C-009

学生によるプロセッサ設計実験に基づいたハード/ソフト協調学習システムの評価

Evaluation of a Hardware / Software Co-learning System based on Processor Design Experiments

井手 純一† 志水 建太† 山崎 勝弘† 小柳 滋†

Junichi Ide Kenta Shimizu Katsuhiro Yamazaki Shigeru Oyanagi

1. はじめに

半導体の高集積化に伴い，システムLSIに求められる機能が多様化してきている．その中でシステムの性能や制約に従ってプロセッサの命令の追加や構成をカスタマイズすることが多く，ハードウェアとソフトウェア両方の知識やプロセッサにおける命令セットとマイクロアーキテクチャの知識が必要である．このような背景から，本研究室では学習者が命令セットとマイクロアーキテクチャを設計し，その境界を学習できるハード/ソフト協調学習システム（Hardware Software Co-learning System:HSCS）を開発している[4]-[10]．また，昨年度，プロセッサ設計支援ツール（命令セット定義ツール(Instruction Set Information Define:ISID)，ISIDアセンブラ，プロセッサデバッガ・モニタ）の設計と実装を行った[10]．本システムの特徴は，学生が独自の命令セットを定義し，設計したプロセッサをFPGAボード上で実機検証を行うことで，ハードウェアとソフトウェアのトレードオフを考察できる点である．本論文では，本研究室の2007年度の4回生が本システムを用いて独自の命令セットプロセッサの設計を行い，2006年度のプロセッサの設計実験と比較して，学習効果の検証について述べる．また，プロセッサ設計支援ツール(命令セット定義ツール，ISIDアセンブラ，プロセッサデバッガ・モニタ)とシステム全体の評価を行う．

2. ハード/ソフト協調学習システム

2.1 システムの概要

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設計を通してソフトウェアとハードウェアの知識を習得し，両者のトレードオフを理解していくことを目的とした教育システムである．図1にHSCSの構成と学習フローを示す．HSCSは，MIPSのサブセットとして定義したMONI[6]命令セットを用いたアセンブリプログラミングとプロセッサ設計を行うプロセッサ学習システム，及び学習者が独自の命令セットとプロセッサを設計するプロセッサ設計支援ツールで構成されている．まず，学習者はMONI命令セットを用いてアセンブリプログラミングを行い，MONIシミュレータで実行する．これにより，プログラムのデバッグや各命令によるプロセッサの動作が理解でき，アセンブリプログラミング技術とプロセッサアーキテクチャの知識を習得する．次に，実際にMONIプロセッサをHDLで設計しFPGAボードコンピュータ上で動作検証して，プロセッサの基本的な設計能力を習得する．さらに，学習者は命令セット定義ツールを用いてMONI命令セットの拡張や，独自の命令セットを定義し，ISIDシミュレータを用いて定義した命令セットのアセンブリプログラムをデバッグし，命令セットが有効か検証する．そして，学習者は定義した命令セットのプロセッサ設計を行う．実機上での検証はプロセッサデバッガ・モニタを使用する．プロセッサデバッガによりプロセッサのデバッグを容易にし，プロセッサモニタはPC上での検証を可能にする．以上のように，ソフトウェアとハードウェアの学習を進めていく．

本研究室では，2004年度から4回生の卒業研究の導入実習としてHSCSを利用したアセンブリプログラミングと，プロセッサ設計と検証を行い，HSCSの評価を行っている．

