

アセンブリ言語教育支援システム Sim AI の設計と実装

小林晴紀^{†1} 和田幸一^{†2}

^{†1} 法政大学大学院 ^{†2} 法政大学

本稿は法政大学応用情報工学科のアセンブリ言語教育を支援するシステム「Sim AI」(CASL II Simulator for Applied Infomatics of Hosei University)の設計及び実装について述べる。システムは指導者と学習者の教育レベルやその利用形態に応じた支援を可能とする標準モードと拡張モードを提供する。標準モードでは、初学者の利用を考慮したアセンブリ言語プログラミングを支援する入力補助機能や、プログラムの可視化シミュレーション機能、デバッグ機能を装備する。また、アセンブリ言語と機械語の対応を容易く把握できるように、リロケートブルな機械語を可視化することを可能としている。拡張モードでは数値表現を効果的に可視化する機能やアドレッシングモードの拡張機能、マクロ命令のユーザ定義機能、機械語プログラミング機能を装備する。

1. はじめに

アセンブリ言語教育には様々な困難が伴う。アセンブリ言語は記述の抽象度が低く、各々の命令は人間にとっては単純な処理であるため、プログラムの作成・理解が難しい。また、アセンブリ言語と機械語はほぼ一対一対応しているため、学習者は、機械語が計算機でどのように処理するかを想定しながらプログラムを作成しなければならない。そこで、本研究はアセンブリ言語教育を支援するシミュレータシステム Sim AI(CASL II Simulator for Applied Infomatics of Hosei University)の設計、開発を行う。本システムは法政大学理工学部応用情報工学科のアセンブリ言語講義を支援することを目的とする。以下にシステム全体像を示す。本システムは支援機能として標準機能セット(図1①～③)と拡張機能セット(図1④～⑦)を提供する。標準機能セットは、初学者がアセンブリ言語と計算機の依存関係を確実に理解することを目的とする。

そのために有効と考えられる、実行状態へのプログラム変換過程、計算機の内部状態とプログラム動作を忠実に可視化する機能を装備する。また、学習効率を上げるプログラム作成・理解負担を軽減するエディタ機能を装備する。拡張機能セットは、データ構造やマクロ命令のユーザ定義を可能とする。この機能を用いることで学習者は、アセンブリ言語レベルで抽象度の高いプログラミングを学習できる。この他にも、拡張機能セットは機械語プログラミングを可能とする。本システムではアセンブリ言語とその仮想計算機として CASL II・COMET IIを採用する[1]。COMET IIは仮想計算機であり実機と比較して、アーキテクチャが単純で分かりやすいという特徴がある。CASL II・COMET IIは基本技術者試験にも採用されている。本システムの拡張機能セットは、CASL II・COMET IIの単純なアーキテクチャを維持した上で、学習不可能なアドレッシングモードや数値表現を効果的に学習できる

