

情報工学実験Ⅱ
実験 3-命令実行フェーズ-

レポート作成者: 055702B

池野谷克俊

共同実験者: 055722G

小林佑亮

提出日 2006 年 11 月 13 日 月曜日

1 実験目的

機械語 (マシン語) 命令をフェーズ毎に実行させ, そのときのコンピュータ内部の状態を観測することにより, 各フェーズでどのような処理が行われているかを調査し, 機械語命令の実行の仕組みを理解することを目的とする.

2 実験概要

いつもは,HLT 命令がくるまでプログラムを連続で動作させていたが, 今回の実験は 1 フェーズ毎にプログラムを実行した. またその各段階でどのような動作が起きているかを学ぶために 6 つのプログラムをもちいて実験してみた. 6 つのプログラムとは SUB 命令, LD 命令 (即値アドレスモード), LD 命令 (絶対アドレスモード), SCF 命令, AND 命令, BZ 命令を用いた簡単なプログラムである. その後, 8 ビットの 2 進数 m, n に対し, 商 $m \div n$ を求めるアセンブラプログラムを作成し, その動作を確かめた.

3 実験結果

3.1 問題

- 実験 (1),(2),(3) の結果について
 - 実行フェーズを確認するために実行した各アセンブラプログラムとそれに対応する機械語 (マシン語) プログラムを全て示せ. 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.
- 実験 (4) の結果について
 - 各自で作成したアセンブラプログラムと (a) ~ (e) のそれぞれの場合の実行結果およびフローチャートを示せ. なお, アセンブラプログラムには, 必ず, 機械語 (マシン語) プログラムも併記すること. 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.
 - 各自で作成したアセンブラプログラムがどのような動作をするプログラムなのかを, フローチャートなどを用いて説明し, 各実験結果の正当性を示せ.

3.2 解答

- 実験 (1) について

ACC には 07H が格納されている。

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	00	NOP
01	A2 05H	SUB ACC 05H
03	0F	HLT

SUB 命令の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	01	00	07	00	00	00
P0 実行後	P1 点灯	02	00	07	00	01	00
P1 実行後	P2 点灯	02	00	07	00	01	A2
P2 実行後	P3 点灯	03	00	07	00	02	A2

- 実験 (2) について

– 2-1

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	00	NOP
01	62 05H	LD ACC 05H
03	0F	HLT

LD 命令 (即値アドレス) の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	01	00	00	00	00	00
P0 実行後	P1 点灯	02	00	00	00	01	00
P1 実行後	P2 点灯	02	00	00	00	01	62
P2 実行後	P3 点灯	03	00	00	00	02	62

– 2-2

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	00	NOP
01	64 07H	LD ACC [07H]
03	0F	HLT

LD 命令 (絶対アドレス) の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	01	00	00	00	00	00
P0 実行後	P1 点灯	02	00	00	00	01	00
P1 実行後	P2 点灯	02	00	00	00	01	64
P2 実行後	P3 点灯	03	00	00	00	02	64
P3 実行後	P3 点灯	03	00	00	00	07	64

- 2-3

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	00	NOP
01	2F	SCF
02	0F	HLT

SCF 命令の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	01	00	00	00	00	00
P0 実行後	P1 点灯	02	00	00	00	01	00
P1 実行後	P2 点灯	02	00	00	00	01	2F

- 2-4

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	00	NOP
01	E2 05H	AND ACC 05H
02	0F	HLT

AND 命令の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	01	00	00	00	00	00
P0 実行後	P1 点灯	02	00	00	00	01	00
P1 実行後	P2 点灯	02	00	00	00	01	E2
P2 実行後	P3 点灯	03	00	00	00	02	E2

- 実験 (3) について

IX = 02H の場合

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	AA 01H	SUB IX 01H
01	39 05H	BZ 05H
04	0F	HLT
05	0F	HLT

BZ 命令 (分岐条件不成立時) の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	02	00	00	01	01	AA
P0 実行後	P1 点灯	03	00	00	01	02	AA
P1 実行後	P2 点灯	03	00	00	01	02	39
P2 実行後	P3 点灯	04	00	00	01	03	39

IX = 01H の場合

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	AA 01H	SUB IX 01H
01	39 05H	BZ 05H
04	0F	HLT
05	0F	HLT

BZ 命令 (分岐条件成立時) の実行フェーズ表							
フェーズ	LED	PC	FLAG	ACC	IX	MAR	IR
実行直前	P0 点灯	02	01	00	00	01	AA
P0 実行後	P1 点灯	03	01	00	00	02	AA
P1 実行後	P2 点灯	03	01	00	00	02	39
P2 実行後	P3 点灯	04	01	00	00	03	39

- 実験 (4) について

IX の初期値は 00H

アセンブラプログラム		
番地	機械語	アセンブラ言語
00	BD 01H	ADD IX (01H)
02	39 10H	BZ 10H
04	C9	EOR IX IX
05	65 00H	LD ACC (00H)
07	BA 01H	ADD IX 01H
09	A5 01H	SUB ACC (01H)
0B	32 07H	BZP 07H
0D	AA 01H	SUB IX 01H
0F	0F	HLT
10	6A FFH	LD IX FFH
12	0F	HLT

