情報工学実験実習

第 1 回

マイクロコンピュータ応用

実験年月日 2018 年 11 月 12 日（月）
提出年月日 2018 年 12 月 3 日（月）

班番号 6
報告者 3 年 19 番 6 班 末田 貴一
共同実験者

7 番 川上 求
42 番 山崎 敦史
47 番 ロンサン

2018/11/12 マイコン 1

issue

- [x] 課題 5 のレイアウト崩れ
- [x] 表紙のミス

目的

今回は，前学期のマイクロコンピュータ基礎で行ったアセンブリ言語の復習を行います。
今回復習するのは，データの転送，ループ，条件分岐，LED の制御です。

使用装置

MT-Z を用いる

実験

課題 1

8500H の数値に 5 を足した数値を，8501H に書き込むプログラムを作りなさい。

プログラムを図 1.1 に示す。

アドレス	機械語	ニーモニック	コメント
8400		ORG 8400H	
8400	3A 00 85	LD A, (8500H)	8500H の値を A レジスタに転送
8403	C6 5	ADD A, 05H	A レジスタの値に 5 を足す
8405	32 01 85	LD (8501H), A	A レジスタの値を 8501H に転送
8408	C3 00 00	JP 0000H	モニタプログラムにジャンプ
8500		ORG 8500H	
8500	1	DB 01H	01H を入れておく
8501	0	END	仮に 0H を入れておく

図 1.1 課題 1

課題 2

8500H と 8501H の数値を足した数値を 8502H に書き込むプログラムを作りなさい。

プログラムを図 1.2 に示す。

アドレス	機械語	ニーモニック	コメント
		ORG 8400H	
8400	3A 00 85	LD A, (8500H)	8500H の内容を A レジスタに転送
8403	47	LD B, A	A レジスタの内容を B レジスタに転送
8404	3A 01 85	LD A, (8501H)	8501H の内容を A レジスタに転送
8407	80	ADD A, B	A に B を加える
8408	32 02 85	LD (8502H), A	A レジスタの内容を 8502H に転送
840b	C3 00 00	JP 0000H	モニタプログラムにジャンプ

アドレス	機械語	ニーモニック	コメント
8500		ORG 8500H	
8500	1	DB 01H	01H を入れておく
8501	2	DB 02H	02H を入れておく
8502	0	DB 00H	仮に 0 を入れておく

図 1.2 課題 2

課題 3

8500H と 8501H の数値を比較し，大きい方を 8502H に書き込むプログラムを作りなさい

- データを比較するときには `CP` を使います。
  比較した結果はフラグレジスタに保存されます。
- プログラムの任意の場所にジャンプするときは `JP` を用います。
  `JP 番地`で無条件に任意の番地にジャンプし，`JP X 番地`でフラグレジスタが `X` の時に任意の番地にジャンプします。

プログラムを図 1.3 に示す。

アドレス	機械語	ラベル	ニーモニック	コメント
8400			ORG 8400H	
8400	3A 00 85		LD A, (8500H)	8500H の値を A レジスタに転送
8403	47		LD B, A	A レジスタの値を B レジスタに転送
8404	3A 01 85		LD A, (8501H)	8501H の値を A レジスタに転送
8407	B8		CP B	B と比較
8408	F2 12 84		JP P, MORE	フラグレジスタが正ならば MORE にジャンプ
840b	78		LD A, B	B レジスタの値を A レジスタに転送
840c	32 02 85		LD (8502H), A	A レジスタの値を 8502H に転送
840f	C3 00 00		JP 0000H	モニタプログラムにジャンプ

アドレス	機械語	ラベル	ニーモニック	コメント
8412	32 02 85	MORE:	LD (8502H), A	A レジスタの値を 8502H に転送
8415	C3 00 00		JP 0000H	モニタプログラムにジャンプ
8500			ORG 8500H	
8500	1		DB 01H	01H を入力しておく
8501	3		DB 03H	03H を入力しておく
8502	0		DB 00H	仮に 00H を入力しておく