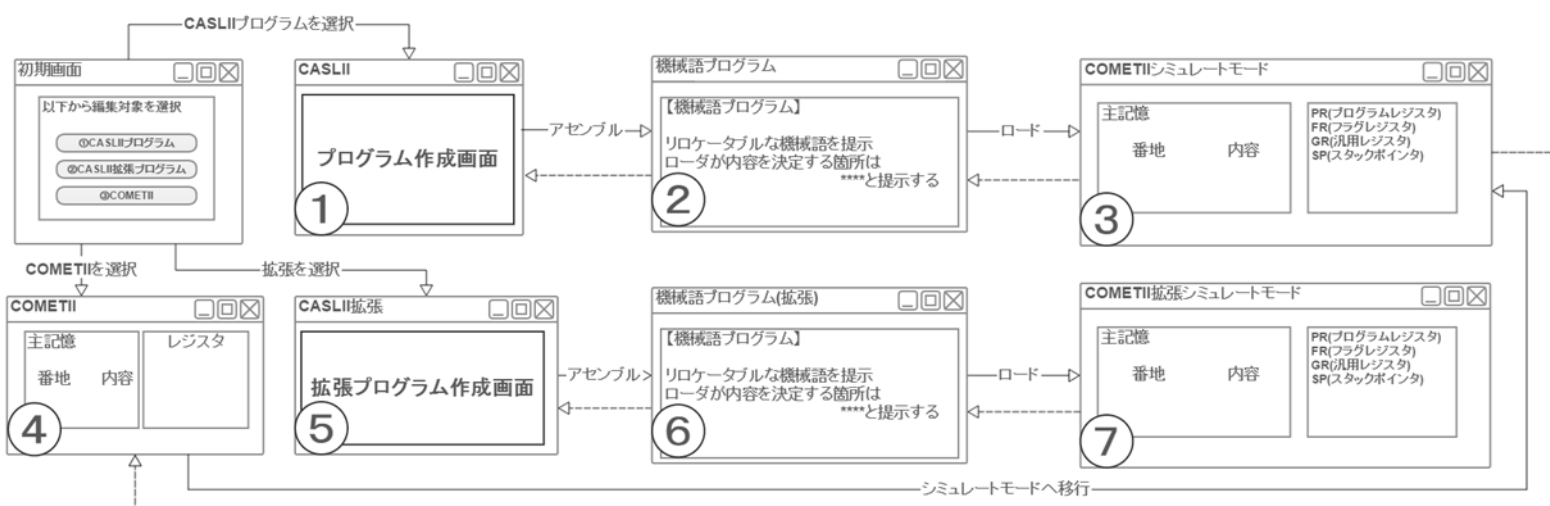


図1 シミュレータ画面遷移

^{†1} 法政大学大学院
Graduate School of Hosei University
^{†2} 法政大学
Hosei University

2. 関連研究[2][3][4][5]

アセンブリ言語教育支援システムに関連する研究として、教育用 MIPS プロセッサシミュレータシステム、計算機アーキテクチャ教育支援システム、初等アセンブラプログラミング評価支援システム、計算機シミュレータ ECAS がある。これらの支援システムの概要を以下に記す。

➤ 教育用 MIPS プロセッサシミュレータシステム[2]

会津大学が開発したプロセッサ内部の動作を可視化するシステム。複数の MIPS マイクロアーキテクチャに対応し、コンピュータ内部の動作や構造を視覚化できる。このシミュレータの課題として、データバスや制御部の定義を変更可能とすることが挙げられている。

➤ 計算機アーキテクチャ教育支援システム[3]

ノイマン型コンピュータ教材 VisuSim を用いた計算機アーキテクチャ教育支援環境。VisuSim は計算機内部の構造・動作の可視化やアセンブリ言語プログラミング支援などを支援する教材である。タブレットとしてもアプリケーションとしても起動可能という特徴を持ち、その特徴を利用したメール機能を組み込むことで学習者指導者間の質疑応答や学習者間の情報交換等の教育支援を円滑にしている。この教材を用いたシステムの課題として前述の基本機能を活かすための補助機能の開発や、LMS との統合が挙げられている。

➤ 初等アセンブラプログラミング評価支援システム [4]

CASL を教材とした授業におけるプログラム評価支援システム。評価対象のプログラムが提示した問題の題意を満たしているかどうかの判定とプログラムに対するアドバイスの作成を行う。このシステムの課題として、冗長命令検出やアドバイス文の精度向上等が挙げられている。

➤ 計算機シミュレータ ECAS[5]

計算機アーキテクチャ教育支援システムの研究はこれまでもいくつか行われている。その中で可視化シミュレーション機能を有する計算機シミュレータは学習効果があるとされている。計算機シミュレータの一つである ECAS では、計算機の構造とレジスタや ALU などの部品との関係、部品と論理ゲートレベルとの関係が観察できるように設計されている。

