

Kontrollern

In i floppykontrollerkort

Här kommer en beskrivning av floppykontrollers (med reservation för eventuella felaktigheter). kan vara bra att ha ifall man vill skriva egna formateringsprogram mm.

1.

Kontrollerkortet styrs via I/O-kommandon på databoardbussen. dessa kommandon har följande utseende:

Basic-instruktion Funktion

Z=INP(7)	Reset alla I/O-kort
OUT 1,kortadr	Card select dvs val av I/O-kort
OUT 0,data	Sänder data till kort
OUT 2,C1	Kommando C1 till kort
OUT 3,C2	Kommando C2 till kort
OUT 4,C3	Kommando C3 till kort
OUT 5,C4	Kommando C4 till kort
INP(0)	Läser data från kort
INP(1)	Läser status från kort

2.

Innan kommunikation med kontrollern kan ske måste kontakt med kortet upprättas med hjälp av Card Select (CS).

MF0:	CS 44
MO0:	CS 45
SF0:	CS 46

BASIC-funktion att använda

```
DEF FNCardselect(C)
  OUT 1,C
  RETURN 0
FNEND
```

3.

Kommandogivning till kontrollern påbörjas genom att I/O-kommandot C1 ges dvs OUT 2,0, då ges CPU:n på kontrollern ett NMI (Non-Maskeble-Interrupt) och den görs sig klar för att ta emot kommandon enligt följande: 4 byte skickas på, port 0 efter varandra, benämnda K0,K1,K2,K3.

BASIC-funktion att använda

```
DEF FNKommando(K0,K1,K2,K3)
  OUT 2,0
  WHILE INP(1) AND 2 : WEND
  Z=FNSändbyte(K0)
  Z=FNSändbyte(K1)
  Z=FNSändbyte(K2)
  Z=FNSändbyte(K3)
  RETURN 0
FNEND
```

BASIC-funktion att använda

```
DEF FNLäs$ LOCAL Sect$,A,J
  WHILE INP(1) AND 64 : WEND
  WHILE J<256
    WHILE (INP(1) AND 1)=0 : WEND
    A=INP(0)
    Sect$=Sect$+CHR$(A)
    J=J+1
  WEND
  RETURN 0
FNEND
```

Buffert till kontrollern innebär att 256 bytes sänds från dator till kontrollerns buffert.

BASIC-funktion att använda

```
DEF FNSkriv(Sect$) LOCAL A,J
  WHILE INP(1) AND 4 : WEND
  WHILE J<256
    WHILE (INP(1) AND 1)=0 : WEND
    A=ASC(RIGHT$(Sect$,J+1))
    OUT 0,A
    J=J+1
  WEND
  RETURN 0
FNEND
```

I instruktionsmod har bitarna 3-5 i K1 följande betydelse:

bit 3	formatera diskett
4	läs adressmärke
5	-

Statusporten indikerar följande information (när status tages direkt från kontrollerkretsen). Obs status är inverterad.

bit 7	NOT READY
6	WRITE PROTECT
5	HEAD LOADED/WRITE FAULT
4	SEEK ERROR/RNF
3	CRC ERROR
2	TRACK 0/LOST DATA
1	INDEX PULSE/DRQ
0	BUSY

DEF FNSändbyte(B)

```
  WHILE (INP(1) AND 1)=0 : WEND
  OUT 0,B
  RETURN 0
FNEND
```

4.

Följande betydelse har de olika koderna i kommandot (Obs att listan är ej helt komplett, och den varierar förmodligen också från kontrollern till kontrollern).