図 1.3 課題 3

課題 4

10 から 1 ずつ引いていき，5 以下になった場合終了するプログラムを作りなさい

プログラムを図 1.4 に示す。

アドレス	機械語	ラベル	ニーモニック	コメント
8400			ORG 8400H	
8400	3A 00 85		LD A, (8500H)	8500H の値を A レジスタに転送
8403	3D	LOOP:	DEC A	A から 1 を引く
8404	FE 05		CP 05H	05H と比較する
8406	C2 02 84		JP NZ, LOOP	フラグレジスタが NZ ならば LOOP にジャンプ
8409	32 00 85		LD (8500H), A	A レジスタの値を 8500H に転送
840C	C3 00 00		JP 0000H	モニタプログラムにジャンプ
8412			END	
8500			ORG 8500	
8500	0A		DB 0AH	0AH(10 進数で 10)を入力しておく

図 1.4 課題 4

課題 5

図 1.2 を参考に 1-10 の和をメモリの 8500H 番地に書き込むプログラムを作りなさい

プログラムを図 1.5 に示す。

アドレス	機械語	ラベル	ニーモニック	コメント
8400			ORG 8400H	
8400	3E 00		LD A, 0	0H を A レジスタに転送する
8402	06 0A		LD B, 0A	0AH を B レジスタに転送する
8404	80	LOOP:	ADD A, B	A に B を加える
8405	5		DEC B	B から 1 を引く
8406	C2 04 84		JP NZ, LOOP:	フラグが 0 じゃないならば LOOP にジャンプ
8409	32 00 85		LD (8500H), A	A を 8500 にロード

アドレス	機械語	ラベル	ニーモニック	コメント
840C	C3 00 00		JP 0000H	モニタプログラムにジャンプ
			END	
			ORG 8500H	
8500	0		DB 01H	仮に 00H を入力しておく

図 1.5 課題 5

課題 6

LED を全て点灯させなさい

プログラムを図 1.6 に示す。

アドレス	機械語	ニーモニック	コメント
5		PB EQU 05H	ポート B のアドレス

アドレス	機械語	ニーモニック	コメント
7		CTL EQU 07H	コントロールポートのアドレス
90		CLWD EQU 90H	コントロールワード
8400		ORG 8400H	
8400	3E 90	LD A, CLWD	コントロールワードを A レジスタに転送
8402	D3 07	OUT (CTL), A	コントロールポートに A レジスタの値を出力
8404	3A 00 85	LD A, (8500A)	8500H の値を A レジスタに転送
8407	D3 05	OUT (PB), A	ポート B に A を出力
8409	C3 00 00	JP 0000H	モニタプログラムにジャンプ

アドレス	機械語	ニーモニック	コメント
8500		ORG 8500H	
8500	FF	DB FFH	FFH を予め入力しておく。
8501		END	

図 1.6 課題 6

課題 7

LED 一つおきに点灯させなさい

プログラムを図 1.7 に示す。

アドレス	機械語	ニーモニック	コメント
5		PB EQU 05H	ポート B のアドレス
7		CTL EQU 07H	コントロールポートのアドレス
90		CLWD EQU 90H	コントロールワード

アドレス	機械語	ニーモニック	コメント
8400		ORG 8400H	
8400	3E 90	LD A, CLWD	コントロールワードを A レジスタに転送
8402	D3 07	OUT (CTL), A	コントロールポートに A レジスタの値を出力
8404	3A 00 85	LD A, (8500A)	8500H の値を A レジスタに転送
8407	D3 05	OUT (PB), A	ポート B に A を出力
8409	C3 00 00	JP 0000H	モニタプログラムにジャンプ
8500		ORG 8500H	
8500	AA	DB AAH	AAH を予め入力しておく。

アドレス	機械語	ニーモニック	コメント
8501		END	

図 1.7 課題 7

課題 8

LED の一つおきの点灯が反転を繰り返すプログラムを作りなさい

プログラムを図 1.8 に示す。

アドレス	機械語	ラベル	ニーモニック	コメント
5			PB EQU 05H	ポート B のアドレス
7			CTL EQU 07H	コントロールポートのアドレス
90			CLWD EQU 90H	コントロールワード
8400			ORG 8400H	
8400	3E 90		LD A, CLWD	コントロールワードを A レジスタに転送

