情報工学実験２
コンピュータアーキテクチャと命令セット

035743A：比嘉雅樹

共同実験者： 035740F:根保光秀
実験日： 2004/11/01
提出日： 2004/11/08

1　実験の目的

KUE-CHIP2 を用いて例題プログラムを実行することにより、KUE-CHIP2 の操作方法を習得する。また、簡単な機械語プログラムを作成し、機械語プログラムの構造を理解する。

2　報告事項

2.1　各実験について結果を報告しなさい

2.1.1　実験書第３章 3.1(p17) の例題プログラムを C 言語に書き換えよ

表 1: プログラム

ARDS	DATA	OPECODE		
00	75 03	ST	ACC,	(03H)
02	C0	EOR	ACC,	ACC
03	B5 03	ADD	ACC,	(03H)
05	AA 01	SUB	IX,	1
07	31 03	BNZ	03H	
09	0F	HLT		

説明

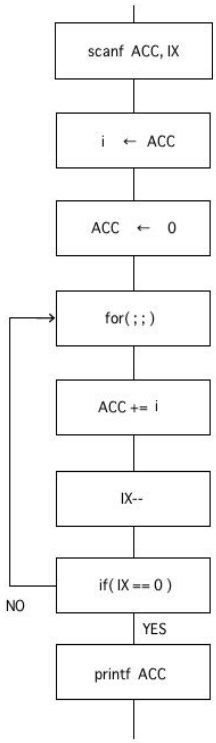
ST ACC, (03H) ⋯ ACC の値を 03H 番地に格納

EOR ACC, ACC ⋯ ACC の初期化

ADD ACC, (03H) ⋯ ACC と 03H 番地の値を足して ACC に入れる

SUB IX, 1 ⋯ IX の値から 1 引く

BNZ 03H ⋯ 直前の命令の値が 0 でないならば 03H の命令にとぶ




```
#include <stdio.h>

int main() {
  int ACC, IX, i;

  printf("Input ACC：");
  scanf("%d", &ACC);
  printf("Input IX：");
  scanf("%d", &IX);

  i = ACC;
  ACC = ACC ^ ACC;
  for ( ; ; ) {
    ACC += i;
    IX--;
    if ( IX == 0 )
      break;
  }
  printf("ACC = %d\n", ACC);
}
```

図 1: プログラムの流れ

2.1.2 以下の各項目について説明せよ。

(a) ACC の内容を観測するには、どのように操作すればよいか
SEL の値を 4 にすることで観測できる。

(b) IX の内容を観測するには、どのように操作すればよいか
SEL の値を 5 にすることで観測できる。

(c) PC の内容を観測するには、どのように操作すればよいか
SEL の値を 2 にすることで観測できる。

(d) ACC に 0x10 をセットするには、どのように操作すればよいか
SEL の値を 4、DATAsw を 01H にして SET を押す。

(e) IX に 0x10 をセットするには、どのように操作すればよいか
SEL の値を 5、DATAsw を 01H にして SET を押す。

(f) PC に 0x10 をセットするには、どのように操作すればよいか
SEL の値を 2、DATAsw を 01H にして SET を押す。

(g) 0x80 番地の内容を見るには、どのように操作すればよいか
ADDRESSsw を 80H(010000000) にして、IMC を CHECK に入れる。

(h) 0x80 番地に 0x34 をセットするには、どのように操作すればよいか
項目 (g) の状態で DATAsw を 34H にして SET を押す。

(i) 0x00 番地から 0x0F 番地までの内容を連続的に見るにはどのように操作すればよいか
ADRINC を連続して押す。

(j) 0x00 番地から 0x0F 番地までの内容を全て 0x11 にセットするにはどのように操作すればよいか
DATAsw を 11H にしたまま SET と ADRINC を交互に連続して押す。

(k) 0x90 番地から 0x9F 番地までの内容を連続的にみるにはどのように操作すればよいか
項目 (g) の状態から ADRINC を連続して押す。

(l) 0x90 番地から 0x9F 番地までの内容を全て 0x11 にセットするにはどのようにすればよいか
項目 (g) の状態から DATAsw を 11H にして SET と ADRINC を交互に連続して押す。

