Open Design Computer Project
—だれでも自由に使用可能なオープンソースコンピューティングシステム—

1. 背景

私たちが使用しているコンピュータシステムは設計がとても古く、現在では不要な命令やハードウェアが多く存在し、プロセッサの物理的規模の増大につながっている。これらは互換性を維持するためであるが、オペレーティングシステムを搭載する上で根本的な部分に大きく関わっている。そして、そもそもこれらのシステムは情報が全てオープンに公開されていないという点がある。

また、私達のように完全オリジナルなプロセッサを設計したり、gcc や binutils の移植などに興味を持っている人多いと思うが、それらの実用的な日本語の資料が全く無いに等しいということもあり、私達が開発を行う中で得たノウ・ハウを WEB 等で公開していきたいと考えた。

2. 目的

私達のように CPU を作ってみたい人や、gcc などの開発ツールの移植に興味がある方々にも、そのノウ・ハウをウェブページに公開しているため、その方々の手助けになるとも思われる。

オープンソースで完全に商用利用が可能なコンピュータシステムの設計・構築を行い、誰でも使用できるようなドキュメントとハードウェアを含めた環境を整える。そして私達と同じような興味を持っている人向けに、参考になるようなノウ・ハウなどを公開することである。

3. 開発の内容

3. 1. 開発内容の全体

本プロジェクトは完全にゼロからコンピュータシステムを構築する。その範囲は独自にプロセッサアーキテクチャの考案、その仕様に対応するプロセッサの実装、そしてそれらに接続される周辺デバイスである。開発ツールは gcc と binutils の移植を行い、開発ツールとハードウェアの検証のためにプロセッサシミュレータの開発した。これらの統合的なコンピューティング・開発環境を手軽に利用するためのターゲットハードウェアボードの開発も行った。

3. 2. 独自のプロセッサアーキテクチャの設計

今回独自に命令セットアーキテクチャから設計を行ったプロセッサアーキテクチャを MIST32 アーキテクチャと名付けた。名前の通り 32bit の RISC プロセッサであり、命令セットレベルでの並列化を行うことを考慮した命令セットを持っていて、比較的高いパフォーマンスを維持しながらも小型なハードウェア規模で実装可能で、それにより低消費電力で動作するように設計してある。

MIST32 アーキテクチャはプロセッサの命令のみの定義ではなく、それは MMU や IO、割り込みの取り扱いについても厳密に定義しており、オペレーティングシステムの搭載を大前提とするアーキテクチャである。

3. 3プロセッサの開発

考案した MIST32 アーキテクチャを実際にプロセッサとして実装した。今回は、パフォーマンス重視の比較的大規模な MIST1032SA プロセッサと、低規模で低消費電力性能を重視した MIST1032ISA の 2 種類のプロセッサを開発した。これらはどちらも Verilog HDL で書いており、BSD ライセンスで WEB 上にて公開している。

MIST1032SA プロセッサは Tomasulo のアルゴリズムを基本とし、レジスタリネーミングを用いた最大 2 イシュー・4 命令並列実行可能なスーパースカラプロセッサである。アウトオブオーダ実行で片方向の分岐予測により投機的実行も行なっている。パイプラインは 8 段で構成されていて、その詳細を図 1 に示す。MIST32 アーキテクチャはもともと命令間の依存性を減らし、命令レベルの並列化を追求したアーキテクチャなので、アウトオブオーダ実行を行う MIST1032SA は、MIST32 アーキテクチャの利点を活かすことができる。しかし、この