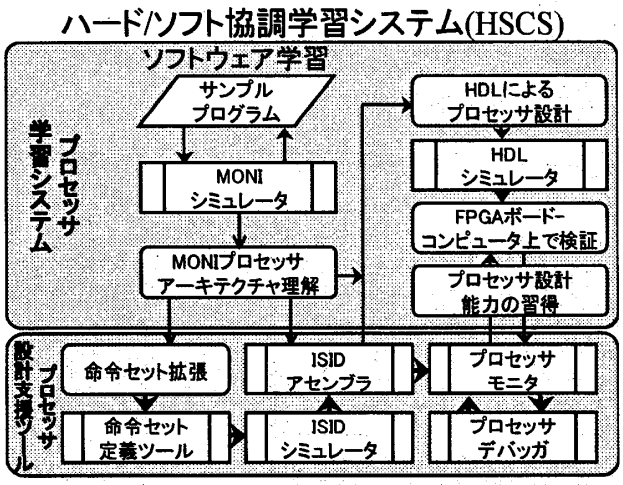

図1 ハード/ソフト協調学習システムの学習フロー

2.2 プロセッサ設計支援ツール

プロセッサ設計支援ツールの役割は，命令セットの定義とプロセッサのデバッグのサポートの二つである．学習者が独自のプロセッサ設計をするためには命令セットの定義と検証を繰り返し行える環境が必要である．命令セット定義ツールでは，フィールド，フォーマット，命令や動作，疑似命令などを入力することで，新たな命令セットを定義できる．ISIDアセンブラは，異なる命令セットで記述されたプログラムのアセンブルが可能である．ISIDシミュレータを用いることで，プロセッサ設計に入る前の段階で，プログラムのデバッグが容易になり，命令セットの評価も行える．

実機上でプロセッサのデバッグを行うために，プロセッサデバッガ・モニタを開発した．図2にプロセッサデバッガ・モニタの構成を示す．学習者が設計したプロセッサをプロセッサデバッガと接続することで，プロセッサの実行や停止，メモリやレジスタのデータの更新などが可能となる．設計するプロセッサのトップモジュールは統一しているため，一つのプロセッサデバッガで様々なプロセッサが検証可能である．プロセッサモニタはホストPC上で実機上のプロセッサの制御とデバッグを行うために，学習者が

†立命館大学大学院 理工学研究科，Graduate School of Science and Engineering, Ritsumeikan University.

入力したデバッグコマンドを解釈し，プロセッサデバッガへ指示を与える．表 1 にデバッグコマンドを示す．表 1 の他にプロセッサを停止する halt，プロセッサをリセットする rst，デバッグコマンドの内容を表示する help，およびプロセッサモニタを終了させる exit のコマンドがある．

表 1　デバッグコマンド

コマンド	ターゲット	意味
write	dm / im / rf	メモリ・レジスタの書き込み
read	dm / im / rf / pc	メモリ・レジスタの読み出し
save	dm / im / rf / pc / bp	メモリ・レジスタ内容を保存
load	dm / im / rf / bp	ファイルからロード
set	pc / bp	PC・ブレイクポイント設定
del	bp	ブレイクポイント削除
list / init	dm / im / rf / pc / bp	メモリ・レジスタの表示 / 初期化
run all		通常実行
run clk N		N クロック実行
run bp		ブレイク実行

dm:データメモリ im:命令メモリ rf:レジスタファイル
pc:プログラムカウンタ bp:ブレイクポイント

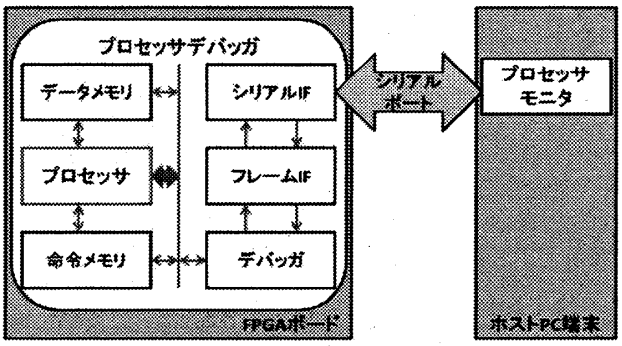

図 2　プロセッサデバッガ・モニタの構成

3. 学生によるプロセッサ設計

3.1 命令セットの定義

ハード/ソフト協調学習システムを評価するにあたり，シングルサイクルとマルチサイクルの 2 つのマイクロプロセッサ SARIS を設計した．命令長は 16bit 固定，3 オペランド方式，全 22 命令で 4 つの命令形式を持ち，80 命令まで拡張できる．

