

C-001

教育用マイクロプロセッサの設計とFPGAボード上での検証

船附 誠弘[†] 中谷 嵩之[†] 山崎 勝弘[†] 小柳 滋[†]立命館大学大学院理工学研究科[†]

1. はじめに

本研究では、ハードウェア記述言語Verilog-HDLによる教育用マイクロプロセッサKUE-CHIP2の設計とFPGAへの実装を行った。また、FPGAボードとKUE-CHIP2教育用ボード[2]の接続を行い、教育用ボード上での操作をすべてFPGAで実行して、実行結果を教育用ボードに表示できるようにした。これにより、教育用ボードの豊富な入出力装置を用いたFPGAボード上のプロセッサの検証と教育用ボード内蔵のKUE-CHIP2の動作の比較が可能になった。本学の実験[4]では、教育用ボードによるアセンブリプログラミングとデータ入出力までを対象としており、プロセッサの設計は行っていない。そこで、本研究を通じてアセンブリプログラミングだけでなく、実際にKUE-CHIP2を設計することで、プロセッサアーキテクチャの動作、ハードウェア記述言語によるトップダウン設計とCADツールの使用法を総合的に理解するための学習環境を試作した。

2. KUE-CHIP2によるハード/ソフト両輪教育

昨今では、ハードウェア記述言語によるハードウェア設計が普及しており、システム全体を理解し、ハードウェアとソフトウェアの関係について理解している人材が必要とされている。本研究では以上の背景より、KUE-CHIP2という単一のアーキテクチャを対象とした、ハード/ソフト両輪教育について検討を行った。

ソフトウェア学習では、KUE-CHIP2命令セットを用いてアセンブリプログラミングを行う。プログラミングを通じて命令の意味を把握し、アーキテクチャの仕様を理解することを目的とする。

ハードウェア学習では、KUE-CHIP2の仕様を満たすハードウェアのRTLモデリングを行い、FPGA上に実装する。EDKツールによる設計を通じて、トップダウン設計を経験し、プロセッサの内部動作を理解することを目的とする。以上のようなハードとソフトの両方の学習を通じてアーキテクチャ全体の理解が可能であると考えられる。

教育環境の実現において、ソフトウェア学習については、本学の学生実験で用いられているKUE-CHIP2教育用ボードやWebアセンブラ、シミュレータ[3]など、充実していると言える。本研究では、既存のKUE-CHIP2教育用ボードにFPGAボードを接続することでハードウェア学習環境の試作を行った。

3. 教育用マイクロプロセッサ KUE-CHIP2 のアーキテクチャ

3.1 KUE-CHIP2 の概要

KUE-CHIP2(Kyoto University Education Chip2)は、京都大学、京都高度技術研究所(ASTEM)、立命館大学で開発された8ビットのマルチサイクルプロセッサである[1][2]。命令は算術論理演算、ロード・ストア、シフト・ローテート、分岐、入出力、フラグ操作など19種類あり、アドレス指定方式はレジスタ直接、即値、絶対、インデックス修飾の4種類である。命令長は1バイトあるいは2バイトであり、命令の種類、アドレス指定方式によって異なる。

