

プロセッサ設計におけるデザインパターンの利用の検討

安倍 厚志† 山崎 勝弘†

立命館大学大学院 理工学研究科†

1. はじめに

半導体の高集積化が進む中、システム LSI へ求められる機能は多様化しており、ハードとソフト両方の知識を有する技術者が求められている。近年、回路規模増加と設計期間の短縮による開発労力が大きなボトルネックになっており、設計期間の短縮に伴い回路を最初から記述していたのでは設計者の負担が大きい。このため、ハードウェア設計において、ハードウェア IP を用いた設計資産の利用が進められている。本研究では、プロセッサ設計においてデザインパターンを利用することにより、理解度の向上と設計期間の短縮を目指す。デザインパターンは、設計仕様、データパス、機能ブロック、評価から構成される。機能ブロック毎に IP を提供し、ソフト IP を拡張することにより、容易に回路を設計することを目指す。本稿では、デザインパターンの概要、及び過去に作られた RISC 型プロセッサのデザインパターン化とその使用法について検討する。

2. ハード/ソフト協調学習システム

2.1 システムの構成

ハード/ソフト協調学習システムとは、学習者がプロセッサをソフトウェアとハードウェアの両面から設計することで、両者の理解を協調的に進め、コンピュータシステムを体系的に学習する教育システムである。本システムは、MIPS のサブセットである MONI を使って、アセンブリプログラム、及び MONI プロセッサで実機検証することで、ハードウェア学習を行うことができる。MONI は語長 16bit で 4 つの命令形式 (R, I5, I8, J) を持ち、全 43 命令、3 オペランド方式である。本システムは、プロセッサ学習システムとプロセッサ設計支援ツールから構成される[1][2]。後者は命令セットの設計を行う命令セット定義ツール、命令セットシミュレータ、命令セットアセンブリ、FPGA 上でプロセッサのデバッグを行うプロセッサデバッガ、及び PC 上でデバッグの行えるプロセッサモニタから構成されている。

2.2 学習の流れ

ソフトウェア学習では、MONI シミュレータでシミュレーションを行い、MONI プロセッサの構造と動作を理解する。次に、ハードウェア学習では、HDL を用いて MONI シングルサイクルプロセッサの設計を行う。シミュレーションを行い、実機検証することでプロセッサ設計能力を習得する。さらに、学習者独自の命令セットを定義し、プロセッサ設計を行う。定義する際に命令セット定義ツールと命令セットシミュレータが用意してあり、デバッグ、シミュレートしながら命令セットの有効性を検証する。独自のプロセッサ回路記述を行い、プロセッサモニタとデバッガを用いて、実機上のデバッグができる。

Design Patterns for Processor Design,
Atsushi Abe and Katsuhiro Yamazaki, Graduate School of
Science and Engineering, Ritsumeikan University.

3. プロセッサ設計におけるデザインパターン

3.1 デザインパターン

デザインパターンとは、仕様を再利用するために、構造のノウハウを設計見本の形で表したものである。過去の設計事例を類型化し、見本化することで設計対象が持つ性質とデザインパターンとの関連性を見つけることができ、関連するデザインパターンを参考にすることによって、設計対象を設計する際の設計指針が予めわかる[3]。本研究では、学習者が設計したプロセッサをデザインパターン化する。すなわち、プロセッサのデータと制御の流れをデータパスで表示し、設計仕様、評価、IP などを提示することで、設計時の再利用とプロセッサアーキテクチャ理解度の向上を目指す。

3.2 デザインパターンの仕様と実行モデル

学習者は一度ハード/ソフト学習システムを通して、プロセッサの設計を行う。既存のシステムでは、シングルサイクル、マルチサイクル、パイプライン化は学習者が最初から設計するので多くの時間がかかる。デザインパターンとして以下のものを提供する。

(1) 設計仕様

回路における機能ブロックのパラメータ、設計概念、ヘッダファイルを持たせる。入出力のビット幅は、ヘッダファイルによって変更が行えるようにする。

(2) データパス

全体の制御線を含めた静的データパスと、命令毎のデータの流れと制御を示す動的データパスを持たせる。

(3) 機能ブロックの IP

ソフト IP とハード IP の 2 種類を提供する。ソフト IP はソースコードにあたり、ハード IP はソースコードをコンパイルしたイメージとなる。ソフト IP では、学習者がインターフェース部分を修正することで、設計している回路へ必要な機能を追加することができる。ハード IP では、機能が固定であるので、学習者はハード IP に合わせた設計を行わなければならない。

(4) 評価

回路に対して設計者が行った検証結果（回路規模など）と改良点を示す。上記のものをデザインパターンとして提供することで、学習者がどのようなプロセッサを作成したいか、回路全体の制御の流れの把握、回路規模の予想や回路の最適化への参考資料となる。また、設計時に IP を再利用することで効率よく設計することが可能である。

次に、ハード/ソフト協調学習システムにおける実行モデルを図 1 に示す。アーキテクチャを理解し、学習者が MONI シングルサイクルプロセッサを HDL 設計するとき、データパスを参考にして、制御の流れ、使われている機能ブロックの理解と把握ができる。また、HDL による独自のプロセッサ設計へ移行する前にデザインパターンを参考にして、マルチサイクル、パイプライン処理の理解

を深めることができ、必要な機能ブロックの選定を行うことでプロセッサの回路設計がスムーズに行える。さらに、プロセッサ設計の段階で IP を用いることで、学習者は自分の好きなように機能を追加することができる。機能の追加は、入力数の増加、ポート名の変更、ビット幅の変更、制御信号の追加を行うことができ、設計時間の短縮が期待できる。

