A-09

パイプラインプロセッサを理解するための教材　RUE-CHIP1

RUE-CHIP1 Processor: Teaching Material for Understanding a Pipeline Processor

神原弘之†　金城良太‡　戸田勇希§　矢野正治¶　小柳滋‖
Hiroyuki Kanbara Ryota Kinjo Yuki Toda Masaharu Yano Shigeru Oyanagi

１．はじめに

MIPS32 命令セットの 5 段パイプラインプロセッサの動作を、学生実験等での実習を通じてより深く理解することを目的に、Ritsumeikan University Education Chip – 1 (RUE-CHIP1) プロセッサを開発した。開発の背景と目標、RUE-CHIP1 プロセッサとその動作を観測／制御するための専用ボードについて以下で説明する。さらに立命館大学情報理工学部で予定されている RUE-CHIP1 プロセッサを用いた学生実験の目的と実施の概要について報告する。

２．RUE-CHIP1 プロセッサへの要求

2.1　開発の背景

今日の携帯電話に代表される組込み機器は、動画の撮影や再生、インターネットに接続しての WWW の表示といった、複雑な機能をサポートすることを要求されている。バッテリで駆動される機器で複雑な計算処理を効率よく実施するため、性能／消費電力比にすぐれた RISC 形式のパイプラインプロセッサが数多く組込み機器に搭載されるようになった。

組込み機器のハードウェアの中枢となった RISC 型パイプラインプロセッサの動作原理については、パターソン＆ヘネシー著の「コンピュータの構成と設計」中で、MIPS32 命令セットの 5 段パイプラインプロセッサを例題として詳しく解説され、計算機アーキテクチャ教育のテキストによく用いられている[1]。

この MIPS32 命令セットの理解を助けるための、命令レベルのエミュレーションを行うソフトウェアがいくつか公開されている。これらのエミュレータを用いてアセンブリ記述プログラムの実行結果を確認することで、MIPS32 命令セットを理解させる学生実験が幅広く行われている。また MIPS プロセッサを搭載した組込み用のボードコンピュータを gdb スタブを介して PC から制御することでも、プログラムの実行を観測することも考えられる。

一方、「コンピュータの構成と設計」で詳細に解説されている 5 段パイプラインの動作は複雑であり、エミュレータを用いての命令の解釈実行の理解と比べて、難度が高い。いくつかの大学では、学生の理解を助けるため、「コンピュータの構成と設計」のパイプラインプロセッサ構造を FPGA 等でそのまま再現して、内部の状態を観測する学生実験が行われている。

† （財）京都高度技術研究所、 ASTEM RI
‡　京都大学 エネルギー理工学研究所、Kyoto Univ.
§　関西学院大学 大学院、Kwansei Gakuin Univ.
¶　京都大学 大学院、Kyoto Univ.
‖　立命館大学 情報理工学部、Ritsumeikan Univ.

このような補助教材はパイプラインの動作を理解するのに有効であるが、前述したエミュレータと比べて、プログラムを実行させた時の各命令で行われる処理を理解することは困難になる。なぜなら、パイプライン処理では、１命令の実行に数クロックを要し、その数クロックの間に後続の命令の処理の一部が並行して実行される。レジスタあるいはメモリへの書込みが完了して、ある命令の実行が反映されたことを確認できた時には、次の命令の解釈実行はすでに途中まで進んでいる。このため各命令単位でどのような実行が行われているか、区別しにくい。またフォワーディング機構により、汎用レジスタの値が更新される前に直前の命令の演算結果が次の命令に正しく伝えられる。これは外部からの観測では、汎用レジスタの値と異なる値が、次の命令でのレジスタ参照に反映されているように見えてしまい、命令単位での動作の理解をより一層困難としている。

2.2　開発の目標

このような課題のもと、以下の２つの目的を両立させる教材とすることを狙いとして、RUE-CHIP1 プロセッサを開発した。

(1) パイプラインプロセッサの内部動作の理解の支援
(2) パイプラインプロセッサの各命令の理解の支援

「(1) パイプラインプロセッサの内部動作」については
（a）パイプラインの各ステージで行われる一連の処理の組み合わせにより正しく命令が解釈されること
（b）パイプライン処理では制御依存とデータ依存が発生すること
（c）データ依存を解決するためにフォワーディング機構が実現すべき機能
を理解することが重要であると考えられる。このため RUE_CHIP1 プロセッサには、以下の機能を搭載することとした。