3.2 KUE-CHIP2 のアーキテクチャ

図1に設計したKUE-CHIP2のデータパスを示す。

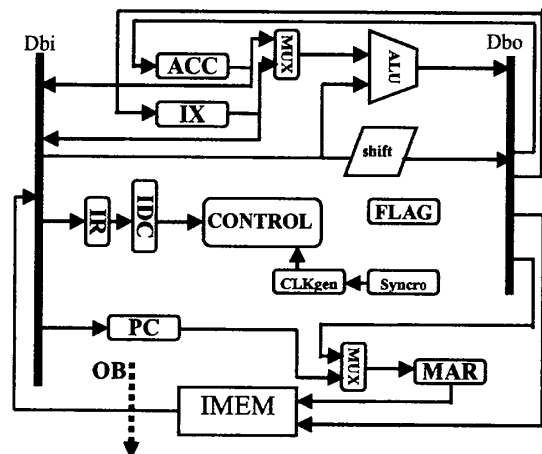


図1 : KUE-CHIP2 のデータパス

プログラムカウンタ(PC)の値がメモリアドレスレジスタ(MAR)に送られ、MARの示すアドレスからメモリ(IMEM)内の命令が読み出される。読み出された命令はデータ入力バス(Dbi)を通り、命令レジスタ(IR)に書き込まれる。IRに格納された命令は命令デコーダ(IDC)によって解釈されて制御回路(CONTROL)からハードウェア各部への制御信号が生成されて命令が実行される。演算用のレジスタとしてアキュムレータ(ACC)、インデックスレジスタ(IX)があり、IXはインデックス修飾にも用いられる。クロックジェネレータ(CLKgen)はマルチサイクル実行におけるフェーズ信号を生成し、シンクロナイザ(Syncro)はスイッチによる非同期入力の同期を行う。各ハードウェアの出力は観測バス(OB)と呼ばれるセレクトに接続しており、任意に選択して内部状態が観測可能である。

3.3 命令実行フェーズ

KUE-CHIP2 は最大5段階の命令実行フェーズを持つマルチサイクルプロセッサであり、各命令のフェーズ動作に応じた制御系を状態遷移化することによって実現される。表1に KUE-CHIP2 の各命令の実行フェーズ動作を示す [1][2]。今回設計した KUE-CHIP2 は、表1の命令実行フェーズに基づいて設計を行った。構成されるレジスタへの書き込みは次の実行フェーズの直前に書き込まれる。

表1: 命令実行フェーズ

	P0	P1	P2	P3	P4
HET			HALT		
NOP			NO OPERATION		
OUT			(ACC)→OBUF	0→OBUF_RE	
IN			(IBUF)→ACC 0→IBUF_FLG_RE	0→IBUF_FLG_CLR	
RCF			0→CF		
SCF			1→CF		
Bcc			(PC)→MAR PC++	STATUS CHECK (MEM)→PC (if satisfied)	
Stm/Ram	PC→MAR	(MEM)→IR	SHIFT A	FLAGSET	
LD			(B)→A		
REG			(MEM)→A		
MM			(MEM)→MAR		
MAR	PC++		(PC)→MAR PC++	(X)→ALU→MAR (MEM)→	(MEM)→A
IX				(MEM)→MAR	
ST				(X)→ALU→MAR (MEM)→	A→(MEM)
MAR					
IX					
REG			(A)→ (B)→ALU→A (CF)→		
MM				(A)→ (B)→ALU→A (CF)→	
MAR			(PC)→MAR PC++	(MEM)→MAR	(A)→ALU→A (CF)→
IX				(X)→ALU→MAR (MEM)→	

4. ハードウェア記述言語による KUE-CHIP2 の論理設計と検証

4.1 ハードウェア記述言語による設計

本研究では、XILINX 社が提供する FPGA 開発ツール FoundationISE を用いて Verilog-HDL での記述、論理合成、配置配線、BIT ファイルの生成を行なった。表2に HDL 行数、実装の規模、設計期間を示す。

表2: 各モジュールの設計規模

	HDL 行数	HW 規模 (SLICE 数)	開発期間 (人月)
CONTROL	1059	99	1.5
クロックジェネレータ	210	21	0.5
ALU	96	42	0.2
SHIFT	49	10	0.1
FLAG	51	5	0.2
Syncro	51	11	0.5
各種レジスタ	91	28	0.1
周辺回路	301	118	0.4
全体	1957	434	3.5

設計対象の FPGA は Spartan II (20 万システムゲート) を用いた。プロセッサの他に周辺回路として、クロック分周回路、7セグメント LED デコーダ、7セグメント LED のダイナミック点灯回路が FPGA 内部に構成されており、それでも FPGA 資源利用率は約 18% と、十分な余裕を持っているため、5 万システムゲート規模の FPGA にも実装可能である。また、最高動作周波数は 15.572MHz であった。実機上

では内部動作を目で確認するために 1Hz から 1MHz の間で動作する。本研究では、動作周波数はクロック分周回路に任意の分周設定を行い、コンフィギュレーション時に静的に動作周波数を変更するものとしている。メモリに関しては、Spartan II FPGA 上の BlockRAM を用いた。