K0	bit 0	Läs sektor
	1	Buffert till värd
	2	Buffert till kontrollern
	3	Skriv sektor
	4	Instruktions mod
	5	Selectdrive
	6	Motor on
	7	
K1	bit 0	Drive Nr
	1	..
	2	..
	3	Instruktion om instruktionsmod
	4	..
	5	..
	6	Buffert i kontrollern
	7	..
K2		Spår Nr (0-79)
K3		Sektor Nummer
	bit 0	Sektor inom "kluster" om ABC832
	1	och bit 5-7 =kluster inom spår!!
	2	
	3	
	4	
	5	Sektor Nr om
	6	ABC830 dvs
	7	160K drive

I kontrollern finns det 3 st buffertar om 256 bytes, (gäller tidigare kontrollers de nya har mycket mera minne, men man kan anropa de 3 beskrivna buffertarna) de används för att lagra de sektorer som skall skrivas/ eller har lästs. Flera bitar i ett kommando kan kombineras ex: Läs sektor + buffert till värd. Buffert anges så här

K1 bit 6 & 7	00	buffert 0
	01	buffert 1
	10	buffert 2

Läsning och skrivning görs genom att sektorn läses in till en angiven buffert, samt genom att en buffert skrivs ut.

Buffert till värd innebär att informationen i angiven buffert överförs till datorn.

Detta är den information som finns i FD179x status-register de som är mer intresserade av själva kontrollkretsen bör läsa Western Digitals databöcker (finns på ELFA). Vid formatering krävs att följande info finns i bufferterna innan formatteringskommandot ges.

Enkel densitet (128byte/sect)

```
6 st 00
1 st FE (ID adress mark)
1 st track-number
1 st side-number
1 st sector number
1 st 00 (sector length)
1 st F7 (2 CRC skrivs)
11 st FF
6 st 00
1 st FB (data adress mark)
128 st Data
1 st F7 (2 CRC skrivs)
27 st FF
```

Dubbel densitet (256byte/sect)

```
12 st 00
3 st F5
1 st FE (ID adress mark)
1 st track-number
1 st side-number
1 st sector-number
1 st 01 (sector-length)
1 st F7 (2 CRC skrivs)
22 st 4E
12 st 00
3 st F5
1 st FB (data adress mark)
256 st Data
1 st F7 (2 CRC skrivs)
54 st 4E
```

Denna information upprepas för varje sektor på skivan. Förutom denna information finns info som inleder och avslutar spår.

Läs adressmärke, vid kommandot skall K2 vara spår nummer. Följande data erhålls:

```
Byte 1 Spåradress
2 Sidnummer
3 Sektoradress
4 Sektorlängd
5 Checksumma 1
6 Checksumma 2
```

Man kan endast läsa och skriva sektorer med de vanliga I/O- kommandona samt läsa adress-märke och initiera formatering. Förutom detta kan kontrollern läsa och skriva hela spår åt gången, när man formatterar så skriver man ett helt spår åt gången med den beskrivna informationen, detta för att kontrollern skall hitta sektorerna vid läsning och skrivning av sektorer. Med skrivning av hela spår kan man åstakomma MYCKET snabba kopieringsprogram (bra att ha vid säkerhetskopiering, se tex programmet fastback för IBM PC), ty man både skriver data och formatterar på engång, men för att göra dessa konststycken med en kontrollers så måste den ha tillräckligt med minne minst så att ett helt spår ryms på en gång.