・パイプラインを構成するレジスタの値を、パイプラインの動作に影響を与えることなく、すべて外部から観測できること
・命令の解釈実行を１クロック単位で実行／停止を制御し、かつ動作クロック周波数を自由に変更できること
・停止中に、汎用レジスタとプログラムカウンタ（PC）の値を書換えて、ある内部状態からプロセッサの動作を再開できること

また「(2) パイプラインプロセッサの各命令の理解」のため、次のような機能を搭載することとした。

・命令レベルのエミュレータのように、１命令の実行を完全に完了した後、次の命令の読出しと実行を行うことで、命令毎に実行結果を観察できる「シーケンシャル実行モード」を搭載すること

この「シーケンシャル実行モード」では、ロード遅延と分岐遅延スロットが存在しない逐次型のプロセッサとして動作し、アセンブリ記述命令の理解を容易にすることを目指している。

2.3 RUE-CHIP1 プロセッサの仕様

RUE-CHIP1 プロセッサは、MIPS 用 gcc (GNU C Compiler) でコンパイルとリンクを行った結果をそのまま実行させることを目指し、MIPS32 命令セットのうち、（浮動小数点コプロセッサと MMU 処理を除く） R3000 の命令をすべて実装した[2]。RUE-CHIP1 がサポートしている MIPS32 命令セットの一覧を表 1 に示す。

図 1 に RUE-CHIP1 のブロックダイアグラムを示す。IF (Instruction Fetch) Stage、ID (Instruction Decode) Stage、EX (Execution) Stage、MEM (Memory) Stage、WB (Write Back) Stage からなる「コンピュータの構成と設計」そのままの 5 段パイプライン構成をとっている。以下では各ステージの動作を説明する。

・IF Stage

プログラムカウンタ（PC）の示す、命令メモリ（Instruction Memory）からの命令の読出しと命令レジスタ（if_IR）への格納を行う

・ID Stage

命令レジスタ（if_IR）のデコードを行い、命令の rs と rt フィールドにより参照される汎用レジスタ（$0, $1, …, $31）の値を id_Rs と id_Rt レジスタに格納する。rs もしくは rt フィールドのレジスタが、直前の 3 命令のどれかにより更新される場合には、フォワーディングした値を id_Rs と id_Rt レジスタに格納する。命令が分岐命令の場合は、分岐条件の判定と分岐先アドレスの計算を行い、プログラムカウンタ（PC）の更新を選択する。

・EX Stage

算術論理演算器（ALU）での演算結果をパイプラインレジスタ ex_C に格納する。乗除算器（MDU）の演算結果は HI と LO レジスタに格納する。ロード／ストア命令の場合は、データメモリ（Data Memory）のアドレスが ex_C に格納される。さらにロード命令の場合は、データメモリに書込む値をパイプラインレジスタ SMD に格納する。

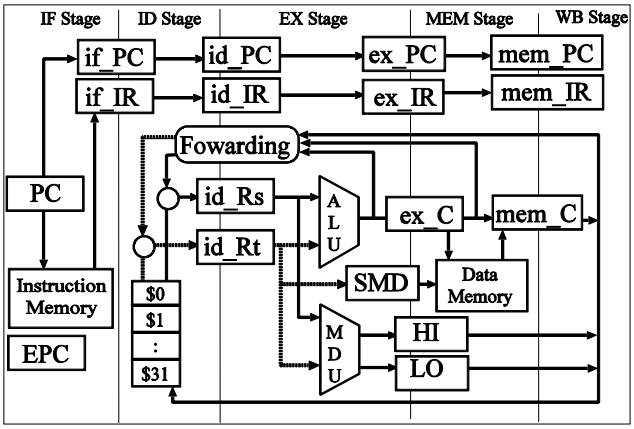

図 1　RUE-CHIP1 プロセッサのブロック図

・MEM Stage

ex_C レジスタの値をメモリアドレスとして、データメモリ（Data Memory）へのアクセスを行う。ロード命令では SMD レジスタの値を書込み、ストア命令では mem_C パイプラインレジスタに読出す。ロード／ストア以外の命令では、ex_C レジスタの値がそのまま mem_C レジスタに転送される。

