

COMP-X シミュレータシステム

宮 原 康 実*, 横 山 文 規**

Simulator System for COMP-X

Yasumi MIYAHARA*, Fuminori YOKOYAMA**

(Received August 31, 1979)

The implementation of a simulator for the COMP-X computer is reported in this paper. The COMP-X is a virtual machine for the teaching of assembler language. The simulator for this simple computer has been developed to assist students taking the qualifying examination for information processing engineers. This is also used to teach the concept of data transfer in the processing unit for students of assembler language.

1. はじめに

最近では、高校、大学において、情報処理教育が盛んに行われている。その中で、ハードウェア教育およびソフトウェア教育の目標の一つとして通商産業省機械情報局情報処理振興課が実施している情報処理技術者試験の内容が対象となっている場合が少なくない。

この試験は、他の国家試験と異なり、合格者に対して資格、免許等を付与する試験ではなく情報処理技術者として備えるべき一定の能力、技術力等について認定しているものであり、次の3つの試験区分に分れている。

1. 特種情報処理技術者試験

情報処理システムの設計に必要な知識および技能。

2. 第1種情報処理技術者試験

プログラムの設計およびプログラムの作成に必要な知識および技能。

3. 第2種情報処理技術者試験

プログラムに必要な知識および技能。

これらの試験を受験する場合、プログラム言語によって問題を選択する。その言語には FORTRAN, COBOL,

PL/I およびアセンブラ言語がある。

本学においても受験を希望する学生が多数ありプログラム実習等も熱心に行われている。このような背景において、第1種では、昭和52年度からアセンブラ言語が必修となり、アセンブラ言語の実習の要求が高まってきている。また、講義におけるアセンブラ言語の実習において、教育の目的とするものは、アセンブラ言語による複雑なプログラム作成能力ではなく、機械語のレベルで、基本的な演算やデータ処理をどのように計算機が行っているのかを習得することである。この場合に既存のシステムを利用すると、アセンブラ言語が複雑であり、本来目的とする指導を行う前にアセンブラ言語自身の指導を行わなければならない。そのため、指導を行う上でもっと簡単なアセンブラ言語の要求もある。

今回、情報処理技術者試験における仮想計算機の仕様が決まったのをきっかけとして、この仕様にあう計算機のシミュレータとアセンブラを作成した。

このシステムは FACOM 270-30のアセンブラ言語で書かれており、FACOM 270-30のモニタのもとで処理プログラムとして動作するように作られている。

以下このシステムについて述べる。

* 電子計算機センター (Computer Center)

** トヨタカラー愛知株式会社 (Toyota Corolla Aichi Co., Ltd.)

2. COMP-X* の仕様

ここでは、COMP-X の仕様の概要について述べ、細い点については文献1)を参考とされたい。また、COMP-X のシミュレーションにともない多くの学生にバッチ処理で利用できるように付加した機能について以下に示す。

2.1 ハードウェアの仕様

(1) 制御・演算装置

レジスタとして BR (base register), GR0~3 (general register), SC (sequence counter) および CC (condition code register) をもっており、BR は実効アドレスの決定、GR0~3 は算術演算と論理演算に用い、その中で GR1~3 は指標レジスタとしても用いる。また、SC はある命令の実行中は、その命令が絡納されているアドレスがセットされる。CC には加算命令または減算命令の実行が終わったとき、その演算結果