表 2 に SARIS の命令形式を示す．RR 形式でレジスタ間の演算命令を，RI 形式でレジスタ値と即値演算を行う命令を定義する．また，I8 形式で条件分岐命令とメモリ・レジスタへのデータ転送命令を，J 形式で JUMP，NOP，HALT 命令を定義している．命令デコードの効率化を図るため op フィールドの上位 2 ビットを形式別に使用し，残り 3 ビットで命令の種類を定義している．J 形式，I8 形式は 8 命令，RR 形式と RI 形式は，フィールド fn で命令を詳細に識別しており，それぞれ 32 命令まで拡張できる．

3.2 プロセッサ設計と検証

プロセッサの設計規模と最大遅延を表 3 に，各プログラムにおけるクロックサイクル数を表 4 に示す．4 段マルチでは J 形式命令，条件分岐命令，ST 命令を 3 クロック，その他は 4 クロックで実行する．設計したプロセッサを HDL シミュレータと FPGA ボード上で検証した．4 段マルチの CPI はシングルの約 3.7 倍という結果が得られた．表 3 の最大遅延より，シングルでの最高動作周波数は 21.2MHz，4 段マルチでは 66.3MHz となり，4 段マルチはシングルの約 3.1 倍の動作周波数である．シングルの CPI は 1 で，4 段マルチの CPI が約 3.7 という結果から実質シングルの方が処理が速い結果となった．4 段マルチの性能を向上させるためには，各モジュールの設計を最適化し、遅延時間を短くして，より高い動作周波数を得ることが必要である．

表 2　SARIS の命令フォーマット

	5	2	3	3	3
RR	op	fn	rs	rt	rd
RI	op	fn	imm	rt	rd
I8	op	immediate			(rd)
J	op	immediate			

表 3　SARIS プロセッサの設計規模と最大遅延

サイクル	スライス数	LUT 数	FF 数	最大遅延 (ns)
シングル	396	692	140	47.42
4 段マルチ	398	744	239	15.077

表 4　各プログラムにおけるクロックサイクル数

SARIS	N までの和	階乗	二次方程式の根の判別
シングル	44	51	73
4 段マルチ	162	189	268

4. ハード/ソフト協調学習システムの評価

4.1 プロセッサ設計支援ツールの評価

命令セット定義ツールと ISID アセンブラを用いることで，検証プログラムを機械語に変換する時間が短縮でき，プロセッサのデバッグを行うだけになり，全体の検証時間とデバッグ時間の短縮ができた．実際に検証プログラムを手書きで機械語に変換し，HDL シミュレータや FPGA ボード上の検証でエラーが発生すると，変換した際の間違いかプロセッサ設計のミスによるエラーか分からないためである．

プロセッサデバッガ・モニタを用いることで，実機上でのデバッグが容易に行えた．表 1 の run clk N，read pc，read rf コマンドで 1 命令実行が可能になり，バグの発見が容易であった．また，HDL シミュレータで正しい結果を得ていても実機上では正しい動作をしない場合があり，set bp，run bp コマンドを用いることでエラー箇所の特定ができた．

4.2 システム全体の評価

MONI 設計者を A，SARIS 設計者を B とする．表 5 に A，B の設計したプロセッサの実装結果，表 6 に A，B のソフトウェアとハードウェアの学習時間を示す．A も B も MIPS アーキテクチャの授業を受けており，A は 2006 年度に，B は 2007 年度に HSCS を用いてソフトウェア学習を行い，ハードウェア学習でそれぞれプロセッサ設計を行った．

A は 8 個のプログラム（整数の階乗・三角形の種類判定・二次方程式の根の判別・素数判定・符号付き整数の商と剰余・直線の方程式・8×8 行列引き算/掛け算/足し算・BCD コード加算/減算）を 25 時間かけて学習し，2 種類のアーキテクチャのプロセッサを平均 21 時間で設計できた．B も 8 個のプログラム（N までの和，乗算，除算，階乗，素数判定，二次方程式の根の判別，最大公約数，バブルソート）を 18 時間かけて学習し，命令セットの定義から独自のプロセッサを設計できた．A はアセンブリプログラミング時に用いた MONI シミュレータ上に表示されているアーキテクチャの図によって，回路イメージが持てたことが設計時間の短縮になったと考えられる．B は一度 MONI のシングルサイクルプロセッサを設計しているため，SARIS の設計がスムーズに行えた．独自の命令セットを考案し設計できたので，本システムが有効であると言える．プロセッサデバッガ・モニタを用いた実機上の検証は，A は表 1 のコマンドの write，load，list / init の一部，及び run all が使用でき，B は全てのコマンドを使用できた．B は実機上でのデバッグ時間が減ったため，プロセッサデバッガ・モニタの効果が大きいと考えられる．

