

知能情報実験Ⅱ：第七回レポート

175751C 宮城孝明

平成30年10月11日

目 次

1	実験目的	2
2	実験概要	2
3	実験結果	2
4	考察	4
5	調査課題	5
6	感想	6
7	引用文献	6

1 実験目的

教育用ワンボードコンピュータ 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 (マシン語) プログラミングの作成を通して, KUE-CHIP2 の命令セットおよび機械語プログラムの構造を理解することを目的とする.

2 実験概要

コンピュータアーキテクチャについて学べる KUE-CHIP2 を用いて, プログラミングを入力し, そして, 出力の確認をした. 初めに, 簡単な説明を受けマニュアルの 1 章目を見ながら, KUE-CHIP2 の使い方を確認した. そして, マニュアルの 3 章目内容に従い, 例題プログラムを実行させた. その次に, (a) (r) までの操作を KUE-CHIP2 で実際に行った. 次に, 先の例題プログラムの少し改変した課題に取り組む. 最後に余裕のある人が, 追加課題を行った. これにより, パソコンの内部の動きがよりわかりやすくなった. さらに, 入力の際は手動で入力することで, アセンブラ言語から機械語 (マシン語) への変換作業を行った. 変換作業の時には, 先にもらった対応表での確認や, 値をロードしストアするときに, データ領域のどこにあるのかを確認に気をつける.

3 実験結果

1. 本実験の結果に関しては, 報告を省略する.
2. 本実験の結果に関しては, 報告を省略する.
3. 本実験の結果に関しては, 報告を省略する.
4. 実験の報告をする.

図 1 のコード見て欲しい. まずは, データ領域 (01H) の値を IX に代入する. その次に, ACC と ACC の排他的論理和をとる. このことにより, ACC の数値は 0 になる. ACC と (00H) を足し算し, その結果を ACC に代入する. その次の階層では, 最初に代入した IX を 1 ずつ減らしていく. 条件分岐では IX が 0 になるまで繰り返す. 戻るところを ACC と (00H) の足し算にする. そうすることで, IX の中身が 1 ずつ経るごとに ACC は (00H) の値が足されていく. これにより, IX が 0 になった時には, ACC の値は (00H) の値と (01H) の値の乗算結果が格納されている. 最後に ACC の値をデータ領域 (02H) に格納す

る. 結果は,(00H) は 4,(01H) は, 5 が代入されているため,(02H) には 20 が格納される.

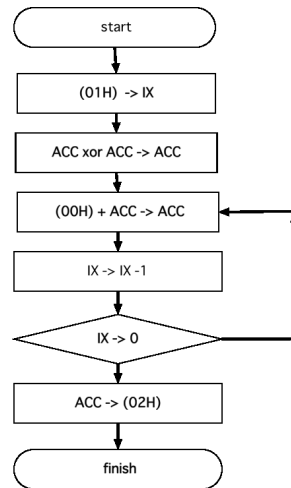


図 1: 実験 (4) のフローチャート

Listing 1: 実験 (4) のコード

	ADRS	DATA	OPECODE
1	00	6D	LD IX, (101H)
2	01	01	EOR ACC, ACC
3	02	C0	ADD ACC, (100H)
4	03	00	
5	04	AA	SUB IX, 1
6	05	01	
7	06	31	BNZ 03H
8	07	03	
9	08	75	ST ACC, (102H)
10	09	02	
11	10	0F	HLT

5. 実験の報告をする.

図 2 のフローチャートを見て欲しい. 追加課題では, 1 から最大値までの足し算することである. そのためまずは, データ領域 (00H) の値 (最大値) を IX に代入する. その次は, ACC と ACC の排他的論理和をとる. このことより, ACC の値が 0 になる. 次の階層では, ACC と IX の足し算をし, その合計を ACC に代入する. そして, IX の値を 1 ずつ減らしていく. 条件分岐では, ACC と IX の足し算のところに戻るようにする. それを IX が 0 になるまで繰り返す. そして, IX が 0 になった時には, ACC の値は 1 から最大値までの和となっている. 最後に ACC の値をデータ領域 (01H) に格納する. 結果は, (00H) に

は 20 が格納されており,1 になるまで処理を繰り返すため (01H) には 210 が格納されている.

