

COMET II 改と CASL II による計算機と機械語の教育

Computer and Machine Language Education Using COMET II-Kai and CASL II

田川 聖治[†]
Kiyoharu Tagawa

1. はじめに

情報系大学の卒業生には、C 言語や Java 言語など高級言語によるプログラミング能力のみならず、機械語やそれに対応するアセンブリ言語の知識も期待される。高級言語とは異なり、機械語の命令は CPU のハードウェア構造に依存する。このため、機械語を習得するためには、論理回路のレベルから CPU の仕組みについても学ぶ必要がある。ただし、機械語は対象とする CPU の種類によって多少の違いはあるが、1 つの機械語を深く学んでおけば、ほかの機械語の学習は容易である。

機械語の教材として、基本情報技術者試験で出題される仮想計算機 COMET II[1] と、その機械語に対応したアセンブリ言語 CASL II が広く利用されている。しかし、COMET II を教材とする問題点は、そのハードウェア構造が極めて抽象的であるため、計算機の動作原理の理解や、CPU のハードウェア構造を考慮した機械語プログラムの効率性の検討が困難なことである。

本稿では、機械語の新たな教材として、仮想計算機 COMET II の仕様を具現化した COMET II 改 [2] を提案する。COMET II 改によれば、CPU のハードウェア構造とその振る舞いを考慮した機械語プログラムの検討が可能となる。さらに、COMET II 改の命令セットは COMET II のものと同じであるため、機械語プログラムの作成にアセンブリ言語 CASL II が利用できる。

仮想計算機 COMET II の仕様を具現化する試みには、COMET II-STAR[3] などがあり、COMET II 互換プロセッサ [4] も開発されている。COMET II 改の設計では、機械語命令を実行する CPU の物理的な振る舞いを矛盾なく説明できる可能な限り単純な構造を目指した。

2. COMET II の仕様

2.1 機械語命令の形式

語長は 16 ビットであり、1 語長と 2 語長の命令が存在する。また、命令のオペランド数は 0, 1, 2 の 3 通りである。語長とオペランド数の組合せから、COMET II の機械語命令の形式は図 1 に示す 5 種類となる。OP は 8 ビットのオペコードであり、r, r1, r2, rx は 4 ビットの汎用レジスタ識別番号を示す。汎用レジスタは計 8 個

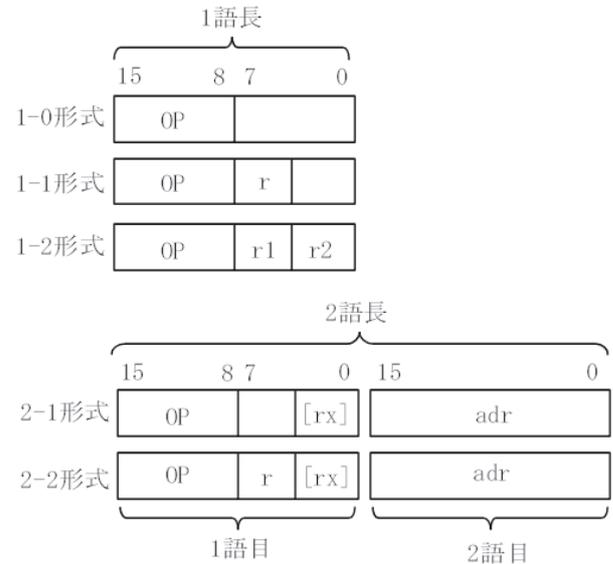


図 1: 機械語命令の形式

ある。識別番号 0 以外の汎用レジスタは指標レジスタとしても使われ、その場合は rx と表記する。また、2 語長の命令の 2 語目は 16 ビットのアドレス adr である。

2.2 アドレス指定方式

実効アドレス (EA) は、メモリ・アドレス (番地) と即値アドレスに大別される。命令の 2 語目のアドレス adr から EA を求めることをアドレス指定と呼ぶ。以下に、COMET II の 4 種類のアドレス指定方式を示す。

- 直接アドレス指定方式：命令に記載されたアドレス adr を、メモリ・アドレスの EA とする。
- 指標アドレス指定方式：命令に記載されたアドレス adr に指標レジスタ rx の内容を加算して、メモリ・アドレスの EA とする。
- 即値アドレス指定方式：命令に記載されたアドレス adr を、即値アドレスの EA とする。
- 指標レジスタを用いた即値アドレス指定方式：命令に記載されたアドレス adr に指標レジスタ rx の内容を加算して、即値アドレスの EA とする。

