CS / 2200 Ramblings

Collected by: Tyler B. Olsen

2236MXE Terms:

CCV Communications Control Vector -- 29 Bytes to setup Async TC code.

Byte 01	Number stop bits	Transmission rate
Byte 02	Transmission mode	Break signal enable
Byte 03	Number data bits per character,	Parity option
Byte 04	Received data substitute character for parity framing errors.	
Byte 05	Received data timeout interval, in units of 0.1 seconds.	
Byte 06	End of record detection flag.	
Byte 07	Upshift character.	
Byte 08	Downshift character.	
Byte 09	Break signal transmit interval, in units of 10 milliseconds.	
Byte 10	Break signal detect interval, in units of 10 milliseconds.	
Byte 11-18	Up to four pairs -- Special character with transmission delay.	
Byte 19-20	reserved.	
Byte 21-29	are MXE vector byte positions for flow control options.	

CSV Communications Status Vector -- 8 bytes flow control options.

Byte 01	Break signal received.
Byte 02	Modem signals. (DSR and received line signal detector)
Byte 03	Parity / framing errors.
Byte 04	Binary count of number of characters in receive buffer.
Byte 05	Binary count of number of EOR characters in receive buffer.
Byte 06	Received data timeout countdown.
Byte 07	Binary count of number of characters in transmit buffer.
Byte 08	Modem signals. (DCD, RI, DSR, CTS, DTR, RTS, XOFF rcvd, XOFF sent)

$GIO commands for Async communications on MXE port.

. Set CCV
. Read CSV
. Load transmit code translation table
. Load receive code translation table
. Disconnect
. Send Break Signal
. Start receiving data.
. Transfer received data to CPU.
. Send data
. Send, then receive data
. Stop transmitting
. Continue transmitting
. Reset controller
. Set signals.

Question:
I have switched to the DS and my applications seem to run slower.

Answer 1: You may have replaced Phoenix drives on two disk controllers with a single DS cabinet. You now have one channel to the CPU instead of two.

Answer 2: The raw speed of the Phoenix is faster than a DS. If your software does not take advantage of the cache memory available it may well run slower. However, the DS has a cache memory of up to 1K sectors. On some reads to disk as many as 32 sectors are brought into cache. The Phoenix has a much smaller cache of 8 sectors.

Answer 3: One VAR has found that the time for an expected response could drop from 2 seconds to 20 seconds. If you move files to the DS surfaces properly this slowness will not occur.

1. Optimize the index size,
2. Use the new hash algorithm,
3. Move heavily used files close to the index area.

-- 1. Optimize the index size,
Determine how many files will eventually be on the disk surface.

Set an index of a size that will be used to 75-80 % capacity.

A full index uses 16 file names per sector, 15 in sector 0. 75% of capacity would be 12 names per sector. A 10 sector index is saturated with 159 names, comfortable with 120. A default index of 24 sectors has room for 383 file names.

-- 2. Use the new hash algorithm,

When you move files to the DS cabinet it may pay to use the new disk index hashing algorithms. A description of the differences was detailed in an article for TechKnowledge in September 1987. The old "hash" algorithm "hashed" items to every third sector in the index and then. The new algorithm "hashes" more equitably to every sector. The new algorithm also uses a look-ahead rather than a look-behind feature.

-- 3. Move heavily used files near the index area.

Files should also be moved so that the heavily used files occur close to the index sectors to minimize head