

## ディスク装置高速化技術の基礎的実験と検証

森 俊 也

稲 置 慎 也

### 【概要】

パソコン等で利用される基幹部品であるハードディスク、光ディスク、フロッピーディスク等のディスク装置は、頭脳部分である中央処理装置CPUに比較して遙かに低速であるが、近年のオペレーティングシステム(以下OS)の高度化に伴う肥大化により要求されるメインメモリの補完機能としてスワップがハードディスクに対しておこなわれるため、ディスク装置の速度がOSそのものの速度や快適性に大きな影響を与える。このため高度なOSを用いる現代ではCPUのみならずディスク装置の速度を向上させることが重要となり、ディスク製造メーカにより物理的な性能向上の努力が続けられている。

ここではソフトウェアによりディスク装置の速度性能を向上させる一つの方法として、インターリーブ値(interleave)の変更によるセクタ配列の変更をおこなう事により、ディスク装置の総合的なアクセス時間が変化する事を利用した高速化テクニックについて検証した。

### 1. はじめに

かつては日本でパソコンといえばNECのPC-88/98シリーズの事をさすことが当たり前であった時代が長く続いていたが、そのような事も過去の時代となって久しい。わが手元のPC-98とPC-88も廃棄処分の憂き目となるが、ディスク装置の基本構造は今でも変わらないので、退役前に最後のご奉公をしていただくべく実験機材とした。現代では低速なCPUとHDDではあるが、構造は現在のもので変わらず、動作が遅いだけなので実験結果が際立ち分かりやすくなり好都合と考えた。さらにディスク装置の中でも一段と低速な5.25inchフロッピーディスク装置を用いることで、実験値に大きな差が出るのが期待でき実験を簡素化することができる。

### 2. セクタ並びの変更

PC-98ならびにPC-88シリーズで利用されたオペレーティングシステムはN<sub>88</sub>-DISK BASICであるが、それ以前のPC-8001シリーズで採用されていたN-DISK BASICより読み込み・起動は低速であった。起動のためのシステムトラックはN-DISK BASICが3トラックなのに対してN<sub>88</sub>-DISK BASICでは4トラックに増えているだけなので、CPU

が高速化されていることを無視して考えても3分の4倍以下となるはずであるのに、むしろ起動時間はN<sub>88</sub>-DISK BASICのほうがN-DISK BASICよりも異常に長くなっている。その理由として挙げられるのは、N-DISK BASICでは1トラック16セクタをまとめて読み込むのに対して、N<sub>88</sub>-DISK BASICでは起動デバイスの種類が増えたことから汎用性を重視して1セクタずつ分けて読み込んでいるためと考えられる。

ところが同じフロッピーディスクでもNEC純正の8inchフロッピーディスクのメディアはインターリーブ値が13となっており、通常のIBMフォーマットのメディアを使うよりも高速に動作する。この理由は以下のようなものである。

