

## BISC (Bus Instruction Set Computer) 型プロセッサ BISC-2

水谷 竜也<sup>†</sup>, 吉田 亘<sup>†</sup>, 新居 英明<sup>††</sup>, 山下 幸彦<sup>†</sup>

<sup>†</sup> 東京工業大学 大学院理工学研究科

<sup>††</sup> 東京工業大学 工学部

本論文では、BISC アーキテクチャを採用した新しいプロセッサ BISC-2 を提案する。BISC アーキテクチャでは、命令をレジスタ間のデータ転送のみに限定しているため、機能の追加変更に対して命令セットを変更する必要がなく、またプロセッサ内の各ユニットの独立性が高い構成となっていることから、容易に様々な機能を追加変更することができる。BISC-2 では、従来の BISC 型プロセッサの問題点を改善するために、メモリアクセスを高機能化し割り込み機能を追加している。また、命令の効率的な並列実行とアウトオブオーダー実行を実装することによって処理の高性能化をはかっている。最後に、シミュレーションによって BISC-2 の性能評価を行う。

## BISC (Bus Instruction Set Computer) Processor BISC-2

Tatsuya Mizutani<sup>†</sup>, Wataru Yoshida<sup>†</sup>, Hideaki Arai<sup>††</sup>, and Yukihiro Yamashita<sup>†</sup>

<sup>†</sup> Graduate school of Science and Engineering, Tokyo Institute of Technology

<sup>††</sup> Faculty of Engineering, Tokyo Institute of Technology

In this paper, we propose a novel BISC-based processor called BISC-2. In the BISC architecture, since its instructions are limited only to data transfers between registers, it is not necessary to change its instruction set when functions of a processor are modified and/or added. Moreover, the BISC processors have highly independent units. Therefore, the BISC architecture has an efficient property, that is, it is easy to add and modify functions of a processor. To solve some problems of a conventional BISC processor, in the proposed BISC-2, memory access is made more efficient and interrupt execution is involved. Furthermore, the ability of data transfer is improved by using efficient parallel execution and out-of-order execution. Finally, we evaluate the performance of the BISC-2 by simulation.

### 1 まえがき

現在のプロセッサでは、処理の高速な実行を実現するために、その使用目的に応じて機能を追加したり変更することが必要となっている。例えば、画像や音声、三次元グラフィックスといったマルチメディアデータ処理を高速に実行するために、プロセッサにそれぞれの目的に応じた機能の追加が行われている。しかし既存のプロセッサでは、パイプライン、スーパースカラ、分岐予測、命令のアウトオブオーダーなどの機構が取り入れられており、

容易に機能を追加変更することができなくなっている。この問題点を解決するために、BISC (Bus Instruction Set Computer) とよばれるマイクロプロセッサの新しいアーキテクチャが提案されている [1]。このアーキテクチャでは、命令がレジスタ間のデータ転送のみに限定されており、機能の追加や変更に対して命令セットを変更する必要がない。算術論理演算やメモリアクセスなどの各機能は、命令によって制御されるのではなく、命令によって転送されるデータで間接的に制御される。こ

のため、各機能ユニットの独立性が高い構成となっている。機能を追加するときは、機能ユニットをレジスタファイルのレジスタに接続するだけでよく、容易に様々な機能を追加したり変更することができる。また、機能ユニット独自にパイプラインなどの高速化手法を導入することも容易である。

BISCアーキテクチャを採用した初のプロセッサとして BISC-1 が開発されている [2]。BISC-1 は 2 重化された内部バスをもち、2 つの命令を並列に実行できる。また、算術論理演算、メモリアクセス、分岐制御をおこなう 3 つの機能ユニットをもつ。シミュレーションによる実験の結果、BISC-1 のいくつかの問題点が明らかになっている。この問題点を解決するために、本論文では新しい BISC 型プロセッサ BISC-2 を提案し、その仕様を明らかにする。BISC-1 からの主な変更点は、メモリアクセスの高機能化、割込み機能とアンパックユニットの追加、命令の効率的な並列実行とアウトオーダー実行機能の追加である。