アドレス	機械語	ラベル	ニーモニック	コメント
8402	D3 07		OUT (CTL), A	コントロールポートに A レジスタの値を出力
8404	3E AA		LD A, 08H	値 08H を A レジスタに転送
8406	D3 05	SHIFT:	OUT (PB), A	ポート B に A レジスタの値を出力
8408	EE FF		XOR FFH	A レジスタと FFH との排他的論理和をとる
840A	CD 40 84		CALL TIMER	タイマーを呼び出す
840D	C3 06 84		JP SHIFT	シフトにジャンプ
8440			ORG 8440H	
8440	21 00 40	TIMER:	LD HL, 4000H	値 400H を HL レジスタに転送
8443	5F		LD E, A	A レジスタの値を E レジスタに転送

アドレス	機械語	ラベル	ニーモニック	コメント
8444	2B	TLOOP:	DEC HL	HL レジスタの値から 1 を引く
8445	7C		LD A, H	H レジスタの値を A レジスタに転送
8446	B5		OR L	A の値と L の値の論理和をとる
8447	20 FB		JR NZ, TLOOP	フラグレジスタが NZ ならば TLOOP にジャンプ
8449	7B		LD A,E	E レジスタの値を A レジスタに転送
844A	C9		RET	ルーティン終了
844B			END	

図 1.8 課題 8

課題 9

図 1.3 を参考に LED の点灯位置が左にシフトするプログラムを作りなさい

プログラムを図 1.9 に示す。

アドレス	機械語	ラベル	ニーモニック	コメント
5			PB EQU 05H	ポート B のアドレス
7			CTL EQU 07H	コントロールポートのアドレス
90			CLWD EQU 90H	コントロールワード
8400			ORG 8400H	
8400	3E 90		LD A, CLWD	コントロールワードを A レジスタに転送
8402	D3 07		OUT (CTL), A	コントロールポートに A レジスタの値を出力
8404	3E 01		LD A, 01H	値 08H を A レジスタに転送
8406	D3 05	SHIFT:	OUT (PB), A	ポート B に A レジスタの値を出力
8408	7		RLCA	A の内容を左にシフトする

アドレス	機械語	ラベル	ニーモニック	コメント
8409	CD 40 84		CALL TIMER	タイマーを呼び出す
840C	C3 06 84		JP SHIFT	シフトにジャンプ
8440			ORG 8440H	
8440	21 00 40	TIMER:	LD HL, 4000H	値 400H を HL レジスタに転送
8443	5F		LD E, A	A レジスタの値を E レジスタに転送
8444	2B	TLOOP:	DEC HL	HL レジスタの値から 1 を引く
8445	7C		LD A, H	H レジスタの値を A レジスタに転送
8446	B5		OR L	A の値と L の値の論理和をとる
8447	20 FB		JR NZ, TLOOP	フラグレジスタが NZ ならば TLOOP にジャンプ

アドレス	機械語	ラベル	ニーモニック	コメント
8449	7B		LD A,E	E レジスタの値を A レジスタに転送
844A	C9		RET	ルーティン終了
844B			END	

図 1.9 課題 9

課題 10

LED の点灯位置が，はじめに左にシフトし，左端にきたら右にシフト，右端にきたら左にシフトするようなプログラムを作りなさい。
プログラムを図 1.10 に示す。

8402	D3 07		OUT (CTL), A	コントロールポートに A レジスタの値を出力
8404	3E 01		LD A, 01H	値 08H を A レジスタに転送
8406	D3 05	SHIFTL:	OUT (PB), A	ポート B に A レジスタの値を出力
8408	BF 80		CP 80H	A レジスタと 80H を比較する
840A	CA 17 84		JP Z, SHIFTR	0 ならば右シフトにジャンプ