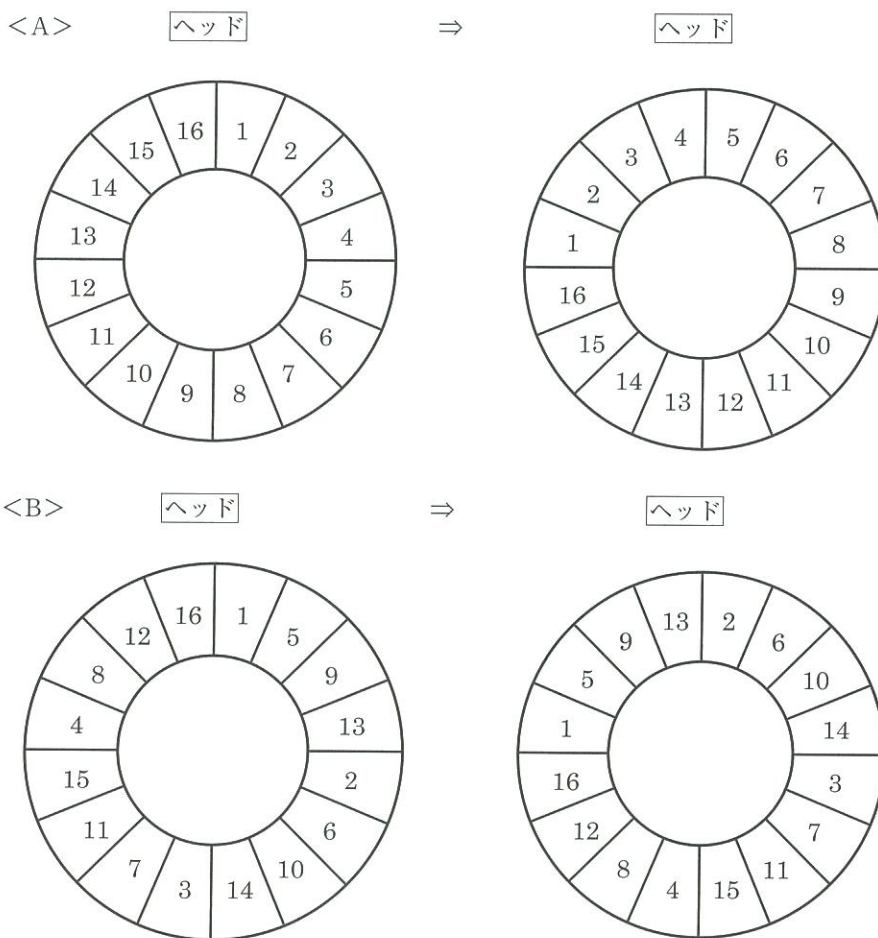


Fig.1 セクタ並び変更の原理(ディスクの回転方向は反時計回り)

ディスクの物理フォーマットとはメディアのトラック部分に、先頭部分からトラックの何番目のセクタであるとヘッダを書き込むことである。住所に例えるとトラックは丁部、ヘッダは番地部に当たる。さらに言えば、買ったばかりの白いノートがブランクディスク、トラックはページ、セクタは白いページに定規で引いた各ページの行とたとえられよう。5.25inchディスクでは、このセクタが1～16の16個あるため、この16セクタを区別して特定のセクタに対し読み書きができるようになる。このセクタは通常1～16まで順番に並んでいるが、これを飛び飛びにすればインターリーブ値を変えたことになる。

前述のようにN<sub>88</sub>-DISK BASICでは1セクタずつ読み書きをおこなうようになっているが、1セクタ読み込み後に次のセクタを読むことが可能になるまでに多少の時間がかかる。このため、セクタが順番に並んでいる時に、第1セクタ読み込み後に第2セクタを読み込もうとした際には、第1セクタの直後にある第2セクタの先頭はヘッドを通り過ぎてしまい、第2セクタがもう1周してくるまで待たなければならなくなる(fig. 1の<A>の状態)。このことは全てのセクタ読み込みに際して発生する事象なので、1回転分のロスタイムは僅かなものであっても、この繰り返しによるロスタイムの積算は無視できない大きさとなる。これがN<sub>88</sub>-DISK BASICの起動時間が長くなった原因である。

そこで、セクタを離して配置してみればどうだろうか。セクタ1と2の間に、いくつかのセクタを配置しておけば、セクタ読み込みの間のブランクタイムに不要なセクタを通過させることができるだろうという考え方である。たとえばセクタ1と2の間にセクタ5、9、13というように3つのセクタを介在させる。このようにするとトラック上のセクタの並びは、1、5、9、13、2、6、10、14、3、7、11、15、4、8、12、16となるが、セクタ1を読み込み、その処理をしているわずかの間に不要なセクタ5、9、13がヘッド上を通り過ぎる。そして丁度セクタ2を読み込もうとする瞬間にヘッド上に、まさにそのセクタ2が到着しているという目論見である。これによりディスク1回転弱の時間を待つロスタイムを省略することができるわけである(fig. 1の<B>の状態)。