表 5　プロセッサの実装結果

プロセッサ	サイクル	スライス数	FPGA 使用率 (%)	最高動作周波数 (MHz)
MONI	シングル	578	30	21.4
	4 段マルチ	783	40	51.6
SARIS	シングル	396	19	21.2
	4 段マルチ	398	20	66.3

表 6　ソフトウェアとハードウェアの学習時間

単位：時間

設計者	A		B	
プロセッサ	MONI		SARIS	
サイクル	シングル	4 段	シングル	4 段
MONI アセンブリプログラミング	13		12	
MONI シミュレータ上のデバッグ	12		6	
ソフトウェア学習合計	25		18	
命令セット定義	0		10	0
アーキテクチャ設計	1	3	5	20
HDL 設計	24	38	35	60
HDL シミュレータ上のデバッグ	14	30	40	35
プロセッサデバッガ・モニタを用いたデバッグ	25	48	15	10
ハードウェア学習合計	64	119	105	125

5. おわりに

ハード/ソフト協調学習システムを用いて，2007 年度の 4 回生が命令セットを考案し，プロセッサを設計した．本論文では，2006 年度の 4 回生によるソフトウェア学習，ハードウェア学習と比較して，プロセッサ設計支援ツールとシステム全体の評価について述べた．ソフトウェア学習の効果により，MONI プロセッサを比較的早く設計でき，独自の命令セットを考案し設計することもできた．また，二人の学習者は複数のアーキテクチャを設計することができた．命令セット定義ツールと ISID アセンブラを用いると，検証プログラムを手書きで機械語に変換する時間がなくなり，全体の検証時間とデバッグ時間を短縮できる．プロセッサデバッガ・モニタを用いた実機上での検証では，デバッグコマンドを全て使用すると，プロセッサのデバッグ時間の短縮につながる．

今後の課題は，命令セット定義ツール，ISID シミュレータ・アセンブラを用いて命令セットの評価を行うことや，HSCS の検証を多くの学習者に行ってもらうことである．将来的には，本システムを学部の学生実験で利用することが目標である．

参考文献

[1] 末吉，他：KITE マイクロプロセッサによる計算機工学教育支援システム，IEICE，Vol.J84-D-1，No.6，pp.917-926，2001.

[2] 西村，他：教育用パイプライン処理マイクロプロセッサ PICO^2 の開発，IPSJ，Vol.2000，No.2，pp.141-148，2000.

[3] 高橋，他：FPGA を用いたスーパースカラ設計教育に関する一考察，IPSJ，Vol.2003，No.49，pp.43-50，2003.

[4] 池田，他：ハード/ソフト・コラーニングシステムにおける FPGA ボードコンピュータの設計，情報処理学会，第 66 回全国大会論文集，5T-5，2004.

[5] 大八木，他：ハード/ソフト・コラーニングシステムにおけるアーキテクチャ選択可能なプロセッサシミュレータの設計，情報処理学会，第 66 回全国大会論文集，5T-6，2004.

[6] 中村，他：プロセッサアーキテクチャ教育用 FPGA ボードコンピュータシステムの開発，FIT2004，LC-008，2004.

[7] 難波，他：マイクロプロセッサの設計と検証に基づいたハード/ソフト・コラーニングシステムの拡張，FIT2005，LC-002，2005.

[8] Hoang Anh Tuan，et al.:A FPGA Based Hardware / Software Co-learning System，IEICE Technical Report，RECONF 2005-48，pp.43-48，2005.

[9] 難波，他：ハード/ソフト協調学習システムのための命令セット定義ツールとプロセッサデバッガの開発，FIT2006，N-009，2006.

[10] 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.