・WB Stage

mem_C レジスタに格納されている算術論理演算もしくはメモリ読出しの結果を、汎用レジスタ（$0, $1, …, $31）に格納する。

RUE-CHIP1 プロセッサは、将来的にμITRON あるいは Linux OS を動作させることを目指し、完全な例外を実装した。キャッシュメモリと TLB は省略されている。

2.4 RUE-CHIP1 プロセッサの実装

Xilinx 社の FPGA : Spartan 3E-1200 を搭載した Digilent 社の Nexys-2 FPGA Board を用いて RUE-CHIP1 プロセッサを実装した。入出力と主記憶を含む FPGA Board 上の RUE-CHIP1 プロセッサの構成を図 2 に示す。主記憶には、FPGA Board 上の 16MByte SDRAM を利用し、8000 0000 から 8000 7FFF の領域がプログラムメモリに、C000 0000 から C000 7FFF の領域がデータメモリに割当て、プログラムの実行に使用することができる。

入出力はメモリマップド I/O 方式であり、FPGA Board 上に予め用意されている

・8bit LED
・7 セグメント LED
・VGA ディスプレイポート

に割当てたアドレスへの書込みにより出力処理を行うことができる。入力についても同様に FPGA Board 上の

・8bit トグル SW
・4bit プッシュ SW
・PS/2 キーボード

に割当てられたアドレスから読み出すことができる。「4bit プッシュ SW」と「PS/2 キーボード」は、SW あるいはキーが押された場合、RUE-CHIP1 プロセッサに割込み信号が送られ、割込みを用いた入力処理について学習することが可能になっている

RUE-CHIP1 プロセッサのプログラムの実行／中断および内部のレジスタの値の観測や書換えは、RS232C ポートを通じてのコマンド通信で行う。学生実験等の教育現場では、後述する専用の制御／観測ボードから制御と観測を行うが、Teraterm などの PC のターミナルソフトウェアからもコマンドの送受信が可能である。

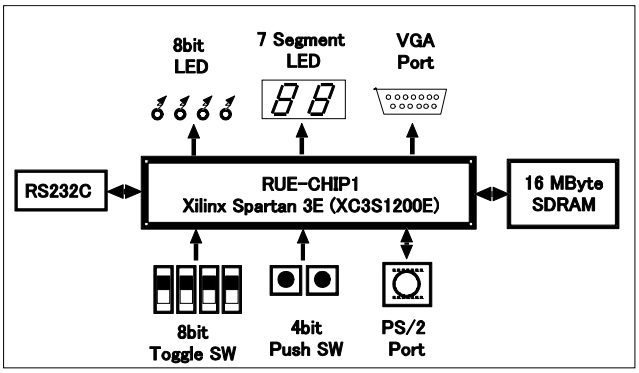

図 2　FPGA Board 上の RUE-CHIP1 プロセッサ

RUE-CHIP1 の動作仕様は Verilog HDL で記述を行い、Xilinx 社の設計ツール：ISE(Integrated Software Environment) 10.1 で論理合成とマッピングを行った。

3. RUE-CHIP1 制御／観測ボード

学生実験などで RUE-CHIP1 プロセッサの動作の制御と内部の観測を行うためのインタフェース用に、RUE-CHIP1 専用の制御／観測ボードを開発した。この制御／観測ボードの概観を図３に示す。RUE-CHIP1 プロセッサを実装した Digilent 社の FPGA Board とは RS232C シリアルケーブルを介して接続する。