### 3. 実験方法

実験には日本電気NECのPC-8801とその5.25inch両面倍密度フロッピーディスクユニットPC-80S31を用い、電源スイッチオンからN<sub>88</sub>-DISK BASICが起動してHow many files? メッセージが表示されるまでの時間をストップウォッチにて手動により計測した。さらに互換機メーカー品のフロッピーディスクユニットEPSON TF-20により追試をおこなった。

物理フォーマットの変更には内蔵フロッピーディスクユニットのFDC(Floppy Disk Controller)であるNECμPD765Aをマシン語で直接コントロールし、WRITE IDコマンドにより物理セクタを構築することによりおこなった。

#### 4. 実験に利用したプログラム

日本電気μPD765A/745/7265FDCユーザーズ・マニュアルによるフォーマッティングのためのWRITE IDコマンドのパラメータをTable. 1に、フォーマットのためのプログラムをProgram. 1に示す。

Table.1 WRITE ID コマンドのパラメータ

WRITE ID			
C	W 1	0 MF 0 0 1 1 0 1 X X X X X HD US1 US0 N SC GPL	データ長/セクタ セクタ数/トラック Gap 3 の長さ (VFO SYNC を含まず) データ領域に書き込むデータパターン
	W 1	D	
E			トラック上のセクタ数分のID情報を メインシステムより転送する
R	R 1	ST0 ST1 ST2 C H R N	READ DATAと同じ (ただし、C, H, Rは無意味)
	R 1	N	

Program. 1

```

;*****
;*
;*          インターリーブフォーマッター          *
;*
;*****
;
;          disk unit system subroutines
;
01AA      SEEK:    EQU   1AAH          ;seek track
02A4      WTFDC:  EQU   2A4H          ;send command to FDC
040E      JDUS0 :  EQU   40EH          ;get physical drive no.
042F      STHEAD: EQU   42FH          ;get head no.
041B      SETTRK: EQU   41BH          ;get cylinder no.
0263      RPHASE: EQU   263H          ;get result status
;
;          system work areas
;
7F0A      ERRCNT: EQU   7F0AH          ;error counter
7F0A      ST 1 :  EQU   7F0EH          ;start address of error status
;
;          constants
;

```

```

0010      EOT:      EQU   10H           ;max sector
0001      N:       EQU   1            ;sector length
0033      GPL:     EQU   33H          ;GAP 3 length
00E5      DATA:  EQU   0E5H         ;processing data
;
;          ORG    7000H              ;program start address
;
;          main routine
;
7000 D7      : ENT:  RST   10H         ;get 1-byte parameter form PC
7001 32B470      LD   (DRIVE), A      ;move parameter to temporal work
7004 4F         LD   C, A            ;set drive no.
7005 CD0E70      CALL FORMAT         ;call interleave format routine
7008 CD5504      CALL 455H           ;get result parameter
700B C3C100      JP    00C1H         ;jump to hot start of system main
;
;          format a disk subroutine
;
700E CD9601      FORMAT: CALL 0196H   ;recalibrate
7011 D8         RET   C              ;return if error occurred
7012 1600       LD   D, 0            ;set 0 to track counter
;
7014 CD2170      UNTIL: CALL FMTSUB   ;call 1 track format(interleave)
7017 D8         RET   C              ;return if error occurred
;
7018 14         INC   D              ;increase track
7019 CD1B04      CALL SETTRK         ;set track
701C FE28       CP    28H            ;end track(40) ?
701E 38F4       JR    C,UNTIL       ;repeat until 40 tracks
;
7020 C9         RET
;
;          format a track subroutine
;
7021 CDAA01      FMTSUB: CALL SEEK    ;seek track
7024 D8         RET   C              ;return if error occurred
7025 3E05       LD   A, 5            ;set error counter
7027 320A7F      RETRY: LD   (ERRCNT), A ;set error counter
702A 3E4D       LD   A, 4DH          ;write ID
702C CDA402      CALL WTFDC         ;send A-register to FDC
702F C5         PUSH BC
7030 D5         PUSH DE
7031 CD0E04      CALL HDUS0         ;get HD, US1, US 0
7034 E7         RST   20H
7035 CD2F04      CALL STHEAD        ;get head
7038 47         LD   B, A
7039 1805       JR    CONT
;
703B 00        NOP
703C 00        NOP
703D 00        NOP
703E 00        NOP
703F 00        NOP
;
7040 CD1B04      CONT:  CALL SETTRK   ;get cylinder
7043 57         LD   D, A
7044 3E01       LD   A, N            ;N