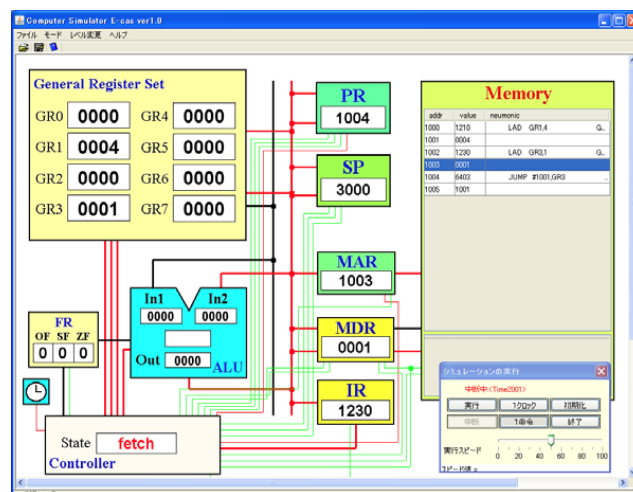


図2 ECAS の画面例

3. 提案システムの設計方針

3.1 先行調査

本システムを新規に開発する上で、システムにおける支援が有効と考えられる箇所を明らかにするためにいくつかの先行調査を行った。はじめに、実際の教育現場で使用されている CASL II シミュレータの運用法を調査した。ここでは、法政大学のアセンブリ言語講義について調査を行った。そして、シミュレーション使用環境及び求められている機能を明らかにした。調査して得られたシミュレーション使用環境及び必要と考えられる機能を表 1 にまとめる。

表 1 先行調査で得られたシミュレーション使用環境

運用法	自学自習、演習、講義解説
主な学習者の特長	プログラムの理解作成に苦勞を要する。 他分野(計算機アーキテクチャ、C言語)の知識を有する
指導者	プログラムの視覚的解説
必須機能	計算機内部状態を可視化しつつプログラムをシミュレーションする機能 プログラム入力補助機能

次に、表 1 のシミュレーション使用環境の観点から既存 CASL II シミュレータについて調査・比較を行う。以下に比較対象である既存の CASL II シミュレータの一覧表を示す。

表2 既存のCASL II シミュレーター一覧

1	CasBuilder
2	WCASL- II
3	CASL2 Visible Inside
4	CASL2シミュレータ
5	ドリームキャスル
6	中身が見えるCaslxxxx
7	CASL2 cmd
8	CASL2する?
9	キャスルシミュレータ
10	CASL & COMET Simulator
11	CCasl II
12	CASL2000 for Windows95,98/NT4.0
13	EduCasl
14	Hello CASL II (ライセンス版)
15	IPA公式シミュレータ
16	InfoCASL
17	Web版キャスル II シミュレータ

調査の結果、計算機内部状態やプログラム動作可視化機能が効果的なこと、学習者の負担を軽減する入力補助機能の提供が効率的学習の観点から望ましいなどの結論を得る。

既存シミュレータの中には計算機アーキテクチャとアセンブリ言語の協調学習を目的とするものがある。(W-CASL II)また、サンプルプログラムを取得・シミュレートするシミュレータも存在する。(Hello CASL II)しかし、いずれも可視化シミュレート機能が対象とするアーキテクチャが不十分であり、プログラム入力補助機能を有したものはドリームキャスルのみであった。特に主記憶の可視化については、格納されているものの説明が不十分で、命令なのか参照している番地なのか分かり辛い。先行調査の結果、既存シミュレータによって条件を満たすことは難しいと判断する。以上が、新規にCASL II シミュレータを開発するに至った要因である。

3.2 設計方針

標準機能セットと拡張機能セットの設計方針について述べる。標準的な機能セットは、学習者がアセンブリ言語の基本を正確に理解できるような機能を提供する。そのために、CASL II・COMET II システムの仕様を忠実に再現する。そして、主機能として可視化機能、デバッグ機能、プログラム入力補助機能を装備する。また、学習者のレベルに応じた機能の提供を可能にするため、上級者向けの拡張機能セットを提供することを考えた。拡張機能セットは、CASL II・COMET II システムでは不十分だと考えられるアドレス指定方式の理解や、データ構造の可視化支援機能、マクロ支援、機械語プログラミング機能等を用意する。以下、標準機能セットの主機能及び拡張機能セットについて概要を

述べる。

(1)可視化機能

関連研究[2][3][5]のように可視化が有意な教育効果を示すという研究報告が多い。関連研究[2][3]はいずれも計算機教育のことだが、これはアセンブリ言語教育とも関連性が高いため、有意な教育効果を示すと考えられる。例えば、ラベルやリテラルについて、プログラムの実行アドレスを可視化しながら実行することで、それがメモリにどのように割り振られているのか理解するのが容易となる。ここでは、CASL II プログラムで操作可能なCOMET II レジスタ内容やメモリ内容の可視化機能を提供することを考える。その際の内部表現についても工夫する。また関連研究[2][3]のような、アセンブリ言語と関連性が高い計算機内部の構造や動作が可視化されていれば計算機アーキテクチャと知識を関連付けることが可能なため、効果的な学習ができると推測される。