プロセッサの設計期間については表2より、全モジュールの中でも CONTROL、クロックジェネレータの設計に全体の半分以上の時間を要した。CONTROL とクロックジェネレータはステートマシンとして記述しているため、他のモジュールと比べて制御構造が複雑であり、HDL の行数も多くなっている。特に、CONTROL は全ての命令の実行フェーズ毎に制御信号を記述しているため、HDL 総行数の半分強にまで膨らんだ。また、周辺モジュールと CONTROL の一部が外部インターフェースの制御に用いられている。プロセッサは、KUE-CHIP2 設計ドキュメント [1] を参考に設計を行ったが、内部構造を所々変更して設計していることも長期化の原因と考えられる。学生が設計することを想定すると、外部インターフェースの部分あらかじめ用意しておき、設計の対象外にすることや、内部の仕様を規定し、設計の学習体系を確立すれば設計期間の短縮化は十分可能であると考えられる。

4.2 各モジュールの設計

4.2.1 クロックジェネレータによる実行制御

KUE-CHIP2 は教育用のマイクロプロセッサであり、内部動作の観測が容易であるという特徴を持つ。そのため、プロセッサに 1 フェーズ実行 (SP)、1 命令実行 (SI)、全命令実行 (SS) の 3 種類の実行モードに応じた制御機構を HDL でクロックジェネレータ内に記述を行った。図2にプロセッサ実行制御機構の状態遷移を示す。

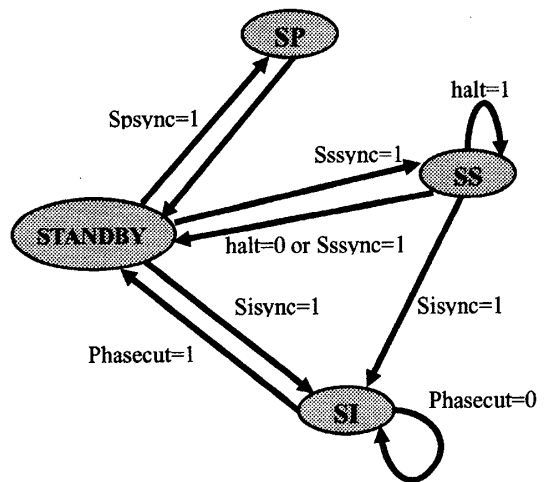


図2: 実行制御機構の状態遷移

STANDBY はプロセッサの停止状態を示し、実行入力とクロックが同期して各実行状態に遷移する。実行状態の間はクロックサイクルの度に命令実行フェーズが次の段階に移行してゆき、制御回路からの制御信号の入力に応じて状態が遷移してゆく。

4.2.2 制御回路 CONTROL

CONTROL は命令デコーダ(IDC)の信号と命令レジスタの内容、クロックジェネレータのフェーズ信号、動作モード信号を元に、プロセッサ内の各モジュールに制御信号が送られる。具体的には、各種レジスタの書き込み、メモリ内容の READ/WRITE 指示、データバスやマルチプレクサの制御、フラグのセット、ALU/SHIFT のオペランド生成が行なわれている。これらの制御はすべて always 文と case 文、if 文でステート化されており、入力信号によってステートを遷移する。ステートを always で記述することで可読性が高いという利点があるが、HDL 行数が非常に大きくなってしまいやすい。そこで、IR 書き込み制御や、SHIFT のオペランド生成などの分岐条件が単純な信号に関しては継続的代入文 assign を用いて行数を抑え、大きくなりすぎないように工夫した。

4.2.3 算術論理演算装置 ALU

図3にALUのモジュール構成を示す。入力には2つのデータ入力と、演算オペランド、キャリーフラグ入力であり、演算結果と演算による各種フラグ結果が出力される。ALUにおける加算器、減算器は両方の演算を切り替えてできる加減算器(ADD8)を設計した。HDL 記述では、全加算器(FULLADDR)のインスタンスを8つ順次桁上がりになるように接続し、演算オペランドが減算、キャリー付減算、または比較減算の時に第2入力を全ビット反転することによって減算を行なうことができる。トップモジュールALUでは、加減算器に論理演算機能を追加している。