ような利点を活かす方式で実装すると回路規模が巨大化し、比較的大規模な FPGA 等がターゲットとなる。

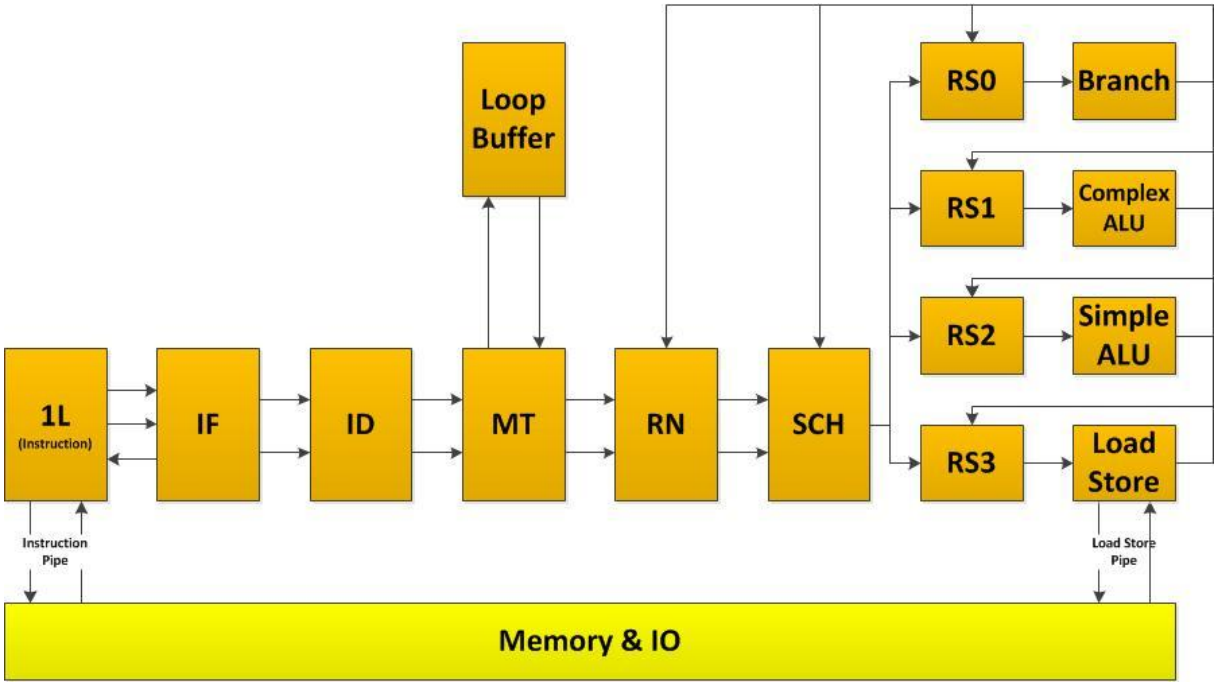

図 1 MIST1032SA のパイプライン

MIST1032ISA プロセッサはシングルパイプラインのシンプルなパイプラインを持つプロセッサである。MIST1032SA の約 1/9 のハードウェア規模で構成され、ローエンドの FPGA にも十分搭載可能なハードウェア規模に収まっている。詳細なパイプラインを図 2 に示す。このプロセッサは MIST32 アーキテクチャの利点を生かしきれているとは言えないが、MIST32 アーキテクチャの環境を安価に構築するのに適しているといえる。

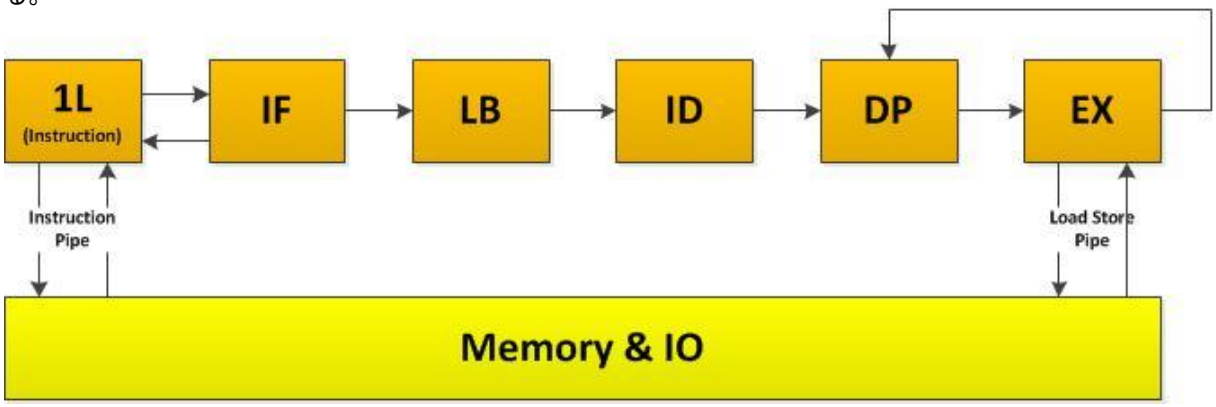

図 2 MIST1032ISA のパイプライン

3. 3ターゲットボードの開発

MIST32 アーキテクチャのソースコードなどは公開しているが、ハードウェアという特性上使用する FPGA によって特定のモジュールを書き換えたり、変更する必要がある。これはある程度の知識がないと行えないため、誰でも使える環境とは言いがたい。そのためターゲットボードを提供することで、すぐに使用出来るハードウェア環境を整えたいと考え、専用ボードを開発した。

新規に製作しなくとも汎用的に使用出来る FPGA は存在するが、コンピュータとして使用するためには RTC やディスプレイ、複数のメモリが必要であり、この条件を満たしながら低価格なボードが存在しなかったので開発した。