[†] 近畿大学, Kindai University

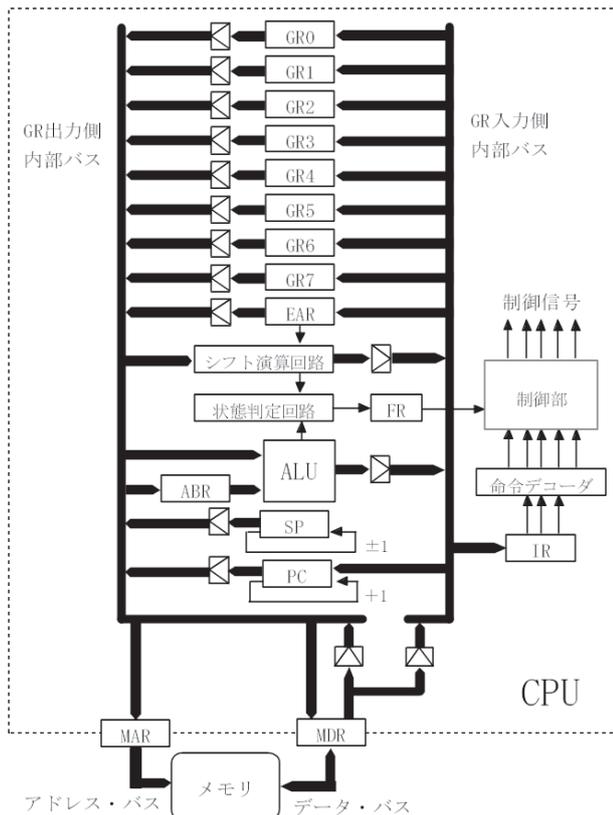


図 2: COMET II 改の回路構成

3. 計算機 COMET II 改

3.1 COMET II 改のレジスタ

以下のように 9 種類のレジスタが存在する。FR を除いて、レジスタは 1 語長のデータやアドレスを保持する。COMET II の命令セットには、GR の内容をインクリメントする 1 語長の命令は含まれないため、定数 1 を保持する定数レジスタはない。また、PC と SP は保持したアドレスの更新機能を持つレジスタとする。

- 汎用レジスタ (GR) : COMET II の仕様に従って、GR は GR0 から GR7 まで計 8 個ある。各 GR はプログラムの実行に際して、1 語長のデータやアドレスを一時的に保持する。また、GR0 を除く GR1 から GR7 は、指標レジスタとしても使用する。
- プログラム・カウンタ (PC) : 次に実行する命令が格納されているメモリのアドレスを保持する。また、PC はレジスタとカウンタの両方の機能を持ち、PC 内の命令のアドレスをインクリメントできる。
- 命令レジスタ (IR) : PC で指定されたアドレスから読み出した 1 語長の命令が格納される。2 語長の命令の場合、IR には 1 語目のみが格納される。

- スタック・ポインタ (SP) : スタック領域の最上段アドレスを保持する。SP は PUSH 命令や POP 命令に従って、スタック領域の最上段アドレスのインクリメントやデクリメントも実行できる。
- メモリ・アドレス・レジスタ (MAR) : CPU が読み書きするメモリのアドレス (番地) を保持する。
- メモリ・データ・レジスタ (MDR) : MAR の番地に読み書きする 1 語長のデータを保持する。
- 実効アドレス・レジスタ (EAR) : 2 語長の命令の 2 語目のアドレス adr が格納される。その後、前述のアドレス指定により、アドレス adr は実効アドレス (EA) に変換されて EAR に保持される。
- ALU バッファ・レジスタ (ABR) : 算術論理演算回路 (ALU) による二項演算では、 $X = X + Y$ のように、一方のデータが上書きされる。ABR は上書きされるデータ X を演算の直前に保持する。
- フラグ・レジスタ (FR) : COMET II の仕様より、演算結果に基づく 3 ビットのフラグを保持する。

3.2 COMET II 改の回路構成

図 2 に CPU を中心とした COMET II 改の回路構成を示す。図 2 の記号 \square は、3 ステート・バッファである。3 ステート・バッファは制御部からの制御信号によって、レジスタや演算回路が出力する瞬間のみ、それらを内部バスに接続する。図 2 では制御信号を省略する。

各 GR と演算回路は入力側と出力側で独立した内部バスに繋がっている。このため、2 つの異なる GR に対するデータの書き込みと読み出しが同時に行える。出力側から入力側の内部バスにデータを送る場合、内部バスに接続したシフト演算回路を通過させる。また、MDR の出力先には 2 つの 3 ステート・バッファが存在し、何れかを選ぶことでメモリから MDR に読み出された命令やデータを入力側と出力側の内部バスに取り出せる。