表-1 命 令 コ ー ド 表

2進表示	16進表示	ニック表示	読み方	機能		
0000	0	H J	Halt and jump	SCに実効アドレスを入れて停止する。その後、スタートボタンを押すとSCの示すアドレスから再び命令の実行が開始される。この命令のGRフィールドは無視される。		
0001	1	J N Z	Jump if GR is not zero	GRフィールドで指定するGRの内容が0でないとき、実効アドレスにジャンプする。		
0010	2	J C	Jump on condition	GRフィールド(第4, 第5ビット)の値によって次のような意味をもつ。		
				第4ビット	第5ビット	意 味
				0	0	何もせずに次の命令に移る。
				0	1	CCの内容が1のとき、実効アドレスにジャンプする。
				1	0	CCの内容が0のとき、実効アドレスにジャンプする。
1	1	無条件に実効アドレスにジャンプする。				
0011	3	J S R	Jump to subroutine	GRフィールドで指定したレジスタに現在のSCの内容+1をセットして、実効アドレスが指定する語の内容をSCとBRに入れる。BRの下位8ビットは0にリセットされる。この命令はBRの内容を変更する唯一の命令であり、任意のアドレスにジャンプすることができる。記憶ブロックの異なるサブルーチンへのジャンプとサブルーチンからの戻りはJ S R命令で行う。		
0100	4	S F T	Shift	指定されたGRの内容を、符号ビット(第0ビット)を除いて、右又は左へ、ADフィールドで指定されるビット数だけシフトする。XRフィールドが、0なら右へ、1なら左へシフトする。シフトの結果、空いたビット位置には、左シフトの場合は0が入り、右シフトの場合は、符号ビットと同じものが入る。		
1000	8	L A I	Load address immediate	実効アドレスの下位8ビットが指定されたGRの下位8ビット(第8ビットから第15ビット)に入り、GRの上位8ビット(第0ビットから第7ビット)はすべて0になる。この命令は、GRの内容を0~255の数値に変更するために用いる。		
1010	A	A D D	Add	指定されたGRの内容と、実効アドレスで指定する語の内容とを加算し、その結果を指定されたGRにセットする。演算の結果が負ならCCに1をセットし、0か正ならCCに0をセットする。		
1011	B	S U B	Subtract	指定されたGRの内容から、実効アドレスで指定する語の内容を引いて、その結果を指定されたGRにセットする。演算の結果が負ならCCに1をセットし、0か正ならCCに0をセットする。		
1100	C	L D	Load	実効アドレスで指定する語の内容を、指定されたGRにセットする。		
1101	D	S T	Store	指定されたGRの内容を、実効アドレスで指定する語に入れる。GRの内容は変わらない。		
1110	E	A N D	And	指定されたGRの内容と、実効アドレスで指定する語の内容とのビットごとの論理積を、指定されたGRに入れる。		
1111	F	E O R	Exclusive or	指定されたGRの内容と、実効アドレスで指定する語の内容とのビットごとの排他的な論理和を、指定されたGRに入れる。		

* COMP-X は、情報処理技術者試験用に構成された仮想計算機の名前である。

によって正ならば0、負ならば1がセットされる。

(2) 記憶装置

1語16ビットとし0を含めて256の整数倍のアドレスから始まる連続した256語を1記憶ブロックとして、最小1記憶ブロックから最大250ブロックまでを実装する。

(3) 入出力装置

COMP-X では入出力命令がなく、入力直接打鍵することにより任意の記憶域またはレジスタに2進数をセットし、出力は計算機を停止した状態で任意の記憶域またはレジスタを読みとることになっているが、本システムではこれにかわる2つの擬似命令を付加した。この命令についてはソフトウェアの仕様のところで述べる。

(4) 命令の種類

命令は全部で12種類あり、これらの命令コードおよび機能は表一に示す。

2.2 ソフトウェアの仕様

COMP-X のためのアセンブラ言語をCAP-Xと呼ぶ。CAP-Xは、プログラムの始めと終わりを示すSTART, END命令、定数、領域を確保するCONST, RESV命令、そしてサブルーチンに分岐するためのADCON命令の以上5つの擬似命令があるが、このほかにCOMP-Xにおける入出力装置のかわりとして、2つの擬似命令を付加した。以下にその命令について示す。

(1) TRC命令 (|TRC| n_1 [, h_1 , n_2])