図 3 開発した MIST32 ターゲット FPGA ボード

4. 従来の技術(または機能)との相違

(開発したソフトウェアの新規性、類似のソフトウェアと比較した場合の優位性等を記述)

今回、命令セットレベルでプロセッサを設計した理由としては既存のプロセッサにおける不要な命令やハードウェアを一新すると共に、命令レベル並列化を意識した設計を行いたいからである。前者は主にハードウェア規模を小さくするためと、命令のみで実現していたものをハードウェアで行うことで高速化することを目的としていて、後者はレジスタリネーミング付き Out-Of-Order 実行に対して命令レベルで最適化することで並列度を上げることを目的としている。

MIST32 アーキテクチャは 32 本の汎用レジスタを持つが、それによってタスクスイッチのオーバーヘッドが大きくなる。そのため、コンテキストスイッチをハードウェアで行ったり、優先度の高いタスクに対してプロセッサが自動的にキャッシュの特定の領域を自動的に乗っ取り、タスクのコンテキストを退避するという仕組みを用いる。また、割り込みも割り込みレベルというのを設けて、IRQ をすぐに発行せず、ハードウェアポーリングモジュールを用いたタイミング制御を行うことで1つのタスクの実行時間を長く確保するという方式を新規に採用している。x86 プロセッサにおいても、ハードウェアでタスクスイッチを行う機能が提供されているが、一部のレジスタが保存されないため最新の Linux などにおいてはその機能がフル活用されていないという問題がある。MIST32 では必要なすべてのレジスタのと、今後追加されるであろうレジスタにも備え、十分な予約領域も備えている。

命令レベルの並列化に対する最適化については、主にフラグレジスタの取り扱いを従来のプロセッサとは全く違う方式を用いている。従来のプロセッサではフラグを更新する命令とフラグを更新しない命令から構成されるのが一般的である。MIST32 ではすべての命令が基本的にフラグを更新するというスタイルを用いている。これはつまり、フラグレジスタのみで考えると現在実行している命令は必ず1つ前の命令にのみ依存することになり、従来のようにずっと前生成されたフラグを参照することはありえないため、アウトオブオーダ実行を行うときにフラグレジスタにおける命令間の依存性緩和され、並列度を大幅に向上することができる。

私達のプロジェクトと類似するプロジェクトとしては Open RISC がある。このプロジェクトも独自に設計した命令セットアーキテクチャのプロセッサを製作し、その開発環境も整えてあるプロジェクトである。Open RISC のハードウェアのライセンス体系は LGPL で、製品化などをすることを考えるとライセンスが柔軟とはいえない。その点私達のプロジェクトでは BSD で公開しているため、製品化に対する障害が少なく、改変や部分的利用も自由である。また、私たちはソースコード以外にも、すべてのマニュアルを WEB で公開し、開発ツールの移植に関するノウ・ハウなども日本語で公開している。

5. 期待される効果

ハードウェアの仕様とその実装、開発ツールとすべての開発環境と、全て揃ったものをオープンソースで公開するので、利用したいと思った人が簡単に使用出来る環境ができている。ハードウェアは BSD で公開するためそのまま製品に組み込んだり、一部を改変して組み込むなど柔軟な利用形態が考えられる。

私達のように CPU を作ってみたい人や、gcc などの開発ツールの移植に興味がある方々にも、そのノウ・ハウをウェブページに公開しているため、その方々の手助けになるとも思われる。

6. 普及(または活用)の見通し

(開発成果に関する利用者の具体的なイメージ[例えば、利用者数など]を可能な限り定量的に記載)

今回製作したシステムは、組み込みではなく、オペレーティングシステムが搭載前提のパフォーマンスを必要とする分野をターゲットとしている。従来こういったプラットフォームでは x86(x86-64)や ARM 等のアーキテクチャが用いられていたが、近年は低消費電力及びハードウェアの小型化により、従来の組み込み系プロセッサと比較的大規模なシステムに使われる汎用プロセッサとの間を埋める消費電力・性能を埋めるプロセッサが求められている。組み込み系プロセッサと汎用プロセッサの間を埋める特徴を持ったプロセッサとして、まずは FPGA 上にプロセッサを構築し開発環境を整えてその性能・特性を評価した上で、実際の製品化へ生かしていければ良いと考えている。FPGA のソフトコアプロセッサとしても、完全にオープンソースかつ BSD ライセンスで実装が公開されているものはなかなか無いために、これらの需要も見込まれる。

7. クリエータ名(所属)

チーフクリエータ : 伊藤剛浩(筑波大学情報学群情報科学類)
コクリエータ : 川田裕貴(筑波大学情報学群情報科学類)

(参考)関連URL
http://open-arch.org

報告集 http://www.ipa.go.jp/about/jigyoseika/10fy-pro/main.html