また、従来のシステムでは、アセンブリプログラムと機械語の対応が掴みづらいという欠点があった。提案システムでは、アセンブリプログラムと機械語の対応が掴みやすくするため、ロード後の記憶装置に加え、ロード前の機械語の可視化を試みる。CASL II・COMET II システムは機械語を記憶装置へロードする時、リロケータブルロードを用いる。ここで、機械語プログラム内の相対アドレスの箇所は実アドレスに変換される。つまり、提案システムではロード後の記憶装置に加え、ロード前の機械語を可視化することにより、機械語とアセンブリ言語の対応が掴みやすくなる。



図3 リロケータブルな機械語の提示例

(2)デバッグ機能

アセンブリプログラムをデバッグするために必要な機能を搭載する。アセンブリ言語において実行時エラーは原因を突き止めるのが困難なことが多い。そのため、プログラムの振る舞いが理解できるだけでなく、エラー原因を絞り込めるような機能を提供する。主な機能として実行単位を調節する機能、実行履歴を確認する機能、シミュレーション条件の設定機能が挙げられる。また、シミュレータ使用者がプログラムを途中で中断してレジスタ内容やメモリ内

容を変更して実行フェーズに復帰できるといったインタラクティブ性も重要である。この機能を搭載することで、シミュレータ使用者は一つプログラムで多数の実行のパターンを検証できることから、効率的な学習が期待できると考えられる。学習に役立つという面では、参照されたレジスタ・メモリや、内容が更新されたレジスタ・メモリを目立たせて、シミュレータ使用者の注意を喚起するといった機能も挙げられる。

(3) プログラム入力補助機能

プログラム作成負担を軽減することで効率的学習が期待できる。高級言語の統合開発環境では、シンタックスのハイライト明示や命令記述例の紹介を記述者の入力に即して動的に行うものが存在する。主な機能としてシンタックスハイライト、オートコンプリート、プリティプリンタ、スペルチェッカー等が挙げられる。これらの機能はプログラム作成負担を軽減する一方で、シンタックスを正確に覚えないなど学習の妨げとなることも考えられる。自力でプログラムを記述する能力を身につけるために、これらの機能は切替できるようにすることが望ましいと考える。

(4) 拡張機能セット

標準機能セットでは支援が不十分だと考えられる箇所がいくつか考えられる。例を挙げると CASL II・COMET II システムはマクロ命令をユーザが定義することを許していない。また、プログラムカウンタやスタックポインタをプログラム中で使用できない。これらの制約を無くすことで、マクロ命令の本来の用途や新たなアドレス指定方式の学習が可能となる。拡張機能セットを用いた拡張モードでは、これらの制約をなくす。

機械語の理解を深めることで、計算機での実行を想定した無駄のないアセンブリ言語プログラムを作成できる。ここでは、機械語プログラミング機能を提供することで機械語の理解を支援することを考えた。

4. Sim AI の開発 [6]

4.1 システムの概要

提案システムは、アセンブリ言語 CASL II を記述するためのエディタ・可視化機能を実現するためのプログラム動作表示画面、シミュレーション機能を実現するためのデバッガから構成される。尚、従来シミュレータに見習い、アセンブリ言語と関連性が高い計算機内部の動作や構造を Java Swing を利用して可視化シミュレーションを実現する。また、レジスタ内容の 2 進、10 進、符号無し 10 進、16 進、文字での提示も実現した。

次に提案システムのユーザインターフェース概要を図 4 に示す。図 4 は、制御画面であると同時にプログラム動作を可

視化する。ユーザインターフェースは上部にファイル操作をはじめとするアクションコマンドが実行されるメニューバーやツールバーがあり、その下にエディタ画面(図 4C)、主記憶表示画面(図 4B)、レジスタ表示画面(図 4A)、インフォメーション画面(図 4D)がある。COMET II のシミュレート実行時には画面が制御部や ALU など COMET II の内部情報を表示する画面に切り替わる。

