

8, 16, 32 ビットのプロセッサを搭載した教育用計算機：KR-CHIP

KR-CHIP : 8bit Accumulator-based, 16bit Accumulator-based and 32bit 5-stage Pipeline Processors for Computer Education

柴田 敦也[†]
Atsuya Shibata

大迫 裕樹[‡]
Yuuki Oosako

東 香実[‡]
Kagumi Azuma

中野 和香子[‡]
Wakako Nakano

神原 弘之[§]
Hiroyuki Kanbara

1 はじめに

IoT (Internet of Things) 機器は、データセンターで大量の情報処理を行うクラウド側と、実世界で情報収集と外部制御を担当するエッジ側の 2 つに大別される。エッジ側では、センサーが収集した情報の加工や選別に必要とされる計算能力、モータやアクチュエータを制御するのに必要なリアルタイム性、消費電力の制約等に応じて、8bit から 32bit に至る様々な組み込み用のプロセッサが用いられている。

このようなプロセッサの動作原理を理解するため、大学等の教育機関では座学に加え、学生実習等でアセンブリ言語を用いたマイコン・プログラミングが行われている。プログラミング経験の浅い学生がアセンブリ・プログラミングを行うには、8bit クラスの単純な命令セットとアーキテクチャのプロセッサを用いることが望ましい。その一方、実際の IoT 機器はインターネット経由の通信を行うため、Linux OS が動作する 32bit プロセッサも多数用いられている。今後も多様化が進むことが予測されるエッジ側の IoT 機器のプログラミングとその動作理解を補助する教材の必要性が高まっている。

このような背景の下

- KUE-CHIP2 (8bit アキュムレータ方式)
- KUECHIP-3F (16bit アキュムレータ方式)
- RUECHIP (32bit 5 段パイプライン)

の 3 つの計算機入門教育に特化したプロセッサを搭載した教育用計算機：KR-CHIP (Kyoto university & Ritsumeikan university education CHIP) を開発した。

本稿ではこの教育用計算機の開発目的、内蔵する 3 つのプロセッサの仕様とその特徴を紹介する。

2 KUE-CHIP2 と KUECHIP-3F プロセッサ

2.1 概要

KUE-CHIP2 は

- 計算機工学の講義に適した簡素な命令セットとアーキテクチャの提示
- 集積回路工学の設計実習で参考となるプロセッサの実装例の提供

を目指して設計されたアキュムレータ方式の 8bit プロセッサである [1][2][3]。ASIC (Application Specific IC) で実現された KUE-CHIP2 を搭載した教育用ボードは、大学等での学生実験に幅広く用いられている [4]。

この KUE-CHIP2 に対し、

- プログラム領域とデータ領域が 8bit (256 バイト) に限られる
- サブルーチン・コールを実現できない

という指摘を教育現場から受け、KUE-CHIP2 をベースに 16bit 化しスタックポインタを追加した KUECHIP-3F プロセッサの設計が行われた [5]。

2.2 ブロック図

KUE-CHIP2 と KUECHIP-3F のブロックダイアグラムを図 1 に示す。入力データバス (DBi)、出力データバス (DBo)、アドレスバス (AB) の他に、内部状態を観測するための観測バス (OB) がある。OB バスを通じて、内部のレジスタやメモリの値を外部から観察できる。また、DBi バス経由で内部のレジスタやメモリの内容を直接書き換えることができる。

演算器は ALU が 1 個あり、演算用レジスタには、アキュムレータ (ACC)、インデックスレジスタ (IX) がある。KUECHIP-3F では、これらの他にスタックポインタ (SP) が追加されている。制御系は、プログラムカウンタ (PC)、命令レジスタ (IR)、命令デコーダ (IDC) などがある。また、メモリアクセスの際のアドレスを保持するメモ