```

```

7046 E7          RST  20H
7047 3E10        LD    A, EOT          ; EOT
7049 E7          RST  20H
704A 3E33        LD    A, GPL          ; GPL

704C E7          RST  20H
704D 3EE5        LD    A, DATA        ; D
704F E7          RST  20H
7050 1E00        LD    E, 0          ; clear sector counter
7052 E5          PUSH HL
7053 21A370      LD    HL, SECSEQ        ; get the address of skew table
;
7056 FB          FMLOOP : EI
7057 76          HALT
;
7058 DBFA        IN    A, (0FAH)
705A E620        AND   20H ;
705C 282C        JR    Z, ERROR
705E 7A          LD    A, D          ; send cylinder no. (C)
705F D3FB        OUT  (0FBH), A
7061 FB          EI
7062 76          HALT
;
7063 DBFA        IN    A, (0FAH)
7065 E620        AND   20H ;
7067 2821        JR    Z, ERROR
7069 78          LD    A, B          ; send head no. (H)
706A D3FB        OUT  (0FBH), A
706C FB          EI
706D 76          HALT
;
7079 DBFA        IN    A, (0FAH)
707B E620        AND   20H ;
707D 280B        JR    Z, ERROR
707F 3E01        LD    A, N          ; send length(N)
7081 D3FB        OUT  (0FBH), A
7083 1C          INC  E          ; increment sector counter
7084 23          INC  HL        ; increment pointer of skew table
7085 7B          LD    A, E
7086 FE10        CP    EOT          ; end sector ?
7088 38CC        JR    C, FMLOOP ; repeat until 16 times
;
708A E1          ERROR: POP  HL
708B FB          EI
708C 76          HALT
708D F3          DI
708E D1          POP  DE
708F C1          POP  BC
7090 CD6302      CALL RPHASE        ; get result status
7093 D0          RET   NC          ; return if not error occurred
;
7094 3A0E7F      LD    A, (ST 1)    ; get error status 1
7097 E602        AND   2
7099 37          SCF
709A C0          RET   NZ          ; return for write proteted
;

```

```

709B 3A0A7F          LD      A, (ERRCNT)
709E 3D              DEC     A
709F 2086          JR      NZ, RETRY          ; repeat until ERRCNT times
;
70A1 37              SCF
70A2 C9              RET
;
;              skew table
;
70A3 0105090D      SEQSEQ: DB    01, 05, 09, 13
70A7 02060A0E      DB    02, 06, 10, 14
70AB 03070B0F      DB    03, 07, 11, 15
70AF 04080C10      DB    04, 08, 12, 16
;
;              temporary work areas
;
70B3 12          TRACK: DB    12H          ; track no.
70B4 01          DRIVE: DB    1          ; drive no.
;
70B5              END
    
```

## 5. 実験結果

PC-8801とPC-80S31の組み合わせによる実験結果をtable. 2に示す。

table. 2 セクタの並びと動作時間

interleave	セクタの並び	A	B	C	D	E
通常	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	15	6	18	4	4
1つおき	1, 9, 2, 10, 3, 11, 4, 12, 5, 13, 6, 14, 7, 15, 8, 16	15	7	18	4	5
2つおき	1, 12, 7, 2, 13, 8, 3, 14, 9, 4, 15, 10, 5, 16, 11, 6	15	7	20	4	5
3つおき	1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 4, 8, 12, 16	8	8	10	1	6
4つおき	1, 14, 11, 8, 5, 2, 15, 12, 9, 6, 3, 16, 13, 10, 7, 4	9	8	11	1	6
14 おき	1, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2	14	13	16	3	14