TRC命令はプログラム内の指定されたステートメントの直前の実行可能な命令の実行終了時の各レジスタおよび指定した記憶装置の内容を出力する命令である。

n_1 (check point No) : プログラム作成者が指定する

任意の10進数 ($0 \leq n_1 \leq 32767$)。

h_1 (memory dump start address) : 内容を表示したい記憶装置のアドレスで16進数4桁の数 ($0000_{16} \leq h_1 \leq FFFF_{16}$)。

n_2 (dump number) : 出力したい語数で16進4桁の数 ($0000_{16} \leq n_2 \leq C8_{16}$)。

但し、記憶装置の内容を出力しない場合は、 h_1 , n_2 は省略してもよい。

(2) SET命令 (|SET|name, h_1 [, h_2])

SET命令はこの命令が記入されたステートメントの直後の実行可能な命令の実行の直前に指定したレジスタまたは記憶装置に値をセットする命令である。

name : レジスタの種類 (BR, GR 0~3, SC, CC) 記憶装置は“MM”を記入する。

h_1, h_2 : セットすべき値で、16進4桁である。

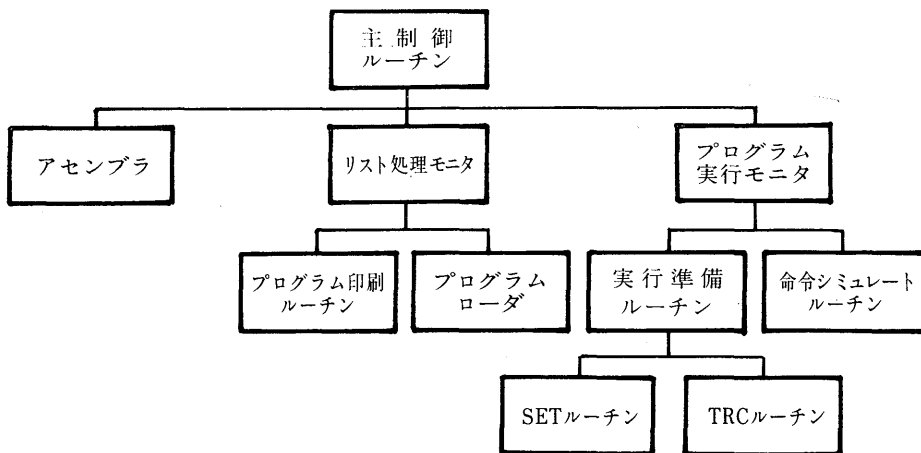
但し、nameに“MM”を指定した場合、 h_1 はアドレス、 h_2 はセットすべき値となる。

3. プログラムシステム

2.で述べた仮想計算機をアセンブラ言語教育に利用するために、FACOM 270-30でシミュレートするシステムを開発した。本システムではCAP-Xで書かれたプログラムをいったん機械語に翻訳したのち、機械語をシミュレートしている。以下システムの設計方針、構成および処理プログラムについて述べる。

3.1 システムの設計方針

本システムは、COMP-Xをシミュレートし、CAP-Xアセンブラを完全にインプリメントすることを目的とする。これにより情報処理技術者試験のアセンブラ言語の修得を容易にする。また、プログラム記憶方式や数の



図一 システム構成

表現形式を理解する場合に利用しやすい仮想計算機となるものにする。

本来 CAP-X には、入出力命令がないために、バッチ処理が行えない。これを解決するために SET, TRC 命令の入出力命令を付加し、バッチ処理を可能とする。

以上の方針にしたがって、次のような構成のシステムを開発した。

3.2 システム構成

このシステムは図-1に示すような10のモジュールから構成されており、次のような機能をもっている。

主制御ルーチン：システムの初期化および各ルーチンの制御を行う。

アセンブラ：CAP-X で書かれたプログラムの COMP-X 機械語への翻訳を行う。

リスト処理モニタ：プログラムの印刷、機械語プログラムの実行記憶装置へのロードの制御を行う。この制御のもとにプログラム印刷ルーチン、プログラムローダがある。また、プログラム内に文法エラーがある場合はプログラムリストの出力指示を行った後主制御ルーチンへ制御をもどす。