## 2 BISC-2 の基本構成

### 2.1 構成

BISC-2 の基本構成を図 1 に示す。また、各部の説明を次に示す。

**バス (Bus)** バスを経由してレジスタ間データ転送を行うことにより、命令を実行する。バスは、レジスタファイルの全てのレジスタと接続している。多重化されており、複数のデータ転送を並列に行うことができる。

**レジスタファイル (Register File)** 64 個の 32 ビットレジスタから構成されており、それぞれのレジスタはすべてのバスに接続されている。レジスタは、機能ユニットに接続されたある特定の機能をもった機能レジスタと、単にデータを蓄えるだけの汎用レジスタが存在する。図 1 のレジスタファイルには機能レジスタのみが記されているが、実際は汎用レジスタも存在する。レジスタの一覧を表 1 に示す。NLR と IDR は機能ユニットに接続されていないが、特定の機能をもつレジスタなので、機能レジスタに含めることにする。

**バス命令バッファ (BIB)** 複数の 12 ビットバッファから構成される。各バッファは、それぞれ転

送命令を格納する。

**命令発行ユニット (IU)** BIB にある命令を発行する。ただし、命令に依存関係がある場合や、対象レジスタが命令発行できる状態でない場合は、発行しない。

**イミディエイトデータバッファ (IDB)** 複数の 32 ビットバッファから構成される。各バッファは、それぞれイミディエイトデータを格納する。

**命令フェッチユニット (IFU)** 命令用キャッシュメモリ (I-cache) から、プログラムカウンタが示すアドレスの命令を読み出し、命令を BIB へ、イミディエイトデータを IDB へ送る。ただし分岐点を越えた先行読みだし時には、Prefetch BIB および Prefetch IDB へ保存する。

**機能ユニット** 機能ユニットには、算術論理演算ユニット、ロードストアユニット、分岐制御ユニット、アンパックユニット、システム制御ユニットがある。それぞれのユニットは機能レジスタと接続している。

### 2.2 命令形式

BISC-2 の命令は、本質的にレジスタ間のデータ転送のみに限定されている。CISC, RISC などの一般的な CPU では、命令語のオペレーション・フィールドで命令の種類を指定する。しかし、BISC-2 は命令が一種類に限られているので、オペレーション・フィールドを必要としない。命令語で指定する必要があるものは、データを送る送信レジスタと、データを受ける受信レジスタの 2 つだけである。BISC-2 の命令形式を図 2 に示す。なお、この命令形式は BISC-1 のものと全く同じである。

bit29 から bit0 を使用して、2 ワードで 5 つの命令を指定できる。bit29 から bit24 まで (D0) は受信レジスタ番号であり、bit23 から bit18 まで (S0) は送信レジスタ番号である。同様に、bit17 から bit6 まで (D1, S1) は次の命令の受信と送信のレジスタ番号を指示する。以下は同様である。また、bit31 と bit30 は、それぞれマークビット (M) とイミディエイトデータビット (I) である。マークビットは、分岐制御機能を実行するワードであるかどうかを示すビットである。マークビットが 0 ならば、プロセッサは次のワードで指定さ