以下、提案システムの特長である計算機内部状態・プログラム動作の可視化機能やユーザ入力補助・リロケートブルな機械語可視化について概要を述べる。

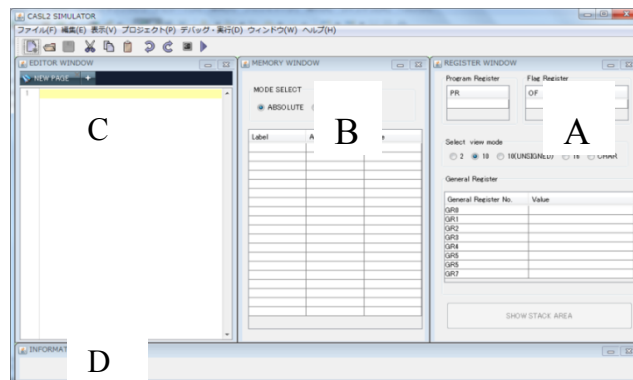


図 4 提案シミュレータの GUI 画面

4.2 ユーザ入力補助機能

コピー、切取、貼付、検索、Undo、Redo 等のエディタとしての基本機能に加えて、プログラムの行番号の表示、オートコンプリート、命令セットの情報表示、スペルチェッカー、シンタックスハイライトを実装する。オートコンプリートに関しては、候補となる命令及び例題コードを表示し、命令を学習者に選択させる。この機能は学習者のコード記述負担を軽減するというメリットがある反面、命令セットの学習面では悪影響となる機能である。そのため、この機能を使用しないようにすることが可能である(Ctrl + Space により切り替え可能)。シンタックスハイライトに関しては、命令系、リテラル、コメント、レジスタ、特殊文字によって、ハイライト色を場合わけして施すことによりプログラム構造を理解しやすいものとしている。以下にエディタの機能例を示す。

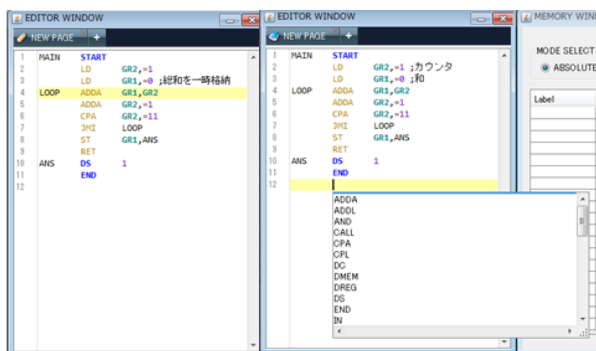


図 5 エディタ画面(左はオートコンプリート無し, 右はオートコンプリート有り)

4.3 可視化機能

(1) レジスタ表示画面

汎用レジスタとスタックの内容は 2 進数や符号無 10 進数, 符号付 10 進数, 16 進数の中からユーザが選択した形で提示する. 2 進以外の提示形式に関しては, 嵩上げ表現, 浮動小数点表現, 補数表現をサポートする. さらに, 学習者の理解を助けるため, 1 命令実行毎に次に参照されたレジスタを黄色に, 更新されたレジスタ内容を赤色に変化させ, 使用者の注意を喚起する. また, プログラム内で文字列またはポインタとしてデータが設定された場合は文字列またはポインタとして提示する. 汎用レジスタが指標レジスタとして使われた場合は (GRx + 番地) または (数 + 番地) と注釈する. これは従来の CASL II シミュレータにはない新たな機能である.

(2) 主記憶表示画面

図 4B のテーブルに機械語がロードされている主記憶の領域を提示する. 二語で構成される命令語(第二語がアセンブラ命令 (DC DS) やマクロ命令 (IN, OUT, RPUSH, RPOP), ラベル, リテラルに割り当てられた番地となる命令)に対しては, 二語目の番地の内容を注釈する. スタック領域と定数領域, リテラル領域, プログラム領域は, ユーザがみて区別がつくように提示する. 尚, 番地の内容もレジスタ表示画面と同様の表示形式を選択できるようにする. このように可能な限り内部のビットパターンについて説明を加えることで, アセンブリプログラムと機械語プログラムの対応を掴みやすくなると考えられる. しかし, プログラムが独自のデータ構造をプログラム中で定義した場合, 図 4B のような 1 語をテーブルの 1 行で表現形式ではデータ構造が把握しづらいという欠点がある. そのため解決策としてデータ構造可視化支援機能を提供することを考えた. この機能を使うことでユーザは主記憶可視化の形式を設定することが可能になる.

