路規模が増大していることが分かる。

これらの結果から、ハードウェア記述言語および回路図入力による設計とともに、浮動小数点パイプライン実装前と同様、DLX-FPGA整数パイプラインを4個のFPGAへ実装できることを確認した。DLX-FPGAプロトタイプ・ボードへの実装完了後、浮動小数点パイプラインの動作確認を行った。

6 おわりに

本稿では、教育用マイクロプロセッサDLX-FPGAに浮動小数点命令を実装するための、DLX-FPGA浮動小数点パイプラインの設計仕様について述べた。また、VHDLおよび回路図入力により設計を行い、DLX-FPGAプロトタイプ・ボードへの実装、ならびに動作確認を行った。その結果、DLX-FPGAに本浮動小数

表 2: DLX-FPGA 浮動小数点パイプラインの設計結果

	FP Register Unit		FP Add Unit-1		FP Add Unit-2		FP Mult Unit-1		FP Mult Unit-2	
	回路図	VHDL	回路図	VHDL	回路図	VHDL	回路図	VHDL	回路図	VHDL
Occupied CLBs	221(55)	226(56)	400(100)	400(100)	---	400(100)	330(82)	400(100)	161(40)	400(100)
Packed CLBs	64(16)	67(16)	350(87)	302(75)	----	396(99)	252(63)	373(93)	130(32)	323(80)
Flip Flops	128(16)	128(16)	177(22)	139(17)	----	87(10)	152(19)	105(13)	64(8)	120(15)
Equivalent "Gate Array" Gates	8,710	8,942	7,004	5,715	---	7,423	9,281	7,055	5,666	6,249

表 3: 浮動小数点パイプライン実装前の DLX-FPGA 整数パイプラインの設計結果

	Register Unit		ALU Unit		Address Unit		Control Unit	
	回路図	VHDL	回路図	VHDL	回路図	VHDL	回路図	VHDL
Occupied CLBs	248(62)	313(78)	384(96)	400(100)	279(69)	318(79)	294(73)	356(89)
Packed CLBs	171(42)	188(47)	234(58)	387(96)	128(32)	144(36)	155(38)	214(53)
Flip Flops	128(16)	128(16)	81(10)	81(10)	256(32)	256(32)	148(18)	157(19)
Equivalent "Gate Array" Gates	9,997	10,047	3,867	7,827	4,019	4,367	3,121	3,777

表 4: 浮動小数点パイプライン実装後の DLX-FPGA 整数パイプラインの設計結果

	Register Unit		ALU Unit		Address Unit		Control Unit	
	回路図	VHDL	回路図	VHDL	回路図	VHDL	回路図	VHDL
Occupied CLBs	303(75)	339(84)	388(97)	398(99)	367(91)	378(94)	394(98)	400(100)
Packed CLBs	171(42)	220(55)	243(60)	397(99)	167(41)	186(46)	242(60)	365(91)
Flip Flops	192(24)	192(24)	98(12)	98(12)	320(40)	320(40)	282(35)	330(41)
Equivalent "Gate Array" Gates	11,343	11,091	4,107	6,001	4,925	5,534	5,232	7,477

点パイプラインを実現した場合、VHDL による設計では、10,000ゲート相当の論理回路を実現可能なFPGAを9個用いて実装が可能であることを確認した。さらに、スキマティック・エディタを用いた回路図入力での設計を行った場合、同様のFPGAを8個用いて実装可能であることを確認した。

DLXの整数乗除算命令は、浮動小数点演算器を利用して実行される。そこで、今後、本研究で実現した浮動小数点演算器を拡張し、整数乗算命令の実装を行う予定である。また、現段階では、DLX-FPGAプロトタイプ・ボード上でさらに4~5個のFPGAが利用可能である。これらのFPGAを利用した浮動小数点除算命令、整数除算命令の実装を検討中である。

謝辞

日頃御討論頂く、本学マイクロ化総合技術センターの田中康一郎助手、柴村英智助手、ならびに本学情報工學部の有田・末吉研究室の諸氏に感謝いたします。

なお、本研究は一部、文部省科学研究費補助金試験研究(B)(課題番号06558043)、および一般研究(C)(課題番号05680278)による。

参考文献

- [1] Hennessy, J.L., and Patterson, D.A., Computer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers, Inc., 1990.
- [2] 中垣, 井上, 久我, 末吉: "上級コース向き教育用マイクロプロセッサ DLX-FPGA の設計と実現," 信学技法 CPSY94-57, 1994.
- [3] Nakagaki, K., Ouchi, M., Inoue, K., Apdulan, B.O., kuga, M., Sueyoshi, T.,: "Design and Implementation of the Educational Microprocessor DLX-FPGA Using VHDL," in Proceedings of the Second Asian Pacific Conference on Hardware Description Languages, pp.147-150, 1994.
- [4] 岩井原ほか: "計算機工学教育一貫教育用マイクロプロセッサ QP-DLX の開発," 情報処理学会研究報告, 93-ARC-100-5, 1993.
- [5] 末吉, 田中, 久我: "教育用マイクロプロセッサ KITE を利用した設計教育の事例報告," 情報処理学会研究報告, 93-ARC-103-12, 1993.
- [6] 末吉 敏則: "教育へのFPGA応用例," 情報処理, Vol.35, No.6, pp.519-529, 1994.
- [7] Xilinx Inc.: The Programmable Logic Data Book, 1994.
- [8] IEEE Inc.: IEEE Standard VHDL Language Reference Manual, 1988.
- [9] Aptix Corporation: System Data Book, 1993.