[†] 奈良先端科学技術大学院大学, NAIST

[‡] 関西学院大学, Kwansai Gakuin University

[§] 京都高度技術研究所, ASTEM RI

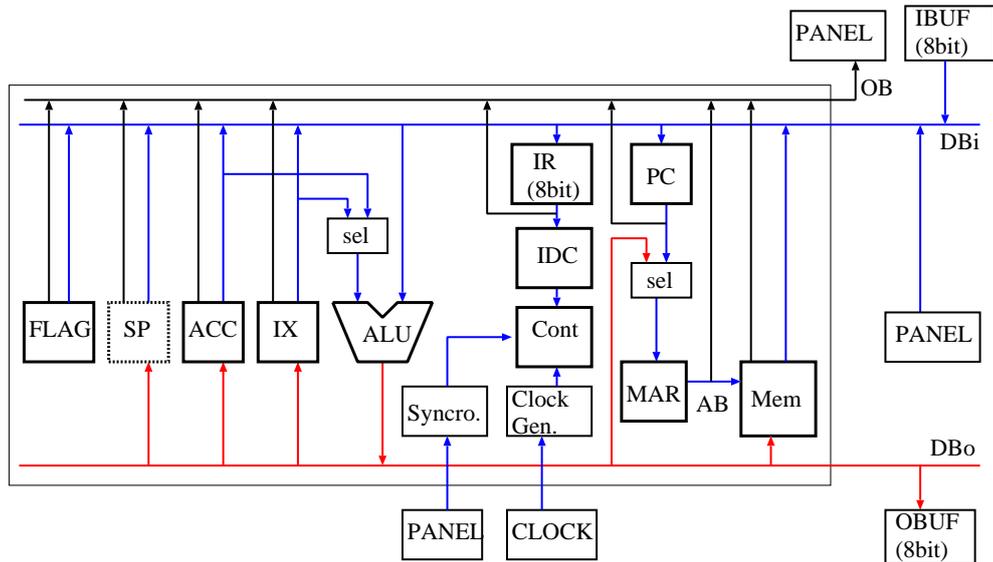


図1 KUE-CHIP2 と KUECHIP-3F のブロックダイアグラム (SP は KUECHIP-3F のみ搭載)

リアドレスレジスタ (MAR) がある。フラグは 4 種類あり、演算の結果により零フラグ (ZF)、ネガティブフラグ (NF)、桁あふれフラグ (VF)、桁上げフラグ (CF) がそれぞれ設定される。

各バス、演算器、レジスタのデータ幅は KUE-CHIP2 は 8bit, KUECHIP-3F では 16bit である。アクセス可能なメモリ領域は、KUE-CHIP2 では命令とデータにそれぞれ 256 バイト, KUECHIP-3F では 65,536 バイトである。

セントロニクス規格互換の入力ポート (IBUF) を読み出す命令と出力ポート (OBUF) に書き込む命令により、外部との通信を行うことができる。

2.3 命令セット

KUE-CHIP2 と KUECHIP-3F の命令セットを表 1 に示す。ロード/ストア、算術論理演算、シフト演算、分岐、実行制御の命令：40 種は、KUE-CHIP2 と KUECHIP-3F で共通する。KUECHIP-3F にはスタックポインタ操作に関連する 6 種の命令が追加され、サブルーチンコールと復帰のプログラミングが可能になっている。

KUE-CHIP2 の命令実行フェーズを表 2 に、KUECHIP-3F の命令実行フェーズを表 3 に示す。両プロセッサは、最大 5 フェーズで 1 命令を実行する。P0, P1 は命令取り出しフェーズであり、P2, P3, P4 が実行フェーズである。

2.4 FPGA 向け設計

KUE-CHIP2 と KUECHIP-3F は、VHDL 言語を用いて設計が行われた。Xilinx 社の Artix-7(XC7A35T) FPGA をターゲットに、Vivado 2015.4 を用いて論理合成とマッピングを行った。動作確認には、Digilent 社の Nexys4-DDR ボードと Cmod A7 モジュールを用いた [5][6]。

3 RUECHIP プロセッサ

3.1 概要

学生実習等を通じて、詳細なパイプライン・プロセッサの動作を理解する教材として RUECHIP プロセッサが開発された。計算機アーキテクチャの初学者が

- パイプラインの各ステージの一連のレジスタ転送の組合せにより、命令が正しく解釈実行されること
- 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
- 制御依存とデータ依存がフォワーディング機構により解決されること

