情報工学実験2

「アセンブラプログラミング」

実施日：2010年11月8日

学籍番号：　０９５７０７Ｂ
氏　　名：　大城佳明
締切日：　2010年11月15日
共同実験者：　095701B　青木史林
095703J　岩瀬　翔

1　実験目的

アセンブラプログラムの作成、ハンドアセンブル (アセンブラプログラムの人手で機械語プログラムに直すこと)、実行の各作業を実際に行うことにより、アセンブラプログラミングの流れを習得する。また、コンパイラとアセンブラの違いや、高水準言語とアセンブリ言語の違いについて理解することを目的とする

2　実験概要

D-A コンバータをブレットボード上に実現し、D-A コンバータの使い方を思い出す。さらにそれを KUE-CHIP2 の出力ポートに接続する。のこぎり波を出力するプログラムの例を KUE-CHIP2 に入力する。入力したプログラムにより、のこぎり波がオシロスコープで確認できるようにする。そうすることで、プログラムの流れとオシロスコープの出力の関係を知ることが出来る。矩形波、山形波、菱形波を出力するアセンブラプログラムを作成し、KUE-CHIPP2 上に実現する。そうすることで、アセンブリ言語の理解と KUE-CHIP2 の使い方を学ぶことができる。作成したアセンブラプログラムをオシロスコープで確認する。オシロスコープの使い方も学ぶことも出来る。

3　実験結果

3.1　実験 (1)

図 2.2 の D-A コンバータをブレットボード上に実現し、KUE-CHIP2 の出力ポートに接続せよ。また、リスト 2.1 のプログラムを KUE-CHIP に入力し、D-A コンバータのアナログ出力端子から図 2.11 に示したようなのこぎり波が出力されることを、オシロスコープを用いて確認せよ。

本実験結果に関しては、報告を省略する。

3.2 実験 (2)

各波形を出力するアセンブラプログラムとその実行結果 (オシロスコープに表示された波形) を示せ。なお、アセンブラプログラムには、必ず機械語 (マシン語) プログラムを併記すること。また、必要に応じて、各プログラムを実行する前のレジスタやメモリの初期値も明記せよ。
各アセンブラプログラミングがどのような動作をするプログラムなのか説明し、実行結果の正当性を示せ。

3.2.1 (a) 矩形波

1. 矩形波のプログラムを表 1 に示した。

表 1: 矩形波 (プログラム)

番地	機械語	アセンブラ言語
00	C0	EOR ACC,ACC
01	C9	EOR IX,IX
02	B2	ADD ACC,FE(H)
03	FE	
04	BA	ADD IX 80(H)
05	80	
06	10	OUT
07	AA	SUB IX 01(H)
08	01	
09	31	BNZ 06(H)
0A	6	
0B	A2	SUB ACC FEH
0C	FE	
0D	BA	ADD IX 80(H)
0E	80	
0F	10	OUT
10	AA	SUB IX 01(H)
11	01	
12	31	BNZ 0F(H)
13	0F	
14	30	BA 02(H)
15	02	

2. プログラムの説明

   (a) 00,01 番地は ACC と IX の初期化を行っている

   (b) 02,03 番地は ACC に 254 を入れた。オシロスコープでは上 (高い位置) から始めるようにする

   (c) 04,05 番地は IX に 128 を入れる。

   (d) 06 番地で今の ACC の値を出力する

   (e) 07,08 番地では IX をデクリメントしている

   (f) 09,0A 番地は IX が 0 でなければ、06 番地に JUNP する。そうすることで、オシロスコープでは一定時間の間、上 (高い位置) を保つことができる。

(g) 0B,0C 番地は ACC の値を 128 引いている。オシロスコープでは下 (低い位置) に移動することになる。

(h) 0D〜13 番地までは 04〜0A 番地と同じ動作をしている

(i) 14 番地は 02 に JUMP する。そうすることで、また上 (高い位置) に戻り、繰り返すことができる。

3. 実行結果を図 1 に示した。

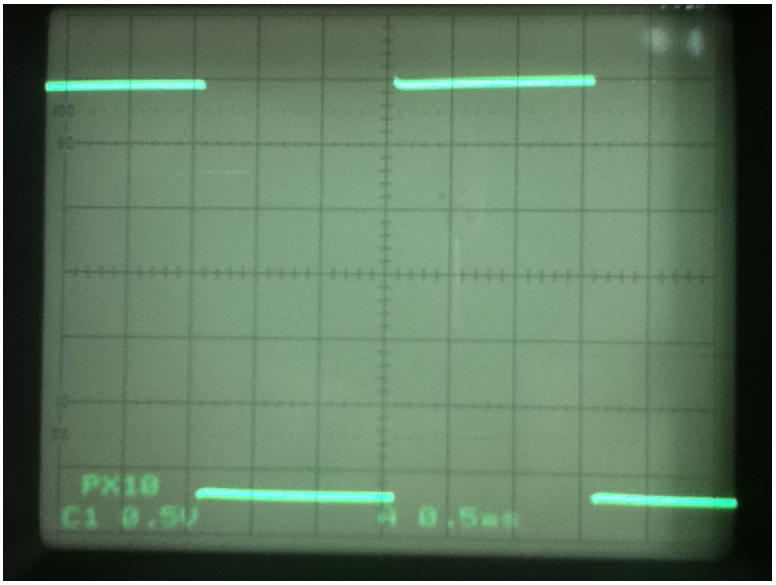

図 1: 矩形波 (オシロスコープ)