「RUE-CHIP1 制御／観測ボード」は 16 文字の英数字を２行表示する蛍光表示デバイスを３個搭載している。それぞれの表示管には

・（命令／データ）メモリのアドレスとその値
・RUE-CHIP1 プロセッサ内部のレジスタの値
・16 進キーボードで入力した 32bit の値

が表示される。値の表示は 16 進数に加え、32bit の値を逆アセンブルして RUE-CHIP1 プロセッサのアセンブリ記述命令で表示することも選択できる。

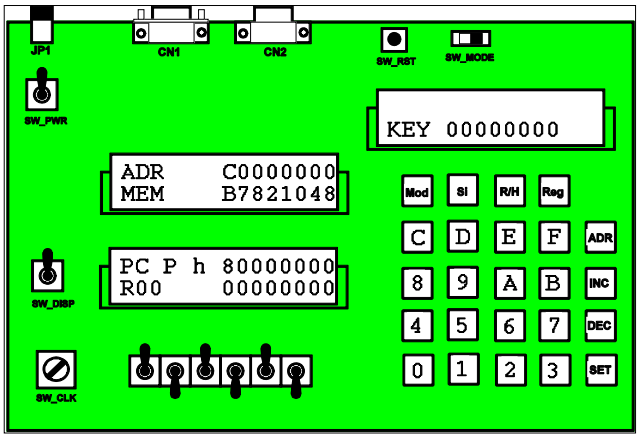

図３　RUE-CHIP1 制御／観測ボードの外観図

搭載されているスイッチ、ボタンの機能のあらましを以下では紹介する。

・16 進キーボードで、メモリアドレスもしくは現在のメモリアドレスに書き込む 32bit の値を指定する。
・RUE-CHIP1 プロセッサ内部のどのレジスタの値を表示するかは、6 bit のトグル SW で選択を行う。
・動作クロック周波数は、4bit のロータリーディップ SW で指定することで、0.1Hz から 100kHz の間で 16 段階に変更する。
・「制御／観測ボード」上の「R/H」ボタンを押すとプログラムの解釈実行が開始され、16 進キーボードで編集したメモリ上の命令／データに基づくプログラムの実行結果を確認することができる。
・プログラム実行中に「R/H」ボタンを押すと、実行は一時中断される。
・「Si」ボタンを押すと、「パイプラインモード」では１クロックだけ命令の解釈・実行が行われ、「シーケンシャル実行モード」では、１命令の解釈実行が完了するまで実行が進められる。
・「パイプラインモード」と「シーケンシャル実行モード」の切換えは、「Mode」ボタンを押すことで交互に切り換わる。
・「RST」ボタンを押すと、RUE-CHIP1 プロセッサの
　・PC は（リセットベクタの）0x8000 0000
　・PC 以外のレジスタはすべて 0x0000 0000
に非同期的にセットもしくはクリアされる。

４．学生実験への適用

立命館大学情報理工学部情報システム学科は、情報システム学の基礎知識として重要なコンピュータアーキテクチャをより深く理解するため、2 回生後期を対象に RUE-CHIP1 プロセッサを用いた学生実験を計画している。この学生実験の目的は以下の(1)～(6)にある。

（1）機械語命令について理解する
（2）機械語命令の実行サイクルについて理解する
（3）パイプラインアーキテクチャについて理解する
（4）アセンブリ言語について理解する
（5）コンピュータ内部のデータ表現について理解する
（6）入出力プログラミングについて理解する

１週あたり 3 時間（２コマ）を割当て、全 7 週で(1)～(6)について学生実験を実施する。

（1）については、16 進キーボードから機械語プログラムを入力して、プログラムの実行結果とプロセッサ内部の PC や汎用レジスタの値の変化を観測することを行う。

（2）については、1 命令だけ命令メモリに書込み、その命令を読み出した後１クロック単位で動作を進めて、パイプラインレジスタの値がどのように変化を観測することで、パイプライン処理の基本的な動作を理解する。

（3）については、パイプラインのスムーズな命令実行を妨げるデータ依存と制御依存を理解することを目標とし、

・データ依存の発生する命令列を実行したときのフォワーディング機構の動作と、ロード遅延の発生
・制御依存が発生する分岐あるいはジャンプ命令での分岐遅延の動作

を観測する。データ依存の発生とは、

ADD $10, $8, $9
ADD $13, $11, $12
ADD $14, $10, $13

のような命令列では、”ADD $14, $10, $13”が参照するレジスタ（$10 と$13）の値が直前の２命令により更新される。パイプライン機構が、図４に示すフォワーディング機構を搭載していない場合、正しい演算結果を得ることができない。その理由は

・”ADD $14, $10, $13”の直前の２命令が WB ステージに到達して $10 と $13 の値を更新する前に、ID ステージで $10 と$13 がパイプラインレジスタ id_Rs と id_Rt に読み出される

というデータ依存が発生するからである。図４のフォワーディング機構がどのように演算結果を正しく id_Rs と id_Rt に提供するかを、１クロック単位で命令の実行を進め、各パイプラインレジスタの値あるいはフォワーディングの結果を観測する。

なお、RUE-CHIP1 プロセッサを用いた学生実験の初年度の 2009 年度は、(1)から(3)までを RUE-CHIP1 プロセッサと制御／観測ボードを使用し、(4)から(6)までは PC 上で