(m) プログラムを 0x00 番地から実行するにはどのように操作すればよいか
SSsw を押す。

(n) プログラムを 0x20 番地から実行するにはどのように操作すればよいか
PC を 20 にセットして SSsw を押す。

(o) プログラムを 1 命令ずつ実行するにはどのように操作すればよいか
SPsw を押す。

(p) プログラムを 1 フェーズずつ実行するにはどのように操作すればよいか
SIsw を押す。

2.1.3 実験書第４章 (p30,31) の例題プログラムを C 言語に書き換えなさい

表 2: １から N までの和を出すプログラム

ARDS	DATA	OPECODE			
00	6C 80	LD	IX,	[N]	
02	62 00	LD	ACC,	0	
04	B1	LOOP:	ADD,	ACC,	IX
05	AA 01	SUB	IX,	1	
07	33 04	BP	LOOP		
09	74 81	ST	ACC,	[SUM]	
0B	0F	HLT			
80		N:	EQU	80H	
81		SUM:	EQU	81H	

説明
最初に 80H に N の値を入れておく

LD IX, [N] ⋯ N の値を IX に読み込む

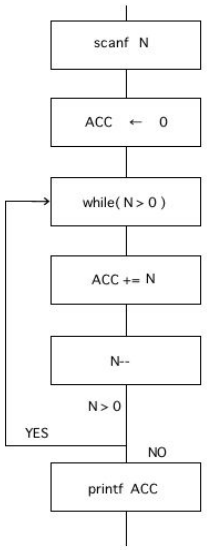
LD ACC, 0 ⋯ ACC の初期化

LOOP: ADD ACC, IX ⋯ ACC と IX の値を足して ACC に入れる。また、ループの為の目印をつける

SUB IX, 1 ⋯ IX の値から１引く

BP LOOP ⋯ 直前の命令の値が０より大きければ LOOP で示す場所へとぶ

ST ACC, [SUM] ⋯ ACC の値を 81H に格納




```
#include <stdio.h>

int main() {
  int ACC, N;

  printf("Input N : ");
  scanf("%d", &N);

  ACC = 0;
  while( N > 0 ) {
    ACC += N;
    N--;
  }
  printf("%d\n", ACC);
}
```

図 2: プログラムの流れ

2.1.4 学籍番号の６個の数を足してその合計を求めるプログラムを作成し、実行しなさい。ただし、６個の数はデータ領域の 0x00〜0x05 番地に予め格納しておくこと。また、このプログラムを C 言語に書き換えなさい。

表 3: プログラム

ARDS	DATA	OPECODE		
00	C0	EOR	ACC,	ACC
01	B5 00	ADD	ACC,	(00H)
03	B5 01	ADD	ACC,	(01H)
05	B5 02	ADD	ACC,	(02H)
07	B5 03	ADD	ACC,	(03H)
09	B5 04	ADD	ACC,	(04H)
0B	B5 05	ADD	ACC,	(05H)
0D	0F	HLT		

下記に C で書いたプログラムを示す。

```
#include <stdio.h>

int main() {
  int num[] = { 0, 3, 5, 7, 4, 3 };
  int ACC;

  ACC += num[0];
  ACC += num[1];
  ACC += num[2];
  ACC += num[3];
  ACC += num[4];
  ACC += num[5];

  printf("%d\n", ACC);
}
```

2.1.5 C とアセンブリ言語の違いや特徴

- アセンブリ言語では、ACC や IX 等が固有の変数として最初から用意されているが、C ではプログラム中で宣言しなければならない。
- アセンブリ言語では、プログラムカウンタやアドレスレジスタ等の値に注意する必要があるが、C では気にせずにプログラムが書ける。
- アセンブリ言語では、プログラムの終了を HLT を使って指定する必要があるが、C では記述しなくてもよい。
- アセンブリ言語の命令は、機械語の命令と 1 対 1 に対応しているので使用する記憶場所や実行速度を考えながらプログラムが書ける。
- アセンブリ言語はハードウェア固有の言語なので、CPU が異なると機械語（文法）も変わる
- アセンブリ言語は機会語の命令と 1 対 1 で対応していて、汎用性が重視されているのでプログラムの作成が面倒。