について理解を深める計算機を提供することが目的である。このため、プロセッサ内部で行われる命令解釈と実行を観測する機能を搭載した。教育用途に特化した 32bit プロセッサであることが最大の特徴である [7]。

3.2 ブロック図

RUECHIP のブロックダイアグラムを図 2 に示す。パターソン&ヘネシーによる「コンピュータの構成と設計」の、ハーバード・アーキテクチャの 32bit パイプライン・プロセッサに基づいている。パイプラインは IF (Instruction Fetch), ID (Instruction Decode), EX (EXecution), MEM (MEMory), WB (Write Back) の 5 段で構成されている [8]。命令の解釈実行に必要なバスに加えて、前述した KUE-CHIP2 / KUECHIP-3F プロセッサと同様に、内部レジスタの値を直接観測する観測バス (OB) を備えている。

以下では各ステージのレジスタ転送レベルの動作を説明する。

3.2.1 IF ステージ

命令メモリ (Instruction Memory) から、プログラム・カウンタ (PC) の示すアドレスの命令を読み出し、命令レ

表 2 KUE-CHIP2 の各命令の実行フェーズ

Phase Instruction		P0	P1	P2	P3	P4		
HLT		(PC) → MAR PC++	(Mem) → IR	HLT				
NOP						No Operation		
OUT						(ACC) → OBUF	0 → OBUF_WE	
IN						(IBUF) → ACC 0 → IBUF_RE	0 → IBUF_FLG_CLR	
RCF						0 → CF		
SCF						1 → CF		
Bcc						(PC) → MAR PC++	STATUS CHECK (Mem) → PC (if condition)	
Ssm Rsm						TCF SET SHIFT A	NF, ZF, VF, CF Set	
LD	ACC					(A) → B		
	IX						(Mem) → A	
	d [d] (d)					(PC) → MAR PC++	(Mem) → MAR (IX) → (Mem) → ALU → MAR	(MEM) → A
ST	[d] (d)					(PC) → MAR PC++	(Mem) → MAR (IX) → (Mem) → ALU → MAR	(A) → (Mem)
	[IX+d] (IX+d)							
SBC ADC SUB ADD EOR OR AND CMP	ACC					(A) → (B) → ALU → A [[CF]] → NF, ZF, VF [, CF] SET		
	IX						(A) → (B) → ALU → A [[CF]] → NF, ZF, VF [, CF] SET	
	d			(PC) → MAR PC++	(Mem) → MAR (IX) → (Mem) → ALU → MAR	(A) → (B) → ALU → A [[CF]] → NF, ZF, VF [, CF] SET		
	[d] (d)							
	[IX+d] (IX+d)							

ジスタ (if_IR) に格納する。

3.2.2 ID ステージ

命令レジスタ (if_IR) に格納された命令のデコードを行う。命令の Rs と Rt フィールドに従い、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の値が、直前の 3 命令のいずれかにより更新される場合、Forwarding モジュールによりフォワーディングされた値が id_Rs と id_Rt レジスタに格納される。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。

3.2.3 EX ステージ

id_IR レジスタの命令が算術論理演算もしくはシフト演算の場合、算術論理演算器 (ALU) の演算結果をパイプラインレジスタ ex_C に格納する。乗除算命令では、乗除算器 (MDU) による演算結果が HI と LO レジスタに格納される。id_IR レジスタの命令がロード/ストア命令の場合、データメモリ (Data Memory) のアドレスが ex_C に格納される。ロード命令の場合は、データメモリに書き込む値をパイプラインレジスタ SMD に格納する。

3.2.4 MEM ステージ

ex_IR レジスタの命令がロード/ストア命令の場合、ex_C レジスタの値を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

ジスタの値を書き込み、ストア命令では mem_C レジスタにデータを読み出す。ロード/ストア命令以外の命令では、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。

3.2.5 WB ステージ

mem_C レジスタに格納されている算術論理演算の結果もしくはデータメモリから読み出したデータを、汎用レジスタ (\$0, \$1, ... \$31) に格納する。