動作する MipsIt シミュレータで実験を行う。2010 年度以降は、(4)から(6)までの実験も RUE-CHIP1 プロセッサと制御／観測ボードを用いて実施することが計画されている。

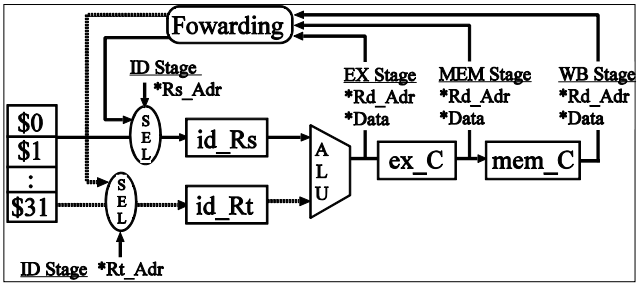

図４　フォワーディング機構

５．おわりに

パイプラインプロセッサの動作の詳細を理解するための教育用プロセッサ：RUE-CHIP1 を開発した。汎用レジスタの他に、PC、パイプラインレジスタ含めたプロセッサ内部の主要なレジスタを観測可能とした

また、パイプラインの動作の理解と、アセンブリ記述プログロラムの動作の理解を両立させるため、通常のパイプライン実行の他に、命令を逐次的に実行するシーケンシャル実行モードを搭載した。立命館大学での学生実験の実施結果をもとに、本教育用パイプラインプロセッサの教育効果について評価を行いたいと考えている。

謝辞

RUE-CHIP1 プロセッサを開発するにあたり、有益なアドバイスを頂いた中谷嵩之氏（当時立命館大学）に感謝いたします。

参考文献

[1] Patterson, D.A. and Hennessy, J.L., コンピュータの構成と設計 第3版, 日経BP社(2006)
[2] Garry Kane, mips RISC アーキテクチャ, 共立出版(1992)

表１　RUE-CHIP1 がサポートする MIPS32 命令セット

ロード命令		ストア命令	
LB	Load Byte	SB	Store Byte
LBU	Load Byte Unsigned		
LH	Load Halfword	SH	Store Halfword
LHU	Load Halfword Unsigned		
LW	Load Word	SW	Store Word
LWL	Load Word Left	SWL	Store Word Left
LWR	Load Word Right	SWR	Store Word Right
算術命令（ALU Immediate）		算術命令（３オペランド）	
ADDI	Add Immediate	ADD	Add
ADDIU	Add Immediate Unsigned	ADDU	Add Unsigned
		SUB	Subtract
SLTI	Set Less Than Immediate .	SUBU	Subtract Unsigned
		SLT	Set Less Than
SLTIU	Set Less Than Immediate Unsigned	SLTU	Set Less Than Immediate
ANDI	AND Immediate	AND	AND
ORI	OR Immediate.	OR	OR
XORI	Exclusive OR Immediate.	XOR	Exclusive OR
LUI	Load Upper Immediate.	NOR	NOR
乗算／除算命令		シフト命令	
MULT	Multiply	SLL	Shift Left Logical
MULTU	Multiply Unsigned	SRL	Shift Right Logical
DIV	Divide	SRA	Shift Right Arithmetic
DIVU	Divide Unsigned	SLLV	Shift Left Logical Variable
MFHI	Move From HI		
MTHI	Move To HI	SRLV	Shift Right Logical Variable
MFLO	Move From LO		
MTLO	Move To LO	SRAV	Shift Right Arithmetic Variable
ジャンプ命令			
J	Jump	JR	Jump Register
JAL	Jump And Link	JALR	Jump And Link Register
分岐命令			
BEQ	Branch on Equal		
BNE	Branch on Not Equal		
BLEZ	Branch on Less than or Equal to Zero		
BGTZ	Branch on Greater Than Zero		
BLTZ	Branch on Less Than Zero		
BGEZ	Branch on Greater than or Equal to Zero		
BLTZAL	Branch on Less Than Zero And Link		
BGEZAL	Branch on Greater than or Equal to Zero And Link		
システム制御コプロセッサ（CP0）命令			
MTC0	Move To CP0	MFC0	Move From CP0
RFE	Restore From Exception		
特殊命令			
STSCALL	STSCALL		
BREAK	BREAK		