- A N<sub>88</sub>-DISK BASIC起動にかかる秒数
- B N-DISK BASIC起動にかかる秒数
- C N<sub>88</sub>-DISK BASIC起動とともにsetinf. n88で指定されたプログラム起動にかかる秒数
- D モニタコマンドCtrl + Rにより1トラック読み込みに要する秒数
- E Nモードで、あるプログラム読み込みに要する秒数

実験のA(N<sub>88</sub>-DISK BASIC起動にかかる秒数)とB(N-DISK BASIC起動にかかる秒数)から分かるように、N-DISK BASICではトラック単位でセクタを連続して読み込むために通常のフォーマットが最適である。一方、N<sub>88</sub>-DISK BASICではシステム起動までの所要時間はセクタを3つおきに配置した場合から減少している。ひとつおき、ふたつおきの場合では、表中では記載されていないが、コンマセカンドの単位でむしろ起動が若

干遅くなる傾向にあった。ところが3つおきの所で起動所要時間が半減し、その後再び時間が漸増する傾向にある。このことよりN<sub>88</sub>-DISK BASICではシステム起動時に、1セクタを読み込んでから次のセクタを読み込むまでの間に、3~4セクタ分回転してしまう事が推察できる。このため、実験に使用した機種でN<sub>88</sub>-DISK BASICを使用する場合は3~4セクタおきにセクタを配置すると効果が高いと言える。

引き続きディスクユニットをEPSON TF-20に変更し追試した結果をTable. 3に示す。

Table.3 セクタ配列とディスク処理時間

セクタ並びの例(16進数)																
①	01.	02.	03.	04.	05.	06.	07.	08.	09.	0A.	0B.	0C.	0D.	0E.	0F.	10
②	01.	09.	02.	0A.	03.	0B.	04.	0C.	05.	0D.	06.	0E.	07.	0F.	08.	10
③	01.	0C.	07.	02.	0D.	08.	03.	0E.	09.	04.	0F.	0A.	05.	10.	0B.	06
④	01.	05.	09.	0D.	02.	06.	0A.	0E.	03.	07.	0B.	0F.	04.	08.	0C.	10
⑤	01.	0E.	0B.	08.	05.	02.	0F.	0C.	09.	06.	03.	10.	0D.	0A.	07.	04
⑥	01.	09.	04.	0C.	07.	0F.	02.	0A.	05.	0D.	08.	10.	03.	0B.	06.	0E
⑦	01.	08.	0F.	06.	0D.	04.	0B.	02.	09.	10.	07.	0E.	05.	0C.	03.	0A
⑧	01.	03.	05.	07.	09.	0B.	0D.	0F.	02.	04.	06.	08.	0A.	0C.	0E.	10
⑨	01.	0A.	03.	0C.	05.	0E.	07.	10.	09.	02.	0B.	04.	0D.	06.	0F.	08
⑩	01.	09.	06.	0E.	03.	0B.	08.	10.	05.	0D.	02.	0A.	07.	0F.	04.	0C
⑪	01.	04.	07.	0A.	0D.	10.	03.	06.	09.	0C.	0F.	02.	05.	08.	0B.	0E
⑫	01.	05.	09.	0D.	04.	08.	0C.	10.	03.	07.	0B.	0F.	02.	06.	0A.	0E
⑬	01.	06.	0B.	10.	05.	0A.	0F.	04.	09.	0E.	03.	08.	0D.	02.	07.	0C
⑭	01.	09.	08.	10.	07.	0F.	06.	0E.	05.	0D.	04.	0C.	03.	0B.	02.	0A
⑮	01.	10.	0F.	0E.	0D.	0C.	0B.	0A.	09.	08.	07.	06.	05.	04.	03.	02