プログラム実行モニタ：命令の実行制御ルーチンのことで、実行可能な機械語プログラムについて、実行する命令の制御を行う。これは仮想計算機の制御装置にあたる。

実行準備ルーチン：SET, TRC 命令の実行場所を登録

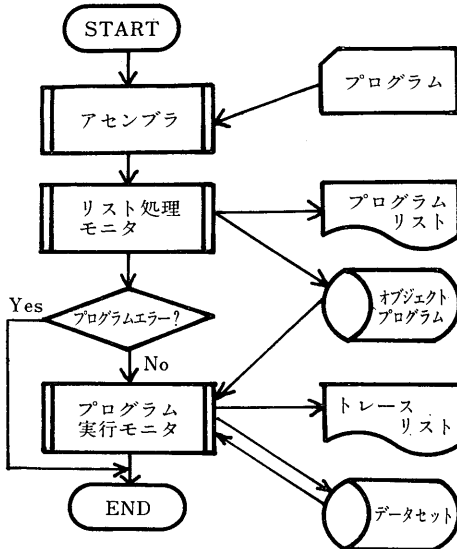


図-2 主制御ルーチン

* 情報表とは、SET, TRC 命令が擬似命令のため機械語を生成しない、そのためどのステートメントで命令の実行をするかという情報や実行回数の情報を登録する表である。

する情報表*があるが、各命令の実行に先立ちSET, TRC 命令の有無の確認を行うルーチンである。また、TRC ルーチン, SET ルーチンもこのもとで制御される。これは入出力装置にあたる。

命令シミュレートルーチン：実行記憶装置に記憶されている命令の取りだし分類を行い、その結果によって命令をシミュレートする。

3.3 処理プログラム

以上に述べた機能をもつルーチンで構成されているこのシステムは、すべて FACOM 270-30 アセンブラ言語で書かれている。これを流れ図で示せば図-2, 3, 4 のようである。