2.2 コンピュータの主要構成要素について、以下の設問に答えよ

2.2.1 入力装置

- 入力装置の役割を簡単に説明せよ
  - コンピュータに処理させるデータやプログラム、操作などを入力するための装置。
- 入力装置の具体例を３つ以上挙げ、それぞれの特徴を簡単に説明せよ
  - キーボード
    最も代表的な入力装置で、キーを押す事で対応した文字コードや制御情報を入力する。
  - タッチパネル
    指で操作するので精度が低く細かい指示はできないが、簡単な操作ですむため ATM 等広く利用されている。
  - ライトペン
    ペン状の入力装置で、先端にある光センサで画面上の情報を読み取る。
- コンピュータを１台選択し、そのコンピュータに内蔵あるいは外付けされている入力装置をすべて列挙せよ
  選択したコンピュータ：ibook G4
  - キーボード
  - トラックパッド

2.2.2 記憶装置

- 記憶装置の役割を簡単に説明せよ
  - 入力されたデータやプログラムを記憶する装置。
- 補助記憶装置の例を５つ以上挙げ、それぞれの特徴や違いなどを簡単に説明せよ
  - ハードディスク
    磁気ディスク装置の一つ、容量は様々でファイルにはランダムアクセス方式でアクセスする。

- フロッピーディスク
  磁気ディスク装置で、磁性体を塗布した一枚の円盤とそれを防護するジャケットで構成される。容量が小さく、データの読み書きの速度も速くないが、安価である。
- CD-R
  光ディスク装置で、一度だけデータの書き込みができる。容量は主に 700MB で、一度データを書き込むと、追加・消去ができなくなる。データの追加、消去ができる CD-RW もある。
- DVD-R
  光ディスク装置で、一度だけデータの書き込みができる。CD と違い、4.7GB、8.5GB、9.4GB と大容量である。データの追加、消去ができる DVD-RW もある。
- MO
  光磁気ディスク装置で、書き換え可能な記憶装置。容量は 128MB、230MB、540MB、640MB のものが一般的。

- 2.2.1 で選択したコンピュータに内蔵あるいは外付けされている記憶装置をすべて列挙し、それぞれの容量を調べよ。

  - メモリ：768MB
  - ハードディスク：約 28GB

2.2.3 出力装置

- 出力装置の役割を簡単に説明せよ

  コンピュータで処理した結果や途中経過を出力するための装置。

- 出力装置の具体例を 3 つ以上挙げ、それぞれの特徴を簡単に説明せよ

  - ディスプレイ
    ブラウン管や液晶の画面上に図形や文字を表示させる装置。
  - プリンタ
    コンピュータの処理結果を紙に印刷する装置。
  - スピーカー
    音声を出力する装置。

- 2.2.1 で選択したコンピュータに内蔵あるいは外付けされている出力装置をすべて列挙せよ

  - 液晶ディスプレイ
  - スピーカー

2.2.4 CPU = 演算装置 + 制御装置

- 演算装置の役割を簡単に説明せよ
  主記憶装置から送られてくるデータについて、四則演算や比較判断などを半導体による論理素子によって高速に行う装置。

- 制御装置の役割を簡単に説明せよ
  主記憶装置に記憶されているプログラムの命令を一つずつ取り出して読解し、核装置に指令を与える装置。

- 2.2.1 で選択したコンピュータに搭載されている CPU を挙げ、その性能や特徴について報告せよ
  - プロセッサ：1.07GHz PowerPC G4
  - 二次キャッシュ：512KB

2.3 以下に示す各レジスタの役割を説明せよ。また、KUE-CHIP2 に無いものを挙げよ

(a) アキュムレータ
演算結果を置いたり、データを一時的に保存するレジスタ。コンピュータには通常１つ以上のアキュムレータ（または同等の機能を持つレジスタ）がある。