8402	**D3 07**		**OUT (CTL), A**	**コントロールポートに A レジスタの値を出力**
840D	C2 10 84		JP NZ, NEXT	0 じゃなければ次のアドレスにジャンプ
8410	7	NEXT:	RLCA	A の内容を左にシフトする
8411	CD 40 84		CALL TIMER	タイマーを呼び出す
8414	C2 06 84		JP NZ, SHIFTL	0 じゃなければ左シフトにジャンプ
8417	D3 05	SHIFTR:	OUT (PB), A	ポート B に A レジスタの値を出力
8419	BF 01		CP 01H	A レジスタと 01H と比較する
841B	CA 06 84		JP Z, SHIFTL	0 ならば左シフトにジャンプ
841E	C2 21 84		JP NZ, NEXT	0 じゃなければ次のアドレスにジャンプ
8421	0F	NEXT:	RRCA	A の内容を右にシフトする
8422	CD 40 84		CALL TIMER	タイマーを呼び出す

8402	**D3 07**		**OUT (CTL), A**	**コントロールポートに A レジスタの値を出力**
8425	C2 17 84		JP NZ, SHIFTR	0 じゃなければ右シフトにジャンプ
8440			ORG 8440H	
8440	21 00 40	TIMER:	LD HL, 4000H	値 400H を HL レジスタに転送
8443	5F		LD E, A	A レジスタの値を E レジスタに転送
8444	2B	TLOOP:	DEC HL	HL レジスタの値から 1 を引く
8445	7C		LD A, H	H レジスタの値を A レジスタに転送
8446	B5		OR L	A の値と L の値の論理和をとる
8447	20 FB		JR NZ, TLOOP	フラグレジスタが NZ ならば TLOOP にジャンプ
8449	7B		LD A,E	E レジスタの値を A レジスタに転送

8402	**D3 07**		**OUT (CTL), A**	**コントロールポートに A レジスタの値を出力**
844A	C9		RET	ルーティン終了
844B			END	

図 1.10 課題 10

考察課題

考察課題 1

アセンブリ言語は低級言語といわれている。
定休言語とはなにか報告しなさい。

低級言語とは機械語に近いプログラミング言語のことである。
人間の使う言語とは程遠く，機械語に近い。

考察課題 2

プログラミング言語には高級言語と呼ばれるものがある。
高級言語とはなにか報告しなさい。
また，高級言語の例を 2 つ報告し，それぞれの言語が主にどのような用途で使われるかなどの特徴も報告しなさい。

英単語等が使用されており人間の使う言語に近い。
コンパイラやインタープリタで機械語に変換される。

例 1

COBOL
Common Business Oriented Language のこと

私が思う最初の高級言語。
グレース・ホッパーの理念として「機械語ではなく、英語に近い言語によってプログラミングできるようになるべきである」があり，そこから誕生した。
未だに銀行等の事務処理で使用されている。
helloworld のサンプルコードを図 2.1.1 に示す。

```
000010 IDENTIFICATION                    DIVISION.
000020 PROGRAM-ID.                       SAMPLE-01.
000030*
000040 ENVIRONMENT                       DIVISION.
000050*
000060 DATA                              DIVISION.
000070*
000080 PROCEDURE                         DIVISION.
000090 MAIN.
000100     DISPLAY "Hello world!"  UPON CONSOLE.
000110     STOP RUN.
```

図 2.1.1 helloworld

出力結果を図 2.2.2 に示す。

```
Hello world!
```

図 2.1.2 output

例 2

Brainfuck または Brainf*ck

可読性は低い。
チューリング完全なプログラミング言語。
ポインタと入力を扱う 8 つの命令によって成立している。
コンパイラを用いて機械語に翻訳する。
helloworld のサンプルコードを図 2.2.1 に示す。

```
>+++++++++[<+++++++++>-]<.>+++++++[<++++>-]<+.+++++++..+++.[-]>++++++++
[<++
++>-]<.>++++++++++[<+++++>-]<.>++++++++[<+++>-]<.+++.------.-------
-.[-]>
```

```
++++++++[<++++>-]<+.[-]++++++++++.
```

図 2.2.1 helloworld

出力結果を図 2.2.2 に示す。

```
Hello World!
```

図 2.2.2 output