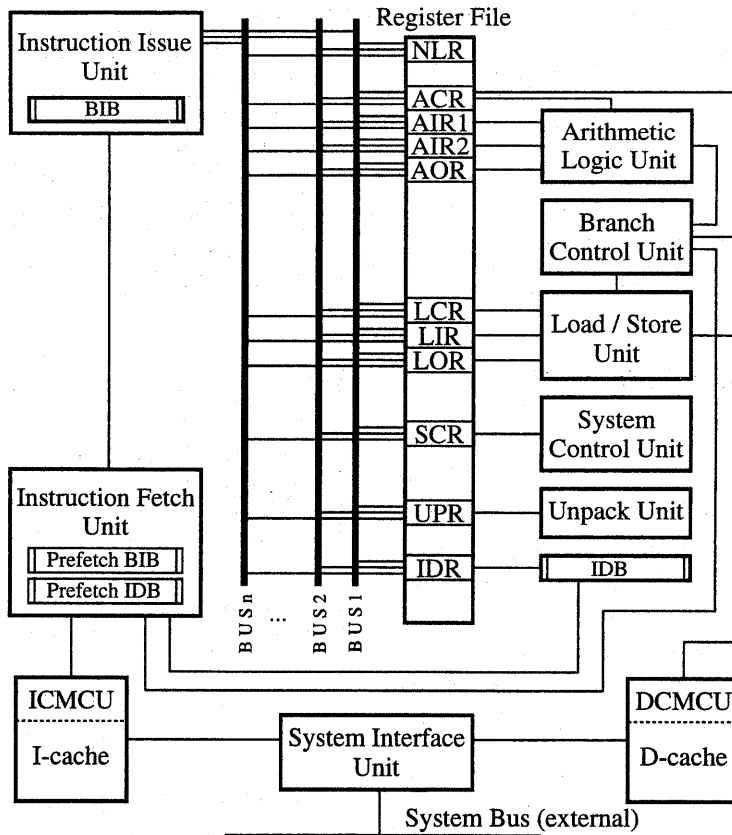


図 1: BISC-2 の基本構成

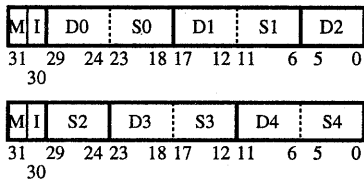


図 2: BISC-2 の命令形式

れる命令を引続き実行する。1 ならば分岐制御機能が実行され、その結果によって、このワードの命令の次に実行される命令が決まる。イミディエイトデータビットは、次のワードがイミディエイトデータであるかどうかを示すビットである。0 ならば次のワードも命令であり、1 ならば次のワードはイミディエイトデータである。

### 2.3 基本動作

BISC-2 の基本動作について述べる。IFU は、内部のプログラムカウンタによって指定されたアドレスのワードをフェッチする。フェッチは ICMCU (Instruction Cache Memory Control Unit) を介して行われ、指定されたワードが I-cache または主記憶から読み出される。フェッチされたデータはデコードされ、命令とイミディエイトデータに分解される。そして命令は IIU へ転送され BIB に蓄えられ、イミディエイトデータは IDB へ転送され蓄えられる。IIU は BIB に蓄えられている命令の先頭から依存関係を調べ、依存関係のない命令を最大でバスの数だけ同時に発行する。IDB はデータを順番に IDR へ送る。転送命令によって IDR からデータを読み出すことで、このイミディエイトデータを使用することができる。

表 1: BISC-2 のレジスタ

種類	用途	レジスタ名
機能レジスタ	0 の出力とデータの破棄	ヌルレジスタ (NLR)
	イミューディエイトデータ	イミューディエイトデータレジスタ (IDR)
	ロードストア	コントロールレジスタ (LCR), インプットレジスタ (LIR), アウトプットレジスタ (LOR)
	算術論理演算	コントロールレジスタ (ACR), インプットレジスタ 1 (AIR1), インプットレジスタ 2 (AIR2), アウトプットレジスタ (AOR)
	アンパック	アンパックレジスタ (UPR)
	システム制御	システム制御レジスタ (SCR)
汎用レジスタ	汎用レジスタ	汎用レジスタ 00 (GPR00), ..., 汎用レジスタ (GPR52)

### 3 BISC-2 の機能

一般的なプロセッサにはレジスタ間転送命令の他に、算術論理演算命令、ロード・ストア命令、分岐命令など様々な命令が用意されている。BISC-2 では命令が本質的にデータ転送のみに限られており、この命令によって算術論理演算ユニットなどの各機能ユニットを直接制御することはできない。各機能ユニットの制御は、機能レジスタへのデータ転送によって間接的に行われる。本節では、各機能ユニットと機能レジスタの動作について説明する。

#### 3.1 算術論理演算

算術論理演算は、算術論理演算ユニット (Arithmetic Logic Unit, ALU) で実行される。ALU の構成を図 3 に示す。

演算を行うためには、転送命令を用いて演算に必要なデータを ACR と AIR1, AIR2 に送る。ACR には加算や減算など実行する演算の種類を指定するデータを、AIR1 と AIR2 にはその演算に必要なデータがすべてそろった時点で演算を開始し、指定された演算に応じて演算結果を AOR に格納する。ただし、AOR が書き込み不可の場合は ALU の動作は停止し、AOR が書き込み可になった後に書き込みが行われる。AOR は演算結果が書込まれると書き込み不可になり、転送命令によってデータが転送されると書き込み可になる。また、演算結果が AOR に書き込まれたとき転送可となり、バスから一度転送すると新しい演算結果が来るまで転送不可となる。