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)
①	16.2	12.1	15.3	32.2	9.4	27.8	50.6	11.0	27.3	48.7	250.6
②	16.9	12.3	15.4	24.5	9.7	28.0	49.9	11.2	27.1	49.2	244.2
③	17.0	13.0	15.7	26.3	10.6	28.0	50.7	12.7	28.8	52.5	255.3
④	16.0	12.5	15.2	25.6	11.7	29.8	51.6	12.9	29.9	53.0	258.2
⑤	9.0	5.0	9.3	16.5	11.0	20.3	42.6	12.4	21.0	43.4	190.5
⑥	9.6	6.0	7.5	17.7	11.7	21.0	43.5	12.7	22.0	44.2	195.9
⑦	10.6	6.4	8.3	19.6	11.5	18.1	38.5	13.5	20.1	39.1	185.7
⑧	11.6	7.1	9.1	21.2	12.5	22.7	45.0	12.6	23.0	45.5	210.3
⑨	11.4	7.0	10.0	21.8	12.8	23.4	45.8	12.9	23.7	45.2	214.0
⑩	11.9	7.7	10.7	22.4	13.2	25.1	47.9	14.4	24.2	46.2	223.7
⑪	12.5	8.5	10.8	24.1	10.3	26.6	48.2	14.4	27.4	47.9	230.7
⑫	13.1	8.9	12.2	25.4	7.7	26.3	48.4	14.4	26.0	48.0	230.4
⑬	13.6	9.5	12.1	26.8	7.7	26.3	47.9	15.7	28.0	49.3	236.9
⑭	14.5	10.1	12.5	28.1	7.7	25.8	47.8	15.1	27.6	49.5	238.7
⑮	14.9	10.9	13.2	29.4	8.3	26.3	48.0	11.4	27.8	51.2	241.4
	12KB (注1)	12,294B分のプログラム (4トラック、49セクタ)			9728バイト (3トラック、39セクタ)			9656バイト (3トラック、38セクタ)			

注1：3トラック、8セクタでシステムイニシャライズ時間を含む

注2：R/W…リードアフターライドモード



**【表中記号の説明】**

- (A) N-<sub>88</sub>DISK BASICシステム立ち上げ時間
- (B) Load
- (C) Save
- (D) save R/W(注2)
- (E) Bload
- (F) Bsave
- (G) bsave R/W(注2)
- (H) INPUT#
- (I) PRINT#
- (J) PRINT#R/W(注2)
- (K) 合計

**6. 考察とまとめ**

フロッピーディスクを題材としてディスク装置のインターリーブ値変更による高速化の基礎実験をおこなった。実験よりインターリーブ値をダイナミックに変更することによりアクセス時間を減少させることができ、ディスク高性能化に寄与することを確認できた。近年のディスク装置、特にハードディスクにおいては回転数アップ、記録密度の向上、DMAの採用、キャッシュメモリ(参考文献9)の増大等により高速化が図られており、インターリーブ値の採用のメリットは少なくなっている。しかしながらCPUの高速化に追いついていないのが現状であり、さらにはより低速な光学ドライブの活用も増えていることから、このような方法の検討は今後も必要であると考えられる。

**【参考文献】**

- (1) 新谷真澄、フロッピーディスクフル活用ガイド、日本ソフトバンク(1986)
- (2) 坂本・玉川・山内、フロッピーディスク活用ハンドブック、秀和システム(1983)
- (3) 日本電気、 $\mu$ PD765A/745/7265FDCユーザーズ・マニュアル、日本電気
- (4) 若松登志樹、Z80わかる機械語入門、新星出版社(1984)
- (5) 牟田・田川・平松・栗山、PC-Techknow8000、システムソフト(1982)
- (6) 栗山・平松・松尾、PC-Techknow8800、システムソフト(1982)
- (7) 藤田・幸田、PC-Techknow9800、システムソフト(1984)
- (8) 森・山本、最新ハードディスク入門、ソフトバンク(1989)
- (9) 森俊也、増設メモリ活用ブック2、ソフトバンク(1991)