図-2はこのシステム全体を制御するためのルーチンで、この制御のもとで各ルーチンが起動される。また、

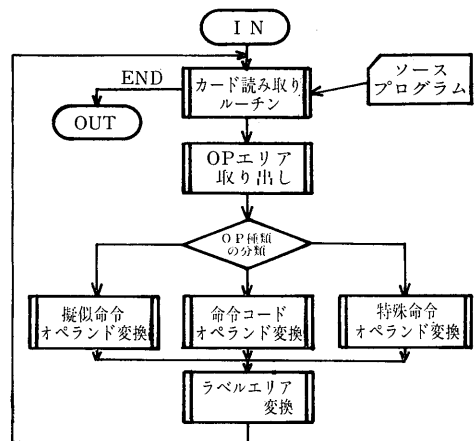


図-3 CAP-X アセンブラ

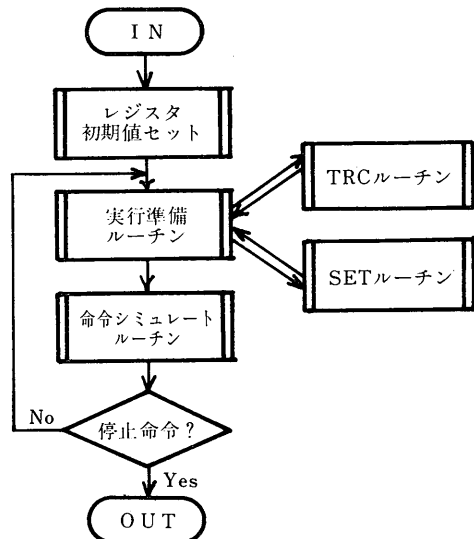


図-4 命令の実行制御ルーチン

FACOM270-30 CAP-X LIST				DATE 54/ 8/31 PAGE 1	
NO.	NAME	OP	VARIABLE=FIELD	LOC	ML
1	C-----	CAP ASSEMBLER TEST PROGRAM			
2		START	16		
3	A	RESV	1	0010	
4	B	RESV	1	0011	
5	A01	RESV	1	0012	
6	A02	RESV	1	0013	
7	A03	RESV	1	0014	
8	A04	RESV	1	0015	
9	A05	RESV	1	0016	
10	CNT	RESV	1	0017	
11	C1	CONST	0001	0018 0001	0000 0000 0000 0001
12	ENT	SFT	0.15	0019 400F	0100 0000 0000 1111
13		ST	0.A01	001A D012	1101 0000 0001 0010
14		ST	0.A02	001B D013	1101 0000 0001 0011
15		ST	0.A03	001C D014	1101 0000 0001 0100
16		ST	0.A04	001D D015	1101 0000 0001 0101
17		ST	0.A05	001E D016	1101 0000 0001 0110
18		ST	0.CN1	001F D017	1101 0000 0001 0111
19		SET	MM.0010.0001		
20		SET	MM.0011.0002		
21		LD	0.A	0020 C010	1100 0000 0001 0000
22		ADD	0.B	0021 A011	1010 0000 0001 0001
23		ST	0.A01	0022 D012	1101 0000 0001 0010
24		TRC	1.0010.0006		
25		LD	1.A	0023 C410	1100 0100 0001 0000
26		SUB	1.B	0024 B411	1011 0100 0001 0001
27		ST	1.A02	0025 D413	1101 0100 0001 0011
28		TRC	2.0010.0006		
29		SET	MM.0010.AAAA		
30		SET	MM.0011.0004		
31		LD	2.A	0026 C810	1100 1000 0001 0000
32		AND	2.B	0027 E811	1110 1000 0001 0001
33		ST	2.A03	0028 D814	1101 1000 0001 0100
34		TRC	3.0010.0006		
35		LD	3.A	0029 CC10	1100 1100 0001 0000
36		EOR	3.B	002A FC11	1111 1100 0001 0001
37		ST	3.A04	002B DC15	1101 1100 0001 0101
38		TRC	4.0010.0006		
39		SET	GR0.F+FB		
40		SFT	2.15	002C 480F	0100 1000 0000 1111
41	LUP	LD	1.C1	002D C418	1100 0100 0001 1000
42		ADD	1.CN1	002E A417	1010 0100 0001 0111
43		ST	1.CN1	002F D417	1101 0100 0001 0111
44		ADD	1.A05	0030 A416	1010 0100 0001 0110
45		ST	1.A05	0031 D416	1101 0100 0001 0110
46		ADD	0.C1	0032 A018	1010 0000 0001 1000
47		JC	1.LUP.0	0033 242D	0010 0100 0010 1101
48		TRC	5.0016.0002		
49		HJ	0.ENT		
50		END	ENT	0034 0019	0000 0000 0001 1001