(b) インデックスレジスタ
外部メモリのアドレスのオフセットや、アドレスそのものを格納するレジスタ。

(c) プログラムカウンタ
次の命令があるアドレスを保存しておくレジスタ。

(d) メモリアドレスレジスタ
データの読み書きをするときに、記憶装置の番地を指定するレジスタ。

(e) 命令レジスタ
実行する命令を一時的に蓄えるレジスタ。

(f) フラグレジスタ
演算後の情報を入れるレジスタ。

(g) スタックポインタ
処理途中のデータを一時的に積み上げておき、そのアドレスを記憶しておくレジスタ。

(h) 汎用レジスタ
　　アキュムレータ、インデックスレジスタなどに自由に使えるレジスタ。

上記のレジスタの中で、KUE-CHIP2 に無いものは、スタックポインタである。

2.4 KUE-CHIP2 の命令を、分類し、各分類毎にアセンブリ言語と機械語の対応表を書け。また、各命令の機能を簡単に説明せよ。

表 4: データ転送命令

LD	60　6F	ACC や IX に値を読み込む
ST	74　77, 7C　7F	ACC や IX に値を保存する

表 5: 演算命令

SBC	80　8F	減算命令、桁上げフラグ CF を考慮する
ADC	90　9F	加算命令、桁上げフラグ FC を考慮する
SUB	A0　AF	減算命令、桁上げフラグ FC を考慮しない
ADD	B0　BF	加算命令、桁上げフラグ FC を考慮しない
EOR	C0　CF	ビット毎の排他的論理和演算命令
OR	D0　DF	ビット毎の論理和演算命令
AND	E0　EF	ビット毎の論理積演算命令
CMP	F0　FF	比較命令
SRA	40,48	算術右シフト命令
SLA	41,49	算術左シフト命令
SRL	42,4A	論理右シフト命令
SLL	43,4B	論理左シフト命令
RRA	44,4C	算術右巡回シフト命令
RLA	45,4D	算術左巡回シフト命令
RRL	46,4E	論理右巡回シフト命令
RLL	47,4F	論理左巡回シフト命令

表 6: 制御命令

NOP	00	何もしない
HLT	0F	停止命令
RCF	20	桁上げフラグ CF をリセットする
SCF	2F	桁上げフラグ CF をセットする

表 7: 入出力命令

IN	1F	入力命令、IBUF の内容を ACC へ移す
OUT	10	出力命令、ACC の内容を OBUF へ移す

表 8: 特殊命令

BA	30	常に成立
BVF	38	VF = 1
BNZ	31	ZF = 1
BZP	32	NF = 0
BP	33	NF = 0 または ZF = 0
BNI	34	IBUF = 0
BNC	35	CF = 0
BGE	36	VF と NF が等しい
BGT	37	VF と NF が等しい、または ZF = 0
BZ	39	ZF = 1
BN	3A	NF = 1
BZN	3B	NF = 1 または ZF = 1
BNO	3C	OBUF = 0
BC	3D	CF = 1
BLT	3E	VF と NF が異なる
BLE	3F	VF と NF が異なる、または ZF = 1

2.5　本実験について考察せよ

今回の実験では、KUE-CHIP2 の操作方法を習得し、簡単な機械語のプログラムの構造を理解した。
次に、アセンブリ言語について調べてみた。

- 機械語とアセンブリ言語
  機械語とは、コンピュータが直接理解できる言語であり０と１の組み合わせでできている。この機械語を英字や数字等の記号を使って、人にわかりやすくした言語の一つがアセンブリ言語である。

- 逆アセンブラ
  マイクロプロセッサのマシン語コードから、それに対応するニーモニック文字列に変換するプログラム。マイクロプロセッサが解釈できるマシン語コードは、単なる数値であり、人間にとっては読みにくいので、通常はアセンブリ言語と呼ばれるニーモニックを作成し、アセンブラというプログラムでこれをマシン語コードに変換し、実行プログラムにする。逆アセンブラは、ちょうどこのアセンブラとは逆の操作を行うプログラムである。逆アセンブラを使えば、アセンブリ言語のソースコードがなくても、それに相当するものをプログラムで生成することができる。しかしソースコード中の変数名などはアセンブルの過程で失われるので、元のソースコードが完全に再現されるわけではない。

参考文献

[1] サクセスガイド　ハードウェア　著：安藤明之

[2] http://www.infonet.co.jp/ueyama/ip/glossary/ax_storage.html

[3] http://www.infonet.co.jp/ueyama/ip/glossary/harddisk.html

[4] http://ew.hitachi-system.co.jp/w/CD-R.html

[5] http://ew.hitachi-system.co.jp/w/DVD.html

[6] http://ew.hitachi-system.co.jp/w/MO.html

[7] http://ja.wikipedia.org/wiki/レジスタ_(CPU)

[8] http://www.kecl.ntt.co.jp/car/parthe/html/lecture/95/kadai.htm

[9] http://aitech.ac.jp/~koikelab/webp/hard/kue-chip2-hosoku.html

[10] http://www005.upp.so-net.ne.jp/h-masuda/ProText/casl2/casl2001.html

[11] http://www.atmarkit.co.jp/icd/root/72/5784272.html