BISC-2 では ALU で行う演算を 6bit で指定す

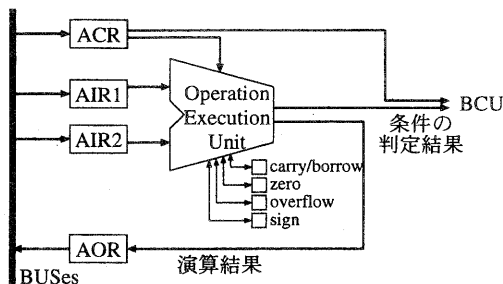


図 3: 算術論理演算ユニット

る。ACR に送るデータは、bit0 から bit29 を 6bit ずつ区切って最大 5 回分の演算を指定し、残りの上位 2bit は分岐で用いる。これについては、3.4 で述べる。演算には、条件付分岐のためにフラグを参照して条件判定を行ない、その結果を BCU に送る演算が含まれている。

#### 3.2 メモリアクセス

メモリアクセスは、ロードストアユニット (Load/Store Unit, LSU) で実行される。LSU には 3 つの機能レジスタ、相対アドレッシングのためのベースアドレスを記憶する 2 つの内部レジスタ、メモリアクセスのモードを記憶する 5bit の内部レジスタ md が接続されている。LSU の機能レジスタは、ロードやストアなどの操作を指定するための LCR、アドレスやデータなどを入力するための LIR、ロードしたデータを出力するための LOR である。LSU で行う操作は 4bit で指定するため、LCR に送るデータには最大 8 回分の操作を 1 度にまとめて指定できる。

ロードやストアなどの操作は LCR と LIR に必要なデータがそろった時点で開始され、命令がロードの場合はデータが LOR に出力される。md の値は絶対・相対アドレッシングモードやアクセスサイズを指定し、ロードとストアは md の値に従って実行される。

### 3.3 アンパック

アンパックは、アンパックユニット (UnPack Unit, UPU) と、UPU に接続されている 1 つの機能レジスタ UPR (UPU Register) で実現する。

UPR にデータを転送すると、そのデータは UPU の内部レジスタに保存される。UPR からデータを転送するときは、内部レジスタの bit31 から bit24 を符号拡張したものが転送される。そして内部レジスタのデータを 8 ビット左へシフトする。このとき下位 8 ビットには 0 が送られる。内部レジスタが 0 になると UPR は転送不可となり、次の UPR へのデータ転送を待つ。

### 3.4 分岐制御

分岐制御は BCU (Branch Control Unit) と BCU に接続されている ACR を用いて行う。また、分岐先のアドレスは LSU の操作で指定され、BCU 内の BAR レジスタに送られる。BISC-2 では、制御機能として無条件分岐、条件付分岐を提供している。

実行する制御機能の種類は、ACR の上位 2bit で指示し、分岐なし、無条件分岐、条件分岐、停止のいずれかを指定する。制御機能の種類を指示するコード番号を ACR に転送した後に、分岐先のアドレスを LSU を介して BCU 内の BAR に送る。無条件分岐の場合は常に BAR に格納されているアドレスに分岐する。条件付分岐の場合は ALU での演算結果により BAR に格納されているアドレスに分岐するか、あるいは分岐せずにそのまま命令を読み続ける。

2.2 で述べたように、I-cache に格納されているプログラム中で制御機能を実行する直前の命令を含むワードは M=1 になっている。M=1 のワードが IFU に読み出され BIB に書込まれた直後に、制御機能が実行される。

無条件分岐は、ACR と BAR にデータが送られた時点で分岐後のアドレスが確定する。従って、ACR と BAR にデータが転送され、マークビットが 1 のワードがメモリから IFU に読み出された時点で、BAR に格納されている分岐先アドレスが IFU 内のプログラムカウンタ (PC) に送られ無条件分岐が実行される。BIB 内の命令をすべて実行する前に、ACR と BAR に必要なデータが転送されマークビットが 1 のワードを読み出すと、実行しているレジスタ間転送は分岐前の命令であり、IFU が読み出すワードは分岐後の命令になる。すなわち、IFU は分岐先の命令を先行して読み出している状態になる。従って、分岐制御による遅延を削減することができる。

また条件付分岐は、分岐後のアドレスが条件の成立/不成立に依存している。条件成立の場合は BAR に格納されている分岐先アドレスに分岐するが、条件不成立の場合は分岐をしない。すなわち、条件成立の場合は、マークビットが 1 のワードがメモリから IFU に読み出された時点で BAR に格納されている分岐先アドレスが PC に送られるが、条件不成立の場合は PC に何も送られず、そのまま次のワードをメモリから読み出す。なお、条件不成立の場合はブロックの先頭アドレスが BAR に送られる。条件は、ALU で算術論理演算を用いて判定する。