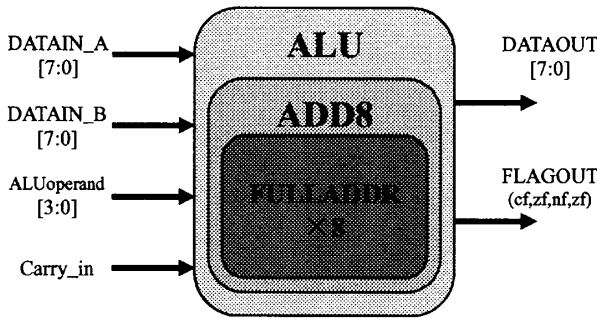


図3: ALUの構成

4.3 シミュレーション

設計したプロセッサをMentor Graphics社のModelSimXEによって論理的な動作検証を行った。サンプルプログラムは本学の実験演習のアセンブリプログラミング課題の一部を用いた[4]。表3に検証用プログラムを示す。以下の検証プログラムからプロセッサの正常動作を確認した。

表3: 検証用プログラムと行数

	プログラム行数
1からNまでの和	7
最大値探索	10
多倍長の加算	8
ユークリッドの互除法	11
バブルソート	23
符号無し1バイト乗算	26
符号無し1バイト除算	39
CRC計算	35

5. FPGA ボードへの実装と検証

5.1 FPGA ボードと教育用ボードの接続

HDL で設計したプロセッサをヒューマンデータ社製FPGA ボード XSP-006 に実装した。本研究では、プロセッサ設計教育を背景としており、豊富な入出力装置によって容易な動作検証、デバッグが行えること、また、設計したプロセッサと内蔵の KUE-CHIP2 の動作比較ができることからFPGA と教育用ボードの接続を行った。図4にその様子を示す。教育用ボードには教育用ボードの外部KUE-CHIP2 への入力信号を出力するコネクタ JP3(15×2)と、教育用ボードの外部 KUE-CHIP2 からの出力信号を入力するコネクタ JP4(20×2)がある。一方、FPGA ボードは教育用ボードの JP3、JP4 の入出力ピンを接続するために50ピンソケット2個(25×2)を使用した。FPGA 端子にはあらかじめ使用されているピンがあり、未使用のFPGA 端子と教育用ボードからの入出力ピンをコネクタ変換用基板上で接続している。また、教育用ボードからFPGA ボード方向へ送られる信号はFPGA の保護のために、変換基板上の抵抗器を介して送信される。教育用ボードはスイッチによって外部のプロセッサに切り替えが可能であり、教育用ボードの入力でFPGA 内のプロセッサが動作し、FPGA の内部動作が教育用ボードのLEDに出力される。

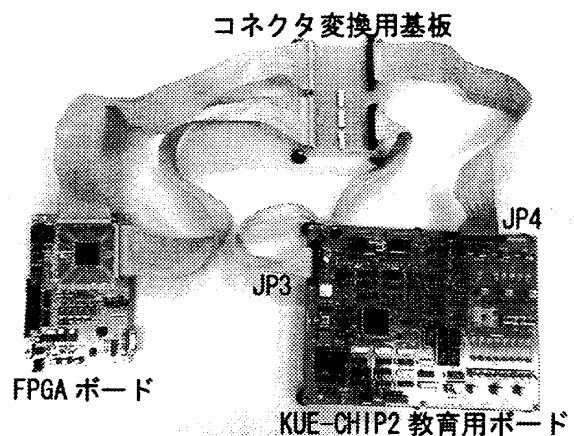


図4: FPGA ボードと KUE-CHIP2 教育用ボードの接続

5.2 実機上での動作検証

FPGA ボードと教育用ボードの接続によるプロセッサの検証方法について述べる。

(1) メモリへのプログラムとデータの書き込み

メモリへの書き込みの方法は、コンフィギュレーション時に静的に書き込む方法と、コンフィギュレーション後のオンボード操作による書き込みの2種類の方法がある。コンフィギュレーション時に書き込む方法は、実行対象のプログラムやデータをあらかじめCOEファイルと呼ばれるテキストファイルに保存しておき、コンフィギュレーション時にメモリに書き込まれる。検証の際、何度もコンフィギュレーションを行なうため、その度メモリの書き込み作業を短縮できる方法である。一方、コンフィギュレーション後の書き込みはボード上のデータトグルスイッチ、SETボタンによって書き換えることができる。対象データやプログラムの部分変更に適した方法である。