3.3 COMET II 改の制御部

制御部には以下の 6 個の状態があり、各状態に応じた制御信号を生成して CPU を動かす。制御部は IF 状態から始まり、幾つかの状態を経て EX 状態で終了する。

- 命令の読み出し (IF) : PC で指定された命令の 1 語目をメモリから MDR を経て IR に読み出す。次に、PC 内の命令のアドレスをインクリメントする。
- 命令の解釈 (ID) : IR 内の命令を解釈する。このとき、命令の 2 語目があるか否かも判定する。

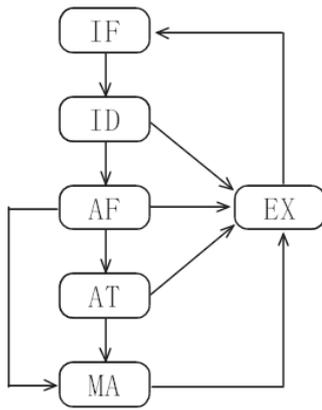


図 3: 制御部の状態遷移図

- アドレスの読み出し (AF) : PC で指定された命令の 2 語目をメモリから EAR に読み出す。次に、PC 内の命令のアドレスをインクリメントする。
- アドレスの変換 (AT) : 命令が指標レジスタを使用している場合、EAR の内容を実効アドレス (EA) に変換する。具体的には、図 2 の CPU において、EAR の内容を ABR に複写する。次に、指標レジスタと ABR の内容を ALU で加算し、その結果の EA を ALU の出力から EAR に上書きする。AT 状態では ALU を使用しても FR は設定されない。
- メモリの読み書き (MA) : EAR 内の EA がメモリ・アドレスである場合、EAR 内の EA を MAR に書き込む。メモリからデータを読み出す場合、MAR で指定されたデータが MDR に複写される。
- 命令の実行 (EX) : CPU 内の演算回路やレジスタを制御して IR 内の命令を実行する。EX 状態における具体的な処理は命令の種類により異なる。

制御部の振る舞いは図 3 の状態遷移図で表現できる。制御部の状態遷移は、クロックパルスに同期して、以下のように命令の語長とアドレス指定方式で決まる。

- 1 語長の命令 : IF → ID → EX
- 直接アドレス指定方式 (メモリ・アドレス) : IF → ID → AF → MA → EX
- 指標アドレス指定方式 (メモリ・アドレス) : IF → ID → AF → AT → MA → EX
- 即値アドレス指定方式 : IF → ID → AF → EX
- 指標レジスタを用いた即値アドレス指定方式 : IF → ID → AF → AT → EX

4. 機械語プログラムの効率性

機械語命令のレベルではあるが、COMET II 改を用いた機械語プログラムの効率性の検討例を示す。

以下の 2 つの命令は汎用レジスタ GR1 に 1 を加算する。メモリ・アドレス ONE の内容は 1 とする。

命令 1A : LAD GR1, 1, GR1

命令 1B : ADDA GR1, ONE

各命令を実行する COMET II 改の振る舞いは以下の通りであり、ノイマンのボトルネックの観点から、メモリからデータ 1 を読み出す命令 1B は効率が悪い。

命令 1A : IF → ID → AF → AT → EX

命令 1B : IF → ID → AF → MA → EX

以下の 2 つの命令は GR1 を 0 に初期化する。

命令 2A : XOR GR1, GR1

命令 2B : LAD GR1, 0

各命令を実行する COMET II 改の振る舞いは以下の通りであり、ノイマンのボトルネックの観点から、1 語長の命令 2A に比べて 2 語長の命令 2B は効率が悪い。

命令 2A : IF → ID → EX

命令 2B : IF → ID → AF → EX

5. おわりに

機械語の新たな教材として、COMET II 改を提案した。COMET II 改によれば、機械語プログラムの作成に CASL II が利用できるとともに、CPU の振る舞いを考慮したプログラムの効率性の検討が可能となる。

今後の課題は、COMET II 改の振る舞いを観察できるアニメーション付きシミュレータの開発である。

参考文献

- [1] 独立行政法人 情報処理推進機構 : 情報処理技術者試験, 別表 1, アセンブラ言語の仕様, 2011.
- [2] 田川聖治 : COMET II 改と CASL II による計算機と機械語の基礎 (増補版), デザインエッグ株式会社, 2018.
- [3] 遠藤敏夫 : 基礎から学ぶコンピュータアーキテクチャ, 森北出版株式会社, 2008.
- [4] 河合一慶, 宮内新, 荒井秀一 : COMET-II 互換プロセッサによる CPU 設計演習環境の開発, 情報科学技術フォーラム (FIT2002), pp. 189-190, 2002.