(3) プログラムの制御移行

プログラムの呼び出し元と呼び出し先を画面上に提示して, 制御の移行を可視化した. 制御の移行を見せることでプログラム構造をわかりやすくするという狙いがある. また, この可視化はアセンブリプログラムレベルと機械語レベルの双方で可能となっている. ここでは, 呼び出し元と呼び出し先を OS, メインプログラム, サブルーチンという分類で分け, 制御の移行を矢印で表現している.

(4) リロケートブルな機械語

アセンブリプログラムをアセンブルしたとき, 記憶装置にロードされる前のリロケートブルな機械語を提示する. ここでは, ロードされる番地に依存する箇所を **** と表現している. ここで提示される内容は, アセンブリプログラムと機械語の対応が掴みやすくなるためにも, ラベル名やリテラル等内容に関する注釈を加える. ここで生成されたリロケートブルな機械語は, ユーザが主記憶の任意の番地にロード可能となっている.

(5) 計算機

仮想計算機 COMET II の内部構造を独自に定義して可視化を行う. 計算機レベルの可視化を行うことで計算機が機械語プログラムを制御する仕組みを効果的に理解させることが可能となる.[5]

4.4 デバッグ機能

プログラム全体を実行した後の値を観察するだけでは, 学習者がプログラムの振る舞いを理解する上で不十分な場合がある. その対策として以下の機能を実装した. 尚, これらの機能は図 6 上部メニューバーの「デバッグ・実行(D)」から選択可能である.

➤ トレース実行

プログラムの実行履歴を別画面で表示する. 実行履歴は上から実行順番に従ってソートされる. 表示内容は PR, FR, GR, SP の内容・メモリ番地及び内容・命令である. ループ文やサブルーチンの箇所は, ループの繰り返し回数やサブルーチンの実行回数等を横に注釈して表示される. そうすることにより, ループ文やサブルーチンの構造や振る舞いの理解するのを手助けする.

➤ ステップ実行

プログラムを 1 ステップ毎に実行する. ユーザの設定によって繰り返し処理を構成する部分を 1 ステップごとに実行したり 1 行ずつ実行し

たりできるようにする。以下で述べるレジスタ・メモリ内容変更機能と組み合わせることにより、学習者-シミュレータ間のインタラクティブなやり取りを可能とする。また繰り返し処理の回数や初期値を変数と仮定して、レジスタの値等を関数として提示する。

- 手戻り実行
ステップ実行を行ったプログラムを命令文一行単位で手戻りする。
- ブレークポイントの設定/解除
ユーザが選んだ任意の行でプログラムを停止する。また、ユーザが指定した範囲内を実行できる。この機能をエラー表示があるときでも可能な限り行えるようにすることでエラーの原因究明の手助けをすることが可能である。
- 停止条件文の追加
レジスタ・変数・アドレスの値や命令の実行回数を用いて作った停止条件文をプログラムに対して設定する。条件式が真の時や偽の時、式の値が変化した時に実行を中断する。
- レジスタ・メモリ内容変更
ステップ実行・手戻り実行・ブレークポイント・停止条件文のようなデバッグによるプログラムの動作停止の際呼び出される。シミュレータ使用者は任意のレジスタ・メモリ内容の値を書き換えることが可能。何も入力しない場合、実行終了となるか、ステップ実行、手戻り実行モードに復帰するか選択可能。尚、ここで書き換えた値はプログラムの記述内容を書き換えるものではないので注意が必要である。
- 命令の実行回数表示
各行やユーザが指定した一定範囲の各命令実行回数を表示する。分岐命令やサブルーチンコールを含むプログラムの理解の手助けをする。
- クロスリファレンス
クロスリファレンステーブルを出力する。プログラムで定義したラベルやサブルーチンの定義された番地や参照している番地を表にまとめて出力する。整列順序としては定義した名前のアルファベット順や定義された番地の若い順から選択可能。

4.5 拡張モード

- アドレス指定方式の拡張
プログラムレジスタPRとスタックポインタSPをオペランドとして使用できる。様々なアドレッシングモードを理解できるようにするためには必要と考えられる。
- ユーザ定義のマクロ命令
START 命令の直前に記述する。最初の一文で<M>と記述し、次の一文で命令の記述方法を定義する。次に命令群を記述する。この塊を1ブロックとする。START 命令の直前であればある程度のブロック数を定義できるようにする。ユーザがプログラムをアセンブルした時に①で行ったような検査をしてアセンブル結果と一緒に検査結果を提示する。命令名やオペランドは、予約語である GR0~GR7 や定数の記述形式、既に存在する命令名以外であれば自由に定義可能とした。以下にプログラム例を示す(赤がマクロ定義部分)。

```

<M>
INC    $1    ; ここに必要なに応じて命令の説明を加える
        ADDA  $1,=1
START
        :
        INC   A
        :
        RET
        END