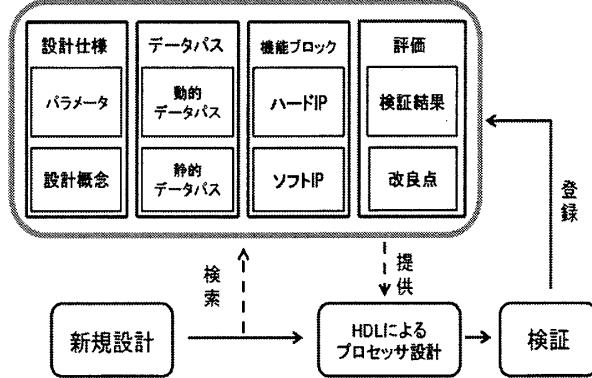


図1：学習システムにおけるデザインパターンの実行モデル

3.3 デザインパターンの再利用

本研究では、設計されたプロセッサをデザインパターンとして再利用するために、図1に示すように設計仕様、データパス、機能ブロック、評価に分けて、取り出す。それらをデザインパターンとして新たに登録することで、後の学習者へのデザインパターンになる。

機能ブロックの IP 化では、プロセッサ設計において、機能追加が行いやすような形でソフト IP を提供できるようにし、ハード IP は、プロセッサ設計において、固定化された機能ブロック IP として再利用を目指す。

4. デザインパターンの使用例

MONI の命令形式を図2に、R 形式命令の回路に、他の命令の回路を追加する例を図3に示す。図3で太線は(a)ではデータの流れを、(b)～(d)では新規に追加した分を示す。まず、学習者は、静的データパスと動的データパスを観察して、必要な制御線やデータの流れでアーキテクチャの理解を深める。次にデータパスを参考として、デザインパターンより必要な機能ブロックを検索する。今回 R 形式を満たすデザインパターンがあった場合、学習者はヘッダファイル、R 形式実現に必要な各モジュール (PC, RF, ALU) を入手できる。

さらに、学習者は入手したデータパスを参考として各モジュールを配線することで回路を作成できる。I5 形式へ拡張する場合、マルチプレクサは 3 つ必要である。ここでは、ソフト IP として 2 入力 1 出力マルチプレクサを作成して自分の回路に組み込むことができる。次に、I8 形式へ拡張する場合、条件分岐は相対アドレスで行うので、現在の PC のアドレスと即値の和によって分岐先のアドレスを計算する。すなわち、8bit から 16bit への符号拡張とアドレス計算を行う機能ブロックが必要となる。J 形式への拡張は、分岐先として、絶対アドレスを用いているので、マルチプレクサの選択信号を 2bit へ拡張する必要がある。以上のように R 形式回路から、I5, I8, J 形式への拡張が容易にできる。

R形式	5	3	3	3	2
I5形式	Opcode	Rs	Rt	Rd	Fn
I8形式	Opcode	Rs	Rt	Immediate	
J形式	Opcode	Rs	Immediate/Address		
	Opcode	Target absolute address			

図2：MONI の命令形式

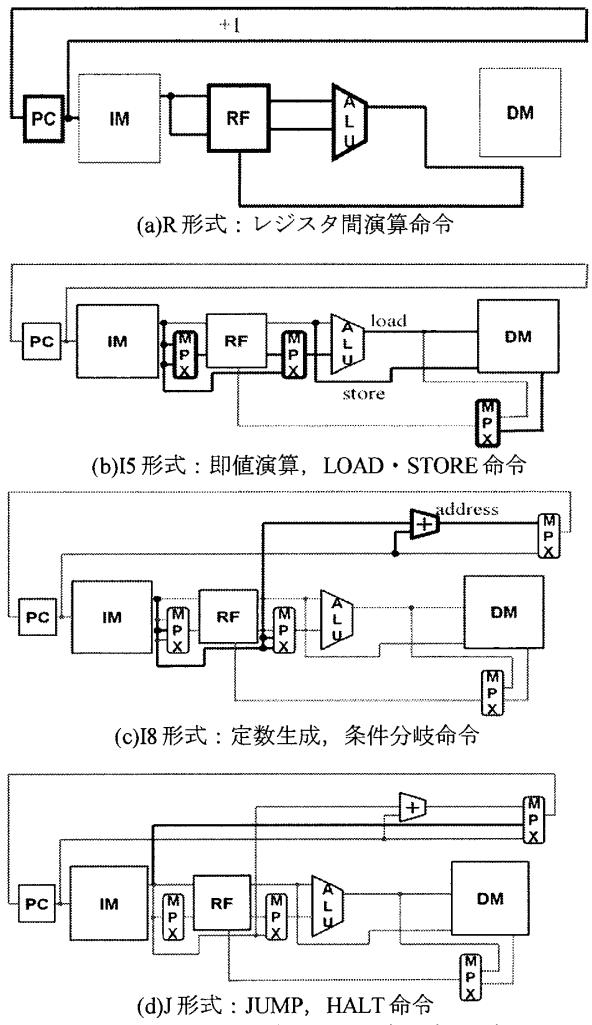


図3：R 形式回路から他形式回路への拡張

5. おわりに

本稿では、デザインパターンを用いたプロセッサ設計における仕様、機能ブロックの IP 化、再利用、及びデザインパターンの使用例について述べた。今後、提案したツールの実装、評価を行う。

参考文献

- [1] 難波, 他 : プロセッサ設計支援ツールの設計・実装とハード/ソフト協調学習システムの評価, FIT2007, LC-002, 2007.
- [2] 志水, 他 : プロセッサ設計教育における命令セット定義ルールと命令セットシミュレータの試作, 情報処理学会, 関西支部大会, A-05, 2008.
- [3] 石井忠俊 : D4 章システムアーキテクチャ設計技術 5, STARC, 2008.