実行結果

条件	m	n	IX(結果)
(a)	0CH	03H	04H
(b)	0CH	05H	02H
(c)	0CH	0EH	00H
(d)	0CH	0CH	01H
(e)	0CH	00H	FFH

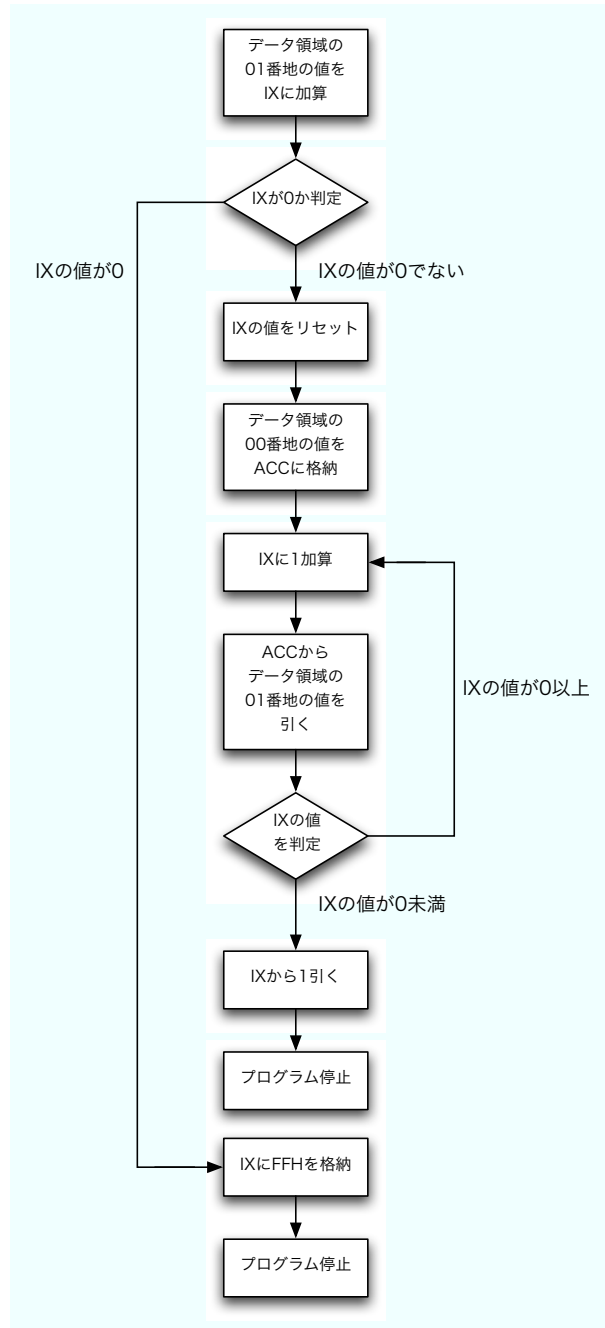


図 1: フローチャート

4 考察

- 実験 (1),(2),(3) の考察について

- P0,P1,P2,P3,P4 の各実行フェーズにおいて, どのような処理が行われているか, 実行した命令の種類毎に考察せよ.

(1) 実験 (1) について

実行フェーズ	処理内容
P0	(PC) の値を MAR に格納し,PC をインクリメント
P1	(Mem) の値を IR に格納
P2	(PC) の値を MAR に格納し,PC をインクリメント
P3	演算する 2 つの数を ALU を用いて演算し, 結果を (今回は ACC に) 格納

(2) 実験 (2)-1 について

実行フェーズ	処理内容
P0	(PC) の値を MAR に格納し,PC をインクリメント
P1	(Mem) の値を IR に格納
P2	(PC) の値を MAR に格納し,PC をインクリメント
P3	(Mem) の値を (今回は ACC に) 格納

(3) 実験 (2)-2 について

実行フェーズ	処理内容
P0	(PC) の値を MAR に格納し,PC をインクリメント
P1	(Mem) の値を IR に格納
P2	(PC) の値を MAR に格納し,PC をインクリメント
P3	(Mem) の値を MAR に格納
P4	(Mem) の値を (今回は ACC に) 格納

(4) 実験 (2)-3 について

実行フェーズ	処理内容
P0	(PC) の値を MAR に格納し,PC をインクリメント
P1	(Mem) の値を IR に格納
P2	01H を CF に格納

(5) 実験 (2)-4 について

実行フェーズ	処理内容
P0	(PC) の値を MAR に格納し,PC をインクリメント
P1	(Mem) の値を IR に格納
P2	(PC) の値を MAR に格納し,PC をインクリメント
P3	演算する 2 つの数を ALU を用いて演算し, 結果を (今回は ACC に) 格納

(6) 実験 (3) について

実行フェーズ	処理内容
P0	(PC) の値を MAR に格納し,PC をインクリメント
P1	(Mem) の値を IR に格納
P2	(PC) の値を MAR に格納し,PC をインクリメント
P3	PC をインクリメント