3.3 命令セット

RUECHIP の命令セットを表 4 に示す。MIPS32 命令セットのうち、浮動小数点コプロセッサと MMU 処理を除いた R3000 の命令をすべて実装した。MIPS 用 gcc (GNU C Compiler) でコンパイル&リンクしたバイナリがそのまま実行できる。浮動小数点型のデータを含む C 言語プログラムは、soft-float (浮動小数点エミュレーション) ライブラリを用いて、浮動小数点演算の整数演算へ置き換えることで、実行できる [9]。

3.4 FPGA 向け設計

RUECHIP は、Verilog-HDL を用いて設計が行われた。Xilinx 社の Artix-7(XC7A35T) FPGA をターゲットに、Vivado 2015.4 を用いて論理合成とマッピングを行った。動作確認には、Digilent 社の Nexys4-DDR ボードと Cmod A7 モジュールを用いた。

表 3 KUECHIP-3F の各命令の実行フェーズ

Phase	P0	P1	P2	P3	P4
Instruction					
HLT			HLT		
NOP			No Operation		
OUT			(ACC) → OBUF	0 → OBUF_WE	
IN			(IBUF) → ACC 0 → IBUF_RE	0 → IBUF_FLG_CLR	
RCF			0 → CF		
SCF			1 → CF		
LD	ACC IX d [d] [IX+d] [SP+d]		(A) → B	(Mem) → A (Mem) → MAR (IX(SP)) → (Mem) → ALU → MAR	(MEM) → A
ST	[d] [IX+d] [SP+d]		(PC) → MAR (PC+2) → PC	(Mem) → MAR (IX(SP)) → (Mem) → ALU → MAR	(A) → (Mem)
Bcc	(PC) → MAR	(Mem) → IR	(PC) → MAR (PC+2) → PC	STATUS CHECK (Mem) → PC (if condition)	
Ssm Rsm	(PC+2) → PC		TCF SET SHIFT A	NF, ZF, VF, CF Set	
SBC ADC SUB ADD EOR OR AND CMP	ACC IX d [d] [IX+d] [SP+d]		(A) → (B) → ALU → A [(CF)] → NF, ZF, VF [, CF] SET	(A) → (B) → ALU → A [(CF)] → NF, ZF, VF [, CF] SET (Mem) → MAR (IX(SP)) → (Mem) → ALU → MAR	(A) → (B) → ALU → A [(CF)] → NF, ZF, VF [, CF] SET
LD	IX, SP		SP → IX		
LD	SP, d		(PC) → MAR (PC+2) → PC	(Mem) → SP	
INC			+2 → (SP) → ALU → SP		
DEC			-2 → (SP) → ALU → SP		
ADD	SP, d		(PC) → MAR (PC+2) → PC	(SP) → (Mem) → ALU → SP	
SUB	SP, d		(SP-2) → MAR (SP-2) → SP	(ACC(IX)) → Mem	
PSH	ACC (IX)		(SP) → MAR (SP+2) → SP	Mem → ACC(IX)	
POP	ACC (IX)		(SP-2) → MAR (SP-2) → SP	(PC+2) → Mem (PC) → MAR	(Mem) → PC
CAL			(SP) → MAR (SP+2) → SP	(Mem) → PC	
RET					

4 KR-CHIP とその教育用ボード

4.1 アーキテクチャ

KR-CHIP のアーキテクチャを図 3 に示す。「KUE-CHIP2」「KUECHIP-3F」「RUECHIP」の3つのプロセッサが Xilinx 社の Artix-7(XC7A35T) FPGA に搭載されている。

KUE-CHIP2 と KUECHIP-3F の主記憶には、FPGA の内蔵ブロックメモリ (Block RAM) を用いている。

RUECHIP は、Cmod A7 モジュール上の外付け SRAM (On Board SRAM (512kB)) を主記憶に用いる。外付け SRAM からの命令フェッチとデータのロード/ストアは、AXI Bus 経由で Memory Controller を介して行われる。