3.2.2 (b) 山形波

1. 山形波のプログラムを表 2 に示した。

2. プログラムの説明

(a) 00,01 番地は ACC と IX の初期化を行っている

(b) 02,03 番地は IX に 255 を入れた。

(c) 04 番地は ACC の値を出力している

(d) 05,06 番地は ACC の値をインクリメントしている。そうすることで、山形の点がだんだんが上昇しているようにみえる。

(e) 07,08 番地は IX の値をデクリメントしている。255 回デクリメントすることになる

(f) 09,0A 番地は IX の値が 0 でなければ 04 番地に JUMP する

(g) 0B 番地は ACC の値を出力する。このとき ACC の値は 255 である

(h) 0C,0D 番地は ACC の値をデクリメントしている。そうすることで、山形の点がだんだん減少しているようにみえる

(i) 0E,0F 番地は ACC の値が 0 でなければ 0B 番地に JUMP する

(j) 10,11 番地は ACC の値が 0 ならば 00 に JUMP する。JUMP することにより、山形を繰り返すことができる

3. 実行結果を図 2 に示した。

表 2: 山形波 (プログラム)

番地	機械語	アセンブラ言語
00	C0	EOR ACC,ACC
01	C9	EOR IX,IX
02	BA	ADD IX,FF(H)
03	FF(H)	
04	10	OUT
05	B2	ADD ACC,01(H)
06	01(H)	
07	AA	SUB IX,01(H)
08	01(H)	
09	31	BNZ 04(H)
0A	04(H)	
0B	10	OUT
0C	A2	SUB ACC,01(H)
0D	01(H)	
0E	31	BNZ 0B(H)
0F	0B(H)	
10	39	BZ 00(H)
11	00(H)	

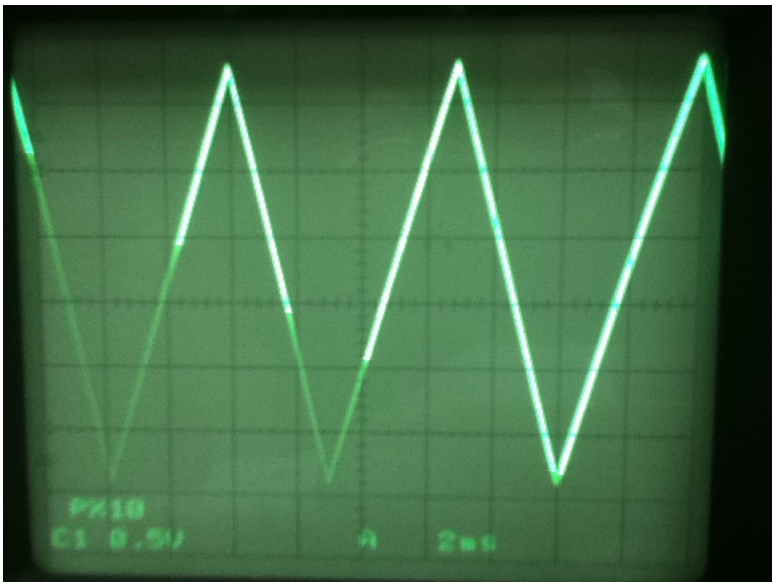

図 2: 山形波 (オシロスコープ)

3.2.3 (c) 菱形波

1. 菱形波のプログラムを表 3 に示した。

表 3: 菱形波 (プログラム)

番地	機械語	アセンブラ言語
00	C0	EOR ACC,ACC
01	B2	ADD ACC,FE(H)
02	FE	
03	10	OUT
04	C2	EOR ACC,FF(H)
05	FF	
06	10	OUT
07	C2	EOR ACC,FF(H)
08	FF	
09	A2	SUB ACC,01(H)
0A	01	
0B	31	BNZ 03(H)
0C	03	
0D	30	BA 01(H)
0E	01	