● 実験 (4) の考察について

- 各自が作成したアセンブラプログラムについて, 可読性や実行効率を改善するための工夫ができないか考察せよ。

今回のプログラムは,IX の値をリセットする場合に EOR を用いて 1 つの命令でリセットできるようにし, 分岐命令などを用いてるため実行効率の面では最善を尽くした。したがってこれ以上工夫することができない。また, 可読性に関しても, これ以上の工夫は考えられなかった。

● その他の考察について

- 本実験を通して得られた新たな知見について詳しく説明せよ。

今回の実験で, 分岐条件の判定は何らかの演算結果の値を判定しているということが分かった。例えば今回の実験 (4) のプログラムの 00H 番地で”LD IX (01H)”でなく,”ADD IX (01H)”としたのはロードでは分岐条件が使えないからである。

5 調査課題

5.1 課題

(a) プロセッサ (CPU) の性能を表す指標に関して、以下の設問に答えよ。

1. プロセッサ (CPU) の性能を表す指標の一つに IPC (instructions per (clock) cycle) と呼ばれるものがある。この IPC とはどのような指標か調査し、説明せよ。また、IPC の他にも、プロセッサ (CPU) の性能を表す指標はたくさんある。IPC 以外の指標についても調査し、5 つ以上挙げて、それぞれの指標について詳しく説明せよ。
2. IPC が 1 の CPU を載せたコンピュータ A と IPC が 2 の CPU を載せたコンピュータ B があり、両方のコンピュータで同じプログラムを同時に実行した。その結果、コンピュータ B の方が IPC が大きいにも関わらず、コンピュータ A の方が先に処理を終了した。この理由について考察せよ。

(b) 今回の実験で詳しく調べたように、1 つの機械語命令は、いくつかのフェーズに分けて実行される。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。また、パイプライン・アーキテクチャを採用した場合の利点と欠点についても詳しく説明せよ。

5.2 解答

- (a)-1 について

IPC とは、1 クロックあたりに実行可能な命令数で CPU の性能を表す指標である。IPC は、『実行命令数 ÷ 所要クロック数』で計算される。例えば 4 つの命令を 10 クロックで処理した場合、

$$\text{IPC} = 4 \div 10 = 0.4$$

となる。CPU の処理性能と、動作周波数、IPC の関係は CPU の処理性能 = 動作周波数 × IPC で表される。

IPC 以外の指標としては、以下のものがある。

– MIPS

MIPS は、CPU が 1 秒間に何百万回の命令を処理することができるかで CPU の性能を表す指標

- MFLOPS
MFLOPS は、技術計算でよく用いられる浮動小数点演算を 1 秒間に何百万回できるかで CPU の性能を表す指標
 - サイクルタイム
サイクルタイムというのは、CPU が読み書きの命令を出してから読み書きが完了し、次の読み取り指令が出せるまでの時間を表す。通常はこのサイクルタイムの逆数（周波数）をとり、1 秒間に何サイクルできるかで CPU の性能を表す。
 - ベンチマークテスト
コンピュータのハードウェアやソフトウェアの処理速度を計算する試験。試験用に作成されたソフトウェアを実行し、処理の完了までにかかる時間を計算することで、制作間の比較を行う。
 - iCOMP
Intel 社が開発した性能指標。整数演算、浮動小数点演算、グラフィックス処理などの性能を計測し、その結果で CPU の性能を表す。
- (a)-2 について
(a)-1 で述べたように、CPU の処理性能と IPC の関係は、
CPU の処理性能 = 動作周波数 × IPC
で表せられる。よって IPC が高くても動作周波数で劣っていると CPU の処理性能が悪い場合がある。今回の問題では、コンピュータ A の動作周波数がコンピュータ B の動作周波数の 2 倍以上あれば、コンピュータ A の方が先に処理を終了することになる。
 - (b) について
パイプライン・アーキテクチャとは全体の処理を各ステージに分割し、各ステージを並列に処理することで、単位時間あたりの処理量を向上させるというもの。この処理方式では、各命令の長さや実行時間が一定で、分岐命令等が無く、順序よく実行できないと実行効率が悪くなってしまふという欠点がある。

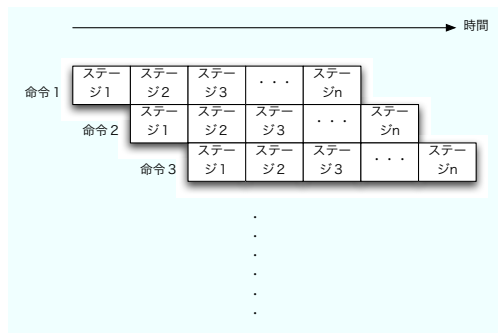


図 2: パイプライン

6 感想

今回の実験はなかなかスムーズに進められた。また、実験(4)のプログラムではTAの方に色々助けてもらったので上手く設計することができた。3回の実験で、アセンブラの基礎をしっかりと学ぶことができたのでとても有意義な実験だったと思う。

参考文献

- [1] VHDL で学ぶデジタル回路設計
 吉田たけお
 尾知博 共著