AXI Bus のスレーブ側には Memory Controller に加

えて、AXI SPI IP と Block RAM(2port) が接続されている。SPI (Serial Peripheral Interface) 規格に対応した Digilent 社の Pmod モジュール (ex. Pmod CAN) を接続すると、RUECHIP プロセッサは外部との通信/制御を高速に行うことができる。SPI 以外の通信インタフェースをもつ各種 Pmod モジュール (ex. Pmod HB5) については、Block RAM(2port) と Pmod モジュール間を接続する User-Circuit を実装することで対応できる。

各プロセッサの

- 主記憶のプログラム/データの編集
- プログラム実行の開始/停止
- 内部レジスタの値の観測

は、KR-CHIP 外部のマイコン (Microcontroller) もし

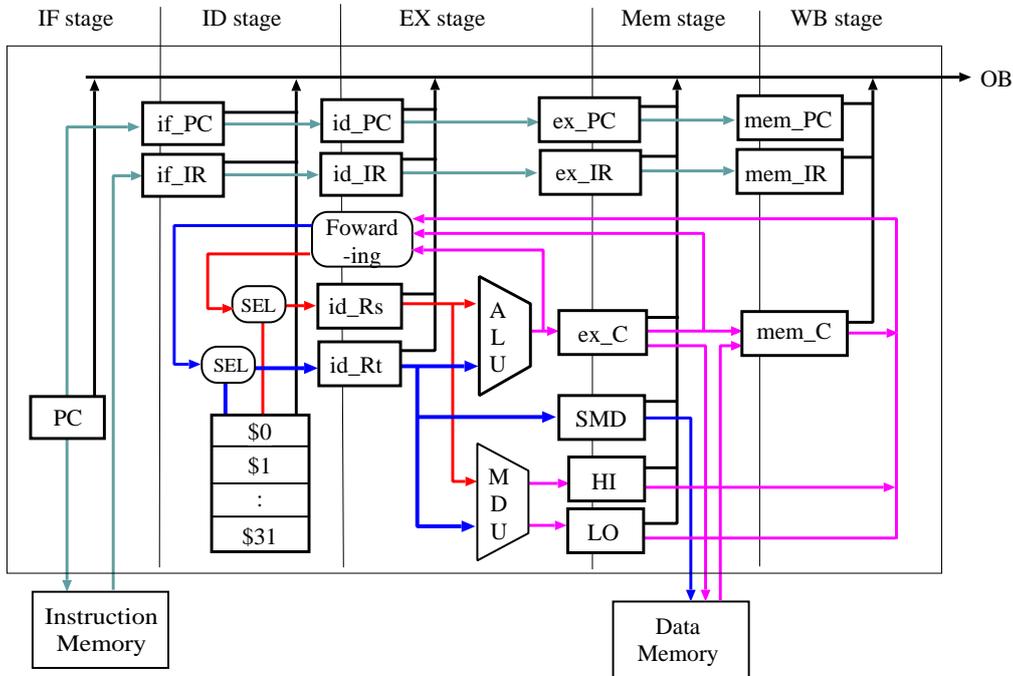


図2 RUECHIPのブロックダイアグラム

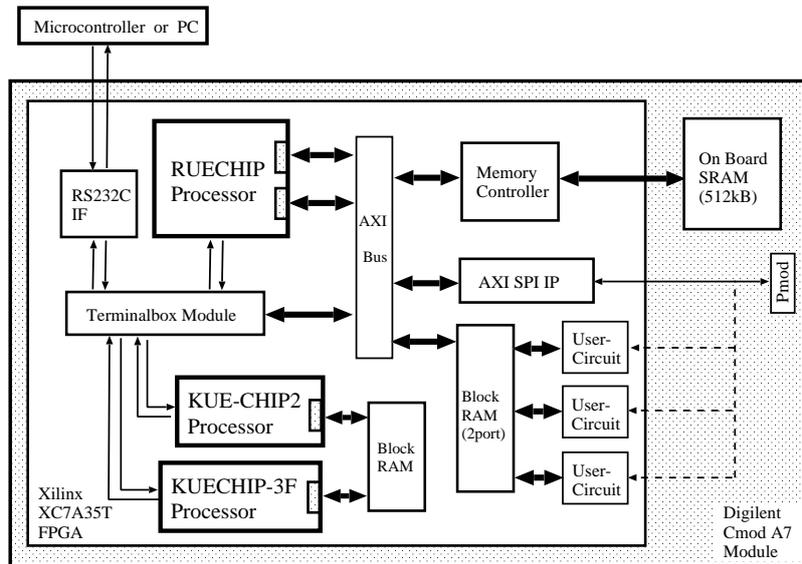


図3 KR-CHIPのアーキテクチャ

くは PC から RS232C IF を介して制御コマンドを送信することで行う。送られてきたコマンドは、図3の Terminalbox Module が解釈を行い、主記憶の編集や各プロセッサの実行/停止を制御する信号を生成する。メモリ編集/プログラム実行制御/内部レジスタの観測を行った結果は、Terminalbox Module が FPGA 外部に実行結果を返信することで行われる。

各プロセッサの実行/停止を Terminalbox Module が制御することで、

- 1クロックフェーズ、1命令単位での命令実行と停止
- 極めて低速 (0.1 Hz) なプログラムの実行

を行うことが可能になっている。これにより、プログラム

を実行しながらプロセッサの観測バス (OB) を通じて、内部レジスタやメモリの値の変化を実際に観測することが可能になっている。

4.2 FPGA による実現

Xilinx 社の Artix-7 (XC7A35T) をターゲットに合成を行い、Cmod A7 モジュールを用いて動作確認を行っている。論理合成には Vivado 2015.4 を使用した。合成結果を表5に示す。LUT は LUT 数, FF はフリップフロップ数, delay は回路の遅延である。() 内の数字は LUT, FF の利用率を示す。