(2) 命令の実行とFPGAの動作観測

命令の実行は3.2で述べた各種実行ボタンを押すことにより、任意のタイミングでの内部状態が観測可能である。ボード上の4ビットのOB_SELスイッチによってプロセッサ内部のレジスタ、バス、メモリの出力内容を動的に切り替えて観測ができ、設計したプロセッサの検証、デバッグを容易に行うことができる。

(3) 命令実行後のメモリ内容の観測

プログラム実行後、メモリに正しい計算結果が書き込まれているかどうかを確認する必要がある。ボード上のINC、DECスイッチによってMARの値をインクリメント、デクリメントすることができる。また、OB_SELをMARに設定してデータトグルスイッチでアドレスを指定することによって任意のアドレスにMARを設定でき、OB_SELの値をメモリにすればメモリの内容が7セグメントLEDに表示され、結果の確認ができる。

以上よりFPGAボードとKUE-CHIP2教育用ボードの接続によってプロセッサの設計、実機検証が可能となり、プロセッサアーキテクチャの理解に有効であると考えられる。

5.3 今後のハード/ソフト教育目標

本研究では、FPGAボードと教育用ボードを接続して、プロセッサの動作検証を行うハードウェア設計環境を試作した。今後のハード/ソフト教育環境として以下のことを目標としている。

(1) 操作性、観測性に優れたFPGAコンピュータの実現

FPGAボードコンピュータに様々なアーキテクチャ設計に対応可能な汎用性を持たせることができれば、設計者の独自性を引き出すことも可能である。すなわち、パイプライン、スーパースカラ、VLIWなど、様々なアーキテクチャが実現できれば、設計者自身の創意工夫がFPGAボードコンピュータによって具現化され、より有用性の高いプロセッサの学習環境になると考えられる。

(2) 最適化コンパイラの設計によるソフトウェア学習

ターゲットアーキテクチャに対応する最適化コンパイラ

の設計により、高級言語とアセンブリ言語との関係を理解することができる。パイプライン、VLIWなどの各種アーキテクチャに対応した最適化コンパイラを理解することが必要であると考えられる。

(3) ハード/ソフト総合学習環境の構築

システムの全体を理解するには、ハードウェア、ソフトウェアの各要素を個別に理解するだけでなく、その各要素間の関係について理解することが非常に重要である。このような1つのアーキテクチャを対象としたハード/ソフトの総合的な学習を通じて、1つ1つの要素の繋がり、つまり、コンピュータアーキテクチャ全体の理解が可能であると考えられる。今後、体系化された総合的な学習環境の構築が必要であると考えている。

6. おわりに

本研究では教育用マイクロプロセッサKUE-CHIP2をHDLで設計し、FPGA上に実装して、動作検証を行った。また、FPGAボードをKUE-CHIP2教育用ボードと接続し、実機上での動作検証を行なった。これにより、アセンブリプログラミングのみならず、プロセッサ設計教育にも使用可能であることを示した。

今後の課題としては、FPGAボードコンピュータ、最適化コンパイラ、さらにはオペレーティングシステムなどのコンピュータアーキテクチャ全体の理解するための開発環境の実現が挙げられる。また、実際の教育現場で実現するためのさらなる調査、考察を行いたい。近年では、システムレベル設計に関する研究、設計環境についても脚光を浴びており、今後のハード/ソフト学習環境への位置づけについても考察する必要がある。

謝辞

本研究において数多くのご相談に応じていただき、貴重なご助言を頂きました京都高度技術研究所の神原弘之博士に深く感謝致します。

参考文献

- [1]神原弘之, 越智裕之, 澤田宏, 浜口清治, 岡田和久, 上嶋明, 安浦寛人: KUE-CHIP2設計ドキュメント version1.10, 京都高度技術研究所, 1993
- [2]神原弘之, 越智裕之, 澤田宏, 浜口清治, 岡田和久, 上嶋明, 安浦寛人: KUE-CHIP2教育用ボードリファレンスマニュアル version1.11, 京都高度技術研究所, 1993
- [3]上嶋明: KUE-CHIP2 Web アセンブラ、シミュレータ, 立命館大学理工学部高性能計算研究室, <http://www.hpc.se.ritsume.ac.jp/kue-chip2/>
- [4]山崎勝弘: 情報学実験Ⅲ「教育用ボードコンピュータ」テキスト, 立命館大学理工学部情報学科, 2005