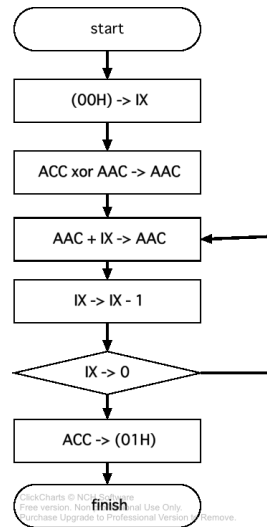


図 2: 実験 (5) のフローチャート

Listing 2: 実験 (5) のコード

	ADRS	DATA	OPECODE
1			
2	00	6D	LD IX (100H)
3	01	00	
4	02	C0	EOR ACC, ACC
5	03	B5	ADD ACC, IX
6	04	AA	SUB IX, 1
7	05	01	
8	06	31	BNZ 04H
9	07	04	
10	08	75	ST ACC, (101H)
11	09	01	
12	10	0f	HLT

4 考察

1. 実験 (1),(2),(3) について

今回の実験では,メモリやカウンタ,レジスタなどの書き込みを変更する際の sel の動作に改善案がある.現状だと,パソコンとは勝手が異なるため,書き込みの際に誤った場所にデータを格納したら確認する時に発見が遅れ,時間を掛けてしまい,学習者の意欲を失わせる可能性がある.初見でありプログラミングに苦手意識がある生徒であれば尚更,そのようなことになってしまう.改善案としては,出力

画面に格納先の情報を載せる. その他に, sel のレバー近辺に格納先をランプのような物で伝えるなどすれば良いと思われる.

2. 実験 (4),(5) について

C 言語や java とアセンブラ言語には, 一度機械語に翻訳しないと行けないという同一の点がある.

相違点としては, アセンブラは単純なプログラムを書くことはできるが, プログラムが複雑になれば行数も増え, 困難になっていく. 単純なため, 書くときのパターンもそれほど多くはない. しかし, 変数は型などを気にしなくて良い.

C 言語や java は, プログラムが複雑になっても対応できる言語である. 書くときのパターンも豊富である. しかし, 文字列や数字, 文字一つの単体によって, 変数の型が変わるという点がある.

5 調査課題

1. コンピュータの主要構成要素に関して

入力装置には, キーボードと音声入力, スキャナ, マウス, バーコード読み取り機など他に多くある.

補助記憶装置には, ハードディスク装置, SSD, SD カード, CD-R/RW ドライブ, ブルーレイドライブ, DVD-R/RW ドライブなどがあげられる.

出力装置には, ディスプレイ, プリンタ, 有機 EL ディスプレイ, プラズマディスプレイなどがあげられる.

私のもつ CPU には, intel Core i5 が使われており, インテルが生産し, i5-6360U の型番になっている. AMD が生産している. Ryzen Threadripper 1950X が型番となっている. さらに, もう一つは TSMC は生産している. A11 Bionic が型番になっている.

intel Core i5 は, 一般層向けの製品である. ゲームや動画編集はできるが, i7 にはまだまだ劣る. AMD は, intel よりも安価で性能の違いもそれほどない. A11 Bionic は, iPhone x/8/8plus に使用されており, 高性能の CPU コア「Monsoon」と高効率の CPU コア「Mistral」を使っている. これをうまく交互に使い分けて CPU の性能を向上させる.

2. 多種多様な記憶装置が用いられる理由として, 現在のノイマン型つまり, 逐次処理を行う PC である. プログラム内蔵方式のため, デー

タは一度バスを通してメモリに格納しなければならない。しかし、プロセッサと記憶装置との間のアクセスが遅ければフォン・ノイマン・ボトルネックになる。これを解決するために、記憶装置の階層化が必要である。プロセッサと記憶装置の間にキャッシュメモリ、記憶装置とハードディスクの間にディスクキャッシュを置くことで、相互の速度差を埋めることができる。搭載箇所としては、はじめにプロセッサのレジスタが一番速い。次に、キャッシュメモリは記憶装置をレジスタの間にあり、アクセス速度速くする役割である。その次が、記憶装置である。プログラムに必要なデータを格納する役割である。ディスクキャッシュは、記憶装置とハードディスクの間にあり、相互のアクセス速度を埋めてくれる役割である。最後に、ハードディスクはデータの保存をしてくれる役割である。レジスタ、キャッシュメモリの順に容量は少ないし、高価であるが、アクセス速度がとても速い。

6 感想

今回の実験は、普段使用する PC と同じ機能を持つが値の格納をデータ領域の何番値にするのか作成者本人が行わないといけなことが、とても新鮮で苦労した。さらに、実行中のエラー報告もないため、格納先で値がちゃんと合っているのかの確認をしなければ、エラーがあるのかもわからないという点は、非常に苦労した。

7 引用文献

参考文献

- [1] wikipedia: https://ja.wikipedia.org/wiki/Intel_Core_i5
- [2] pcinformation: <http://pcinformation.info/cpu/corei9corei7corei5-corei3difference.html>
- [3] chimolog: <https://chimolog.co/btosoca11/>
- [4] weblio: https://www.weblio.jp/wkpja/content/フォン・ノイマン・ボトルネック_フォン・ノイマン・ボトルネックの概要
- [5] hikaku: <https://btopcs.jp/hikaku/hikaku.cpu/>