4.3 KR-CHIP 教育用ボード

KR-CHIP をプログラミングし、命令の解釈実行を観測することができる、教育用ボードの概観を図4に示す。

表1 KUE-CHIP2 と KUECHIP-3F の命令セット

ロード/ストア	
LD	LoaD
ST	STore
算術論理演算	
ADD	ADD
ADC	ADD with Carry
SUB	SUBtract
SBC	SuBtract with Carry
CMP	CoMPare
AND	AND
OR	OR
EOR	Exclusive OR
シフト演算	
SRA	Shift Right Arithmetically
SLA	Shift Left Arithmetically
SRL	Shift Right Logically
SLL	Shift Left Logically
RRA	Rotate Right Arithmetically
RLA	Rotate Left Arithmetically
RRL	Rotate Right Logically
RLL	Rotate Left Logically
分岐	
BA	Branch Always
BVF	Branch on oVerFlow
BNZ	Branch on Not Zero
BZ	Branch on Zero
BZP	Branch on Zero or Positive
BN	Branch on Negative
BP	Branch on Positive
BZN	Branch on Zero or Negative
BNI	Branch on No Input
BNO	Branch on No Output
BNC	Branch on No Carry
BC	Branch on Carry
BGE	Branch on Greater than or Equal
BLT	Branch on Less Than
BGT	Branch on Greater Than
BLE	Branch on Less than or Equal
実行制御	
HLT	HaLT
NOP	No OPeration
IN	INput
OUT	OUTput
SCF	Set Carry Flag
RCF	Reset Carry Flag
SP 操作関連 (KUECHIP-3F のみサポート)	
INC	INCrement
DEC	DECrement
PSH	PuSH
POP	POP
CAL	CALl
RET	RETurn