ACR と BAR への必要なデータの転送と、条件の判定が完了すると、条件分岐実行後のアドレスが確定する。そのため、BIB 内の命令をすべて実行する前にマークビットが 1 のワードを読み出し条件付分岐を実行すると、無条件分岐の場合と同様に分岐先の命令を先行して読み出している状態になる。これは、条件の判定を条件付分岐の実行に先行して行うことができるからである。このような先行判定により、分岐制御による遅延を削減することができる。

### 3.5 割込み制御

BISC-2 の割込み制御は、システム制御ユニット (System Control Unit, SCU) で行われる。SCU には機能レジスタ (SCU Control Register, SCR), PSW (Program Status Word) バッファ、30 ビットレジスタ pswp がある。PSW バッファは、PSW

を一時的に保存するためのバッファである。pswp は PSW を主記憶上に保存するときのアドレスを保持するレジスタである。BISC-2 では、プロセッサ外部からの割込みとソフトウェアによる割込み生成が可能である。ソフトウェアによる割込みは、割込み番号を SCR へ転送することで発生させることができる。

割込みが発生すると、PSW を PSW バッファへ一時退避し、同時に割込みベクタの読み込みを開始する。読み込みが完了すると、ベクタは IFU のプログラムカウンタへセットされ、バッファに一時退避されている PSW をメモリへ退避する。退避が完了すると割込み処理ルーチンの実行を開始する。

## 4 試作チップと性能評価

BISC-2 のチップ試作を行った<sup>1</sup>。チップは、PolySi 2 層、メタル配線 3 層の CMOS 0.35 $\mu$ m 4.93mm 角である。チップサイズの制約のため機能を省略し、キャッシュとシステム制御ユニットを除いたものとなっている。また、バスの多重度は 3 とし、汎用レジスタの数は 19 とした。トランジスタ数は約 20 万である。現在設計は完了し、製造が行われている。配置配線後のシミュレーションにおいて、周波数 30MHz での正常動作を確認している。

実チップは現在製造中であるので、シミュレーションによって設計した BISC-2 の性能評価を行った。動作周波数は 30MHz とし、Dhrystone 2.1 ベンチマークによる VAX MIPS 値を求め、性能を評価する。バスの多重度を変更したときの VAX MIPS 値を表 2 に示す。多重度 4 で 15.94 VAX MIPS を達成している。次に、インオーダー実行を行ったときとアウトオブオーダー実行を行ったときの性能を表 3 に示す。アウトオブオーダー実行の方が約 7% 大きい値となっており、BISC-2 で導入したアウトオブオーダー機能の有効性が確認できる。

<sup>1</sup>本チップ試作は東京大学大規模集積システム設計教育研究センターを通しローム(株)および凸版印刷(株)の協力で行われたものである。

表 2: バスの多重度と性能

バスの多重度	VAX MIPS
1	13.29
2	15.84
3	15.89
4	15.94

表 3: インオーダー実行とアウトオブオーダー実行の性能 (バスの多重度 3)

	VAX MIPS
in-order	14.86
out-of-order	15.89

## 5 むすび

本論文では、新しい BISC 型プロセッサ BISC-2 の提案を行った。BISC-2 では、BISC-1 の問題点を解決するために、メモリアクセスの高機能化、割込み機能とアンパックユニットの追加、命令の効率的な並列実行とアウトオブオーダー実行機能の追加を行い、高性能化をはかっている。その結果、動作周波数 30MHz で 15.94 VAX MIPS を達成している。

BISC-2 は、一般のプロセッサの 1 命令をいくつかのレジスタ間転送に分割して実行していると考えることができる。このため、BISC-2 の命令列は一般のプロセッサの命令列と比較してスケジューリングの自由度が高く、一般のプロセッサ以上に最適化を行うことが可能である。今後は BISC-2 の性能を効率的に引き出すために、BISC-2 に対して最適化したコンパイラの開発が必要である。

## 参考文献

- [1] 山下幸彦: BISC 型多目的プロセッサ, 電子情報通信学会技術研究報告, CPSY96-70, pp. 9-14 (1996).
- [2] 丸子健一, 山下幸彦: Bus Instruction Set Computer 型プロセッサ BISC-1, 電子情報通信学会技術研究報告, CPSY 98-22, pp. 85-92 (1998).