A		B		C	
D		***** TRACE LIST *****			
* CHECK-POINT NO.	1 *	GR0=0003	GR1=0000	GR2=0000	GR3=0000 CC=0 BR=0000 SC=0022 * EXECUTIVE-COUNT* 1 *
E * MEMORY DUMP * 0001 0002 0003 0000 0000 0000					
* CHECK-POINT NO.	2 *	GR0=0003	GR1=FFFF	GR2=0000	GR3=0000 CC=1 BR=0000 SC=0025 * EXECUTIVE-COUNT* 1 *
* MEMORY DUMP * 0001 0002 0003 FFFF 0000 0000					
* CHECK-POINT NO.	3 *	GR0=0003	GR1=FFFF	GR2=0000	GR3=0000 CC=1 BR=0000 SC=0028 * EXECUTIVE-COUNT* 1 *
* MEMORY DUMP * AAAA 0004 0003 FFFF 0000 0000					
* CHECK-POINT NO.	4 *	GR0=0003	GR1=FFFF	GR2=0000	GR3=AAAE CC=1 BR=0000 SC=0028 * EXECUTIVE-COUNT* 1 *
* MEMORY DUMP * AAAA 0004 0003 FFFF 0000 AAAE					
* CHECK-POINT NO.	5 *	GR0=FFFF	GR1=0001	GR2=0000	GR3=AAAE CC=1 BR=0000 SC=0033 * EXECUTIVE-COUNT* 1 *
* MEMORY DUMP * 0001 0001					
* CHECK-POINT NO.	5 *	GR0=FFFF	GR1=0003	GR2=0000	GR3=AAAE CC=1 BR=0000 SC=0033 * EXECUTIVE-COUNT* 2 *
* MEMORY DUMP * 0003 0002					
* CHECK-POINT NO.	5 *	GR0=FFFF	GR1=0006	GR2=0000	GR3=AAAE CC=1 BR=0000 SC=0033 * EXECUTIVE-COUNT* 3 *
* MEMORY DUMP * 0006 0003					
* CHECK-POINT NO.	5 *	GR0=FFFF	GR1=000A	GR2=0000	GR3=AAAE CC=1 BR=0000 SC=0033 * EXECUTIVE-COUNT* 4 *
* MEMORY DUMP * 000A 0004					
* CHECK-POINT NO.	5 *	GR0=0000	GR1=000F	GR2=0000	GR3=AAAE CC=0 BR=0000 SC=0033 * EXECUTIVE-COUNT* 5 *
* MEMORY DUMP * 000F 0005					

図-5 使用例

図-3はCAP-Xで書かれたプログラムの翻訳の流れについて示し、図-4は実行時における命令の制御について示している。

本システムは、FACOM 270-30 モニタのもとで実行され、アカウントの集計処理については既存のシステムプログラムによって処理する。

4. 使用例

システムではバッチ処理で実行するためにSET, TRC命令を付加したが、実際のプログラムでどのように出力されるか図-5に示し、使用例について簡単に説明を加える。

Aの部分は、各行識別番号とともに出力されたソースプログラムリストである。Bの部分には注釈を出力する。また、Cの部分へ、割りふられたロケーションとともにオブジェクトコードの16進数と2進数の両表示で出力する。トレースリストでは、TRC命令を指定するとレジスタの状態を強制的に出力する(Dの部分)。また、Eの部分は出力の選択が可能である。Dの部分は左からチェックポイントナンバー、各レジスタ類の内容を表示する。そして右端にはこの命令の実行回数を出している。SET命令では、19行目のSET命令によってロケーション10へ“1”を代入しているが、そのロケーションの内容が24行目のTRC命令によるメモリダンプ(E

の部分)の結果で代入されたのが確認できる。

5. おわりに

はじめに述べたように本システムは、COMP-X, CAP-Xの実習用に開発したが、他の利用方法として、機械語のレベルでの計算機の処理について習得する場合、現在のように複雑化した計算機を利用するのに比べ、本システムのような簡単なアセンブラ言語を利用して教育を行う方が、効果が十分に得られると考える。また、今回は利用形態の制限からバッチ処理ベースで利用できるシステムの開発を行ったが、昭和54年10月から本学のシステムがM-160ADに変わるのを機会として、TSS処理で学生各個人が計算機と対話しながら利用できるシステムに変更する予定である。

なお、本研究は本学奨励研究費の助成によって行われている。

おわりに、日頃われわれの研究に深いご理解をいただいている本学電子計算機センター所長勝守寛教授、ならびに本稿の内容に適切なご助言をいただいた本学電子計算機センター主任水島章次助教授に感謝の意を表わす次第である。また、本研究の協力者であった上村知枝子女士に感謝する。

参 考 文 献

- 1) 昭和54年度情報処理技術者試験案内書