表4 RUECHIP の命令セット

ロード/ストア	
LB	Load Byte
LBU	Load Byte Unsigned
LH	Load Halfword
LBU	Load Halfword Unsigned
LW	Load Word
LWL	Load Word Left
LWR	Load Word Right
SB	Store Byte
SH	Store Half Word
SW	Store Word
SWL	Store Word Left
SWR	Store Word Right
算術論理演算	
ADD	ADD
ADDU	ADD Unsigned
SUB	SUBtract
SUBU	SUBtract Unsigned
SLT	Set Less Than
SLTU	Set Less Than Unsigned
AND	AND
OR	OR
XOR	eXclusive OR
NOR	NOR
ADDI	ADD Immediate
ADDIU	ADD Immediate Unsigned
SLTI	Set Less Than Immediate
SLTIU	Set Less Than Immediate Unsigned
ANDI	AND Immediate
ORI	OR Immediate
XORI	eXclusive OR Immediate
LUI	Load Upper Immediate
シフト演算	
SLL	Shift Left Logical
SRL	Shift Right Logical
SRA	Shift Right Arithmetic
SLLV	Shift Left Logical Variable
SRLV	Shift Right Logical Variable
SRA	Shift Right Arithmetic Variable
乗算/除算	
MULT	MULTiply
MULTU	MULTiply Unsigned
DIV	DIVide
DIVU	DIVide Unsigned
MFHI	Move From HI
MTHI	Move To HI
MFLO	Move From LO
MTLO	Move To LO
分岐	
BEQ	Branch on EQual
BNE	Branch on Not EQual
BLEZ	Branch on Less than or Equal to Zero
BLTZ	Branch on Less Than Zero
BGEZ	Branch on Greater than or Equal to Zero
BLTZAL	Branch on Less Than or Equal to Zero
BGEZAL	Branch on Greater than or Equal to Zero And Link
ジャンプ	
J	Jump
JR	Jump Register
JAL	Jump And Link
JALR	Jump And Link Register
システム制御 (コプロセッサ (CP0) 関連)	
MTC0	Move To Cp0
MFC0	Move From Cp0
RFE	Retern From Exception
特殊命令	
STSCALL	System CALL
BREAK	BREAK

表 5 KR-CHIP の合成結果

LUT	FF	delay [ns]
14,861 (74.95%)	9,772 (23.49%)	9.167

KR-CHIP は、本ボード上の Digilent Cmod A7 モジュール上の Artix-7 FPGA に実装されている。KR-CHIP 内部の「KUECHIP-2」「KUECHIP-3F」「RUECHIP」のどのプロセッサでプログラムを実行するかは、ボード上のトグルスイッチもしくはスライドスイッチにより選択する。

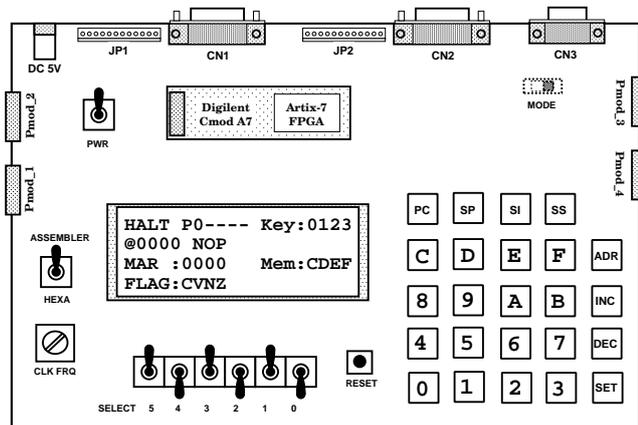


図 4 KR-CHIP 教育用ボード

教育用ボードは 20 文字の英数字を 4 行表示する蛍光表示管デバイスを搭載している。この表示管には

- プロセッサの実行/停止状態
- 16 進キーボードで入力した 8/16/32 bit の値
- KR-CHIP 内部のプロセッサのレジスタの値
- メモリのアドレスとその値 (命令/データ)

が表示される。値の表示は 16 進数に加え、各プロセッサのアセンブリ記述に逆アセンブルして表示することも選択できる。

KUE-CHIP2 と KUECHIP-3F では、プログラム実行/停止を以下の 3 つのモードから選択できる。

- SP : 命令を 1 クロックフェーズ実行して停止する
- SI : 1 命令を実行して停止する
- SS : プログラムの実行を開始する。実行中に SS ボタンが押されると停止する