2. プログラムの説明

   (a) 00 番地は ACC の初期化を行っている

   (b) 01,02 番地は ACC に 254 を入れている。オシロスコープでは上 (高い位置) から始まるようにしている。

   (c) 03 番地は ACC の値を出力している

   (d) 04,05 番地は ACC の値を EOR を使って反転させている。

   (e) 06 番地は ACC の値を出力している

   (f) 07,08 番地は ACC の値を EOR で元に戻している

   (g) 09,0A 番地は ACC の値をデクリメントしている。そうすることで、菱形の 1 本の線の点がだんだん減少しているようにみえる

   (h) 0B,0C 番地は ACC の値が 0 でなければ 03 番地に JUMP する

   (i) 0D,0E 番地は 01 番地に JUMP する。そうすることにより、「XXX」のように繰り返されて、菱形ができる

3. 実行結果を図 3 に示した。

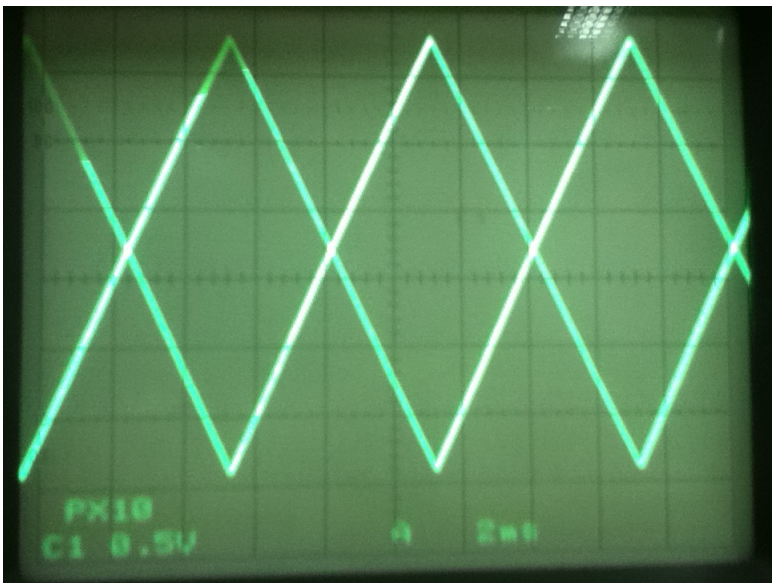

図 3: 菱形波 (オシロスコープ)

4 考察

4.1 実験 (2) の考察について

今回の方法によって出力不可能な波形について考察せよ。

1. 2 本以上の形の違う線

   (a) 今回の方法では 1 本しか出力されないので 2 本以上の違う線の出力はできないと考えられる。

   (b) 例を図 4 に示した。

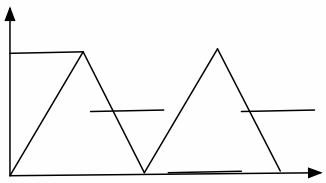

図 4: 2 本以上の線

2. 間隔のあいた線

   (a) 出力されているのは ACC の値なので値がないという状態がないと考えられる。

   (b) 値がない状態がないと言うことは間隔のあいた線が書けないと考えられる。

   (c) 例を図 5 に示した。

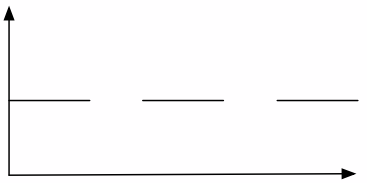

図 5: 間隔のあいた線

4.2 その他の考察について

実験で使用したプログラムとは違うプログラムを考える

1. 菱形波プログラムを使って考える。
2. ADD の変わりに LD を使うことが可能である。
3. FE(H) ではなく FF(H) を使った方が良い。

表 4: 菱形波 (プログラム)

番地	機械語	アセンブラ言語
00	C0	EOR ACC,ACC
01	62	LD ACC,FF(H)
02	FE	
03	10	OUT
04	C2	EOR ACC,FF(H)
05	FF	
06	10	OUT
07	C2	EOR ACC,FF(H)
08	FF	
09	A2	SUB ACC,01(H)
0A	01	
0B	31	BNZ 03(H)
0C	03	
0D	30	BA 01(H)
0E	01	

4. 以上のプログラム (表 4) が完成した。
5. 表 3 よりもわかりやすいプログラムが出来たのはないかと思われる

5 調査課題

5.1 (a) プログラムは、そのプログラム言語の規則に従って記述されている。この段階のプログラムをソースプログラム (原始プログラム) と呼び、ソースプログラムを処理するためのソフトウェアを言語プロセッサ (言語処理) と呼ぶ、アセンブラは、言語プロセッサの一種であり、アセンブラプログラムを機械語プログラムに変換するソフトウェアである。アセンブラ以外の言語プロセッサとしは、コンパイラとインタプリタが挙げられる。これらの３種類の言語において行われる処理の特徴や違いについて詳しく説明し、それぞれのソフトウェアで処理される代表的なプログラミング言語を挙げよ