```

図6 マクロ定義及び使用例

主記憶の提示画面では、ユーザが定義したマクロ命令に対応する箇所の色付けする。マクロ命令はシミュレートモードのステップ実行で1命令として実行するか選択できる。

また、記述されたマクロ命令はプログラムが使えるものかどうか検査する。具体的には、オペランドにどのようなものが記述できるかを検査して以下の4つに分類する。

- (1)レジスタ(r)
- (2)アドレス(adr)
- (3)レジスタとアドレスの両方。
- (4)レジスタとアドレスも記述できない(マクロ命令として使用不可能)

検査結果が(4)の場合、それをユーザに通知して修正を促す。

検査結果が(1)~(3)の場合、ユーザがプログラム中で使えるようファイル形式で保存する。

➤ 機械語プログラミング機能

主記憶とレジスタに直接書き込みできる。2語で構成されるレジスタ・主記憶間の命令を書き込んだ時、2語目のビットパターンをクリックするとその参照箇所にとべるようにする。命令の対応表も必要に応じて提示できるようにする。命令が書き込まれた時、主記憶とPRがセットされた時、シミュレートモードへ移行できる。

5. 実現

移植性を考慮して Java を用いて実現する。GUI 構築は Java SE 向けのクライアント UI ライブラリの Swing を利用している。現時点で実現しているのは、アセンブリプログラムをアセンブルすること、機械語プログラムを記憶装置に読み込み実行し、実行後のレジスタと主記憶の内容を提示すること、実行途中のレジスタや主記憶の内容を取得すること、エディタにおけるシンタックスハイライト等である。このようなシミュレータとして基本的な機能は概ね実装完了している。拡張機能セットやデバッグ機能における停止条件文設定操作は未完成である。

6. まとめ・今後の課題

本稿ではアセンブリ言語における教育支援システム Sim AI の開発、設計について述べた。提案シミュレータは学習者 - シミュレータ間のインタラクティブなやり取りを可能にし、かつ COMET II のレジスタやメモリを効率的に理解させる可視化や学習者のプログラミング能力を向上させるような主記憶のロード設定や表示画面を実現している。今後の研究の課題は未実装のシミュレーション機能と COMET II におけるプログラム動作を可視化する機能を実装することである。

そして 提案システムを実際の教育現場で運用し評価を行う。また、豊富なシミュレーション機能を利用することで、関連研究[4]のようなプログラム評価支援システムへ応用も可能と考えられる。

参考文献

- [1]. 情報処理技術者試験出題範囲 pp.5-12(2011 - 07)
http://www.iittec.ipa.go.jp/1_13download/shiken_yougo_ver2_0.pdf
- [2]. 西牧悠二, 北道敦司, 宮崎敏明, “内部動作を視覚化した教育用 MIPS プロセッサシミュレータシステムの開発” 電子情報通信学会論文誌 Vol. J96 - D No.10 (2013)
- [3]. 今井慈郎, 金子敬一, 中川正樹, “計算機アーキテクチャ教育支援システムの開発と強調学習への適用” 電子情報通信学会論文誌 D Vol. J91- D No.2 pp.188-199(2008)

[4]. 渡辺博芳, 荒井正之, 武井恵雄 “事例に基づく初等アセンブラプログラミング評価支援システム” 情報処理学会論文誌 Vol.42 No.1(2001)

[5]. 堀匠吾, 井本直希, 那須聖史, 中西通雄: “計算機シミュレータ ECAS の学習機能およびユーザ定義部品作成機能の実装”, JSiSE 学生研究発表会 (2012-3)

[6]. 小林, 林, 和田 アセンブリ教育支援システムにおける学習用 CASL II シミュレータの提案 - CASL II & COMET II シミュレータ仕様書, 法政大学理工学部応用情報工学科計算機科学研究室 テクニカルレポート TR14 - 6, 1 - 16, (2014 - 06)