RUECHIP では、以下の 2 つの実行モードをサポートしている。

- SP : 命令を 1 クロックフェーズ実行して停止する
- SS : プログラムを連続で実行し、実行中に SS ボタンが押されると停止する

さらに外部機器との入出力を行うため、以下のような通信ポート/拡張コネクタを備えている。

- 2 台の教育用ボード間で通信実験を行うためのセン

トロニクス規格準拠の 8bit パラレル入出力ポート (CN1 と CN2)

- PC 上でアSEMBルした機械語命令列を転送するための RS232C シリアル通信ポート (CN3)
- Digilent 社の Pmod モジュールを接続可能な Pmod コネクタ : 4 ヶ (Pmod_1 ~ Pmod_4)
- 差分電力解析等のサイドチャネル攻撃への耐タンパ性評価のため、ソフトウェア実行中のプロセッサの消費電力の変動をデジタル・オシロスコープで計測するのに用いるコネクタ (SMA_1 ~ SMA_3) [10]

5 おわりに

本稿では、教育目的に特化した「8bit アキュムレータ方式」「16bit アキュムレータ方式」「32bit 5 段パイプライン方式」の 3 種類のプロセッサを搭載した計算機教育用プロセッサ : KR-CHIP を紹介した。現在、教育用ボードを用いて各プロセッサの動作確認を進めている。実際の教育現場に適用して、その有効性を確認することが将来課題である。

謝辞

KUECHIP-3F を用いた学生実験を担当された立命館大学情報理工学部の國枝義敏教授と TA の皆さんに感謝します。KUECHIP-3F の命令セットとアーキテクチャの改良に際し、ご助言を頂きました立命館大学情報理工学部の越智裕之教授に感謝します。本稿の開発を行なうにあたり、有益なアドバイスを頂いた元立命館大学の中谷嵩之氏、元京都大学の矢野正治氏、元関西学院大学の田村真平氏に感謝します。

参考文献

- [1] 越智裕之, 澤田宏, 岡田和久, 上嶋明, 神原弘之, 濱口清治, 安浦寛人: “計算機工学・集積回路工学教育用 マイクロプロセッサ KUE-CHIP2,” 情報処理学会研究報告, 1992-ARC-96-13 (1992).
- [2] 神原弘之, 越智裕之, 澤田宏, 濱口清治, 岡田和久, 上嶋明, 安浦寛人: “KUE-CHIP2 設計ドキュメント,” 京都高度技術研究所 (1992).
- [3] 神原弘之, 越智裕之, 澤田宏, 濱口清治, 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,” 京都高度技術研究所 (1992).
- [4] 川端英之, 弘中哲夫, 寺内衛: “情報工学基礎実験,” 大学教育出版 (Arp. 2005).
- [5] 東香実, 大迫裕樹, 柴田敦也, 神原弘之, 國枝義敏: “教育用 16bit プロセッサ KUECHIP-3F の開発,” 2017 年度 情報処理学会関西支部 支部大会, A-04 (2017).
- [6] 神原弘之, 安浦寛人, Pankaj Kukkal, Hideaki KOBAYASHI, 野地保, 小栗清: “ハードウェア記述言語の比較,” 情報処理学会誌, Vol. 33 No. 11 (1992).
- [7] 神原弘之, 金城良太, 戸田勇希, 矢野正治, 小柳滋: “パイプラインプロセッサを理解するための教材 RUECHIP1,” 2009 年度 情報処理学会関西支部 支部大会, A-09 (2009).
- [8] デビッド A パターソン, ジョン L ヘネシー: “コンピュータの構成と設計 第 5 版,” 日経 BP 社 (2014).
- [9] 竹林陽, 伊藤直也, 田村真平, 神原弘之, 石浦菜岐佐: “高位合成系 ACAP を用いたモータの浮動小数点モデルの FPGA 上での実行,” 2014 年度 情報処理学会関西支部 支部大会, A-02 (2014).

- [10] H. Kanbara, N. Ito, H. Takebayashi, M. Takenae, T. Tsukamoto: "FPGA Prototyping of a Smart Card Platform for Evaluating Tamper Resistance of Cryptographic Circuits," Proc. Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI 2016), R1-14 (2016).