1. アセンブラ
   (a) 機械語に近いアセンブラ言語を機械語に変換する
   (b) 機械語に変換されると最速レベルで実行される
   (c) 変換後のプログラムのサイズも小さい
   (d) エラーが出た時のの対応が遅くなる
   (e) 例：アセンブリ言語
2. コンパイラ
   (a) 機械語に翻訳した後、実行形式のオブジェクトを作成する
   (b) 一気に機械語に変換されますが、共通な作業をするランタイム・ルーチンなどが付加される
   (c) 人間に読みやすいプログラム
   (d) エラーが出た時のの対応が遅くなる
   (e) 例：C 言語、Java
3. インタプリンタ
   (a) 機械語に翻訳しながら実行する
   (b) 一行一行機械語に翻訳しながらプログラムを実行する
   (c) 処理のスピードはアセンブラに比べて１０００倍ぐらい遅くなる
   (d) 人間に読みやすいプログラム
   (e) プログラミングのミスがあった場合でも容易に中断し、手直しできる
   (f) Python、Perl、Ruby

5.2 (b) 次回の実験で詳しく調べるが、１つの機械語命令は、いくつかのフェーズに分けられて実行される。これはどのようなプロセッサ (CPU) に対しても共通に言えることである。このことを利用して処理能力を向上させるアーキテクチャの１つにパイプライン・アーキテクチャがある。パイプライン・アーキテクチャとはどのようなアーキテクチャか調査し、図表などを用いて分かりやすく説明せよ。

1. 一定数の命令がプロセッサ内で同時に動作できるようにすること
   (a) 実行やそのほかの段階では命令でコードの回路は待機状態になる。

(b) 命令を行っている間は、演算器は休んでいる

(c) 演算を行っている間は、命令を行う部分は休んでいる

(d) 休んでいるのがもったいないので、常に動かそうという考え方

2. 最初の命令が終わり、演算を行っている間に、次の命令をしようというように流れ作業

3. 動きの説明をする (例：３つの命令がある場合)

(a) ADD を３回行うプログラムを作成し、表 5 に示した。

表 5: ADD ３つのプログラム

番地	機械語	アセンブラ言語
00	ADD	ACC,01
01	ADD	ACC,02
02	ADD	ACC,03

(b) まず、命令１は 00 番地を読み取る。(図 6)

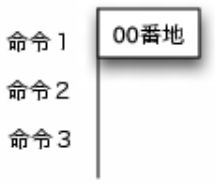

図 6: ADD-01

(c) 次に命令１は命令を読み取り、命令２は 01 番地を読み取る (図 7)

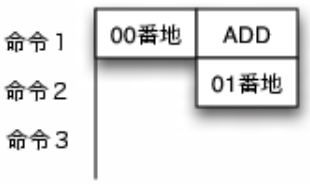

図 7: ADD-02

(d) 次に命令１は演算をし、命令２は命令を読み取り、命令３は 02 番地を読み取る (図 8)

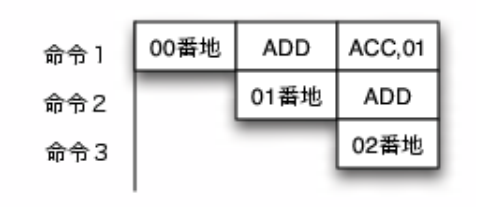

図 8: ADD-03

(e) 次に命令１は終了し、命令２は演算し、命令３は命令を読み取る (図 9)

図 9: ADD-04

(f) 命令２は終了し、命令３は演算する (図 10)

図 10: ADD-05

4. このように処理を行えば倍以上に早くなる

6 感想

1. 実験について
   今回の実験もとても楽しかったです。実機に触り、実際にプログラムを作成したのはとても良かったと思います。さらに実際に動かすことで、視覚的に結果が表れたのはとても新鮮でした。プログラムの作成に時間がかかりましたが、自分の力でプログラムを書くことが出来たの良かったです。今回の実験は特に悪いところは見当たりませんでした。スムーズに実験を進めることが出来ていたのとてもいい実験だったと思います。

2. 報告書について
   今回の報告書はとても楽でした。今までの実験の中で一番早く書き終えることが出来たと思います。その点ではとても良かったと思います。ですが、逆に簡単すぎたのではないかとも思います。調査課題の難易度が今までと比べ物にならないほど簡単でした。実験以外の他の課題もあるので、負担が軽くなるのは嬉しいことです。しかし、これでいいのかという疑問もあります。この件に関しては良いとも言えるし、悪いとも言えるので、何とも言えません。報告書自体は楽しく書けたので良かったと思います。

参考文献

wikipedia
http://ja.wikipedia.org/wiki/

コンパイラとインタプリタとアセンブラの違いって何？
http://oshiete.goo.ne.jp/qa/2540277.html

コンピュータアーキテクチャの話 (124)
http://journal.mycom.co.jp/column/architecture/124/index.html