<5651>
Mikael Liden

BASIC II-funktioner

```
1 REM Insänd av Mikael Liden <5651>
1986-03-16 19.44.36
1000 INTEGER : EXTEND
1010 DIM Sect$=256,Sf$(3)=300
1020 POKE VAROOT(Sect$)+4,0,1
1030 !
1040 !
1050 DEF FNCerr$
1060 IF E9 AND 128 RETURN 'LUCKAN ÖPP
EN'
1070 IF E9 AND 64 RETURN 'SKIVAN SKRI
VSKYDDAD'
1080 IF E9 AND 32 RETURN 'SKRIV FEL'
1090 IF E9 AND 16 RETURN 'SÖK-FEL'
1100 IF E9 AND 8 RETURN 'CRC-FEL'
1110 RETURN 'ÖVRIGT FEL'
1120 FNEND
1130 !
1140 !
1150 DEF FNRsect(Dev$,Sect)
1160 D=FNDsel(Dev$)
1170 IF LEFT$(Dev$,2)='MO' Z=FNCcontco
m(1+2,D,Sect/8,Sect*32)
1180 IF LEFT$(Dev$,2)='MF' Z=FNCcontco
m(3,D,Sect/32,Sect/4*32+(Sect AN
D 3))
1190 IF (FNStatus AND 8)=0 E9=FNrdat :
RETURN E9
1200 RETURN FNRbuf
1210 FNEND
1220 !
1230 !
1240 DEF FNWsect(Dev$,Sect)
1250 D=FNDsel(Dev$)
1260 IF LEFT$(Dev$,2)='MO' Z=FNCcontco
m(12,D,Sect/8,Sect*32)
1270 IF LEFT$(Dev$,2)='MF' Z=FNCcontco
m(12,D,Sect/32,Sect/4*32+(Sect A
ND 3))
1280 Z=FNWbuf
1290 IF (FNStatus AND 8)=0 E9=FNrdat
1300 RETURN E9
1310 FNEND
1320 !
1330 !
1340 DEF FNFormdisk(Dev$,Dd,Ft,Tt)
1350 D=FNDsel(Dev$)
1360 Z=FNFormat(D,Dd,Ft,Tt)
1370 RETURN 0
1380 FNEND
1390 !
1400 !
1410 DEF FNFdisk(Dev$)
1420 Z=FNFormdisk(Dev$,-1,0,79)
1430 Z=FNWait
1440 IF (FNStatus AND 8)=0 E9=FNrdat
1450 RETURN E9
1460 FNEND
1470 !
1480 !
1490 DEF FNDsel(Dev$)
1500 IF LEFT$(Dev$,2)='MF' Z=FNCsel(4
4)
1510 IF LEFT$(Dev$,2)='MO' Z=FNCsel(4
5)
1520 IF LEFT$(Dev$,2)='SF' Z=FNCsel(4
6)
1530 RETURN ASCII(RIGHT$(Dev$,3))-48
1540 FNEND
1550 !
1560 !
1570 DEF FNFormat(Dd) LOCAL A,B,I,J
1580 IF Dd RESTORE 1720 : I=3 ELSE RE
STORE 1690 : I=2
1590 WHILE J<I
1600 READ A,B
1610 Sf$(J)=''
1620 WHILE A<>-1
1630 Sf$(J)=Sf$(J)+STRING$(A,B)
1640 READ A,B
1650 WEND
1660 J=J+1
1670 WEND
1680 RETURN 0
1690 DATA 6,0,1,254,4,0,1,247,11,255,
6,0,1,251,128,64,1,247,97,255,-1,
-1
1700 DATA 256,255,-1,-1
1710 !
1720 DATA 12,0,3,245,1,254,3,0,1,1,1,
247,22,78,12,0,3,245,1,251,197,
64,-1,-1
1730 DATA 59,64,1,247,196,78,-1,-1
1740 DATA 256,78,-1,-1
1750 FNEND
1760 !
1770 !
1780 DEF FNFormsend(Dd) LOCAL I,J
1790 Z=FNFormat(Dd)
1800 Z=FNContcom(4,0,0,0)
1810 Sect$=Sf$(0) : Z=FNWbuf
1820 WHILE Dd
1830 Z=FNContcom(4,64,0,0)
1840 Sect$=Sf$(1) : Z=FNWbuf
1850 Z=FNContcom(4,128,0,0)
1860 Sect$=Sf$(2) : Z=FNWbuf
1870 RETURN 0
1880 WEND
1890 Z=FNContcom(4,128,0,0)
1900 Sect$=Sf$(1) : Z=FNWbuf
1910 RETURN 0
1920 FNEND
1930 !
1940 !
1950 DEF FNFormat(Drive,Dd,Ftrack,Ttrac
k)
1960 Z=FNFormsend(Dd)
1970 Z=FNContcom(16,Drive+8,Ftrack,Tt
rack)
1980 RETURN 0
1990 FNEND
2000 !
2010 !
2020 DEF FNCsel(Card)
2030 OUT 1,Card
2040 RETURN 0
2050 FNEND
2060 !
2070 !
2080 DEF FNRescard=INP(7)
2090 !
2100 !
2110 DEF FNStatus=INP(1)
2120 !
2130 !
2140 DEF FNrdat=INP(0)
2150 !
2160 !
2170 DEF FNWdat(D)
2180 OUT 0,D
2190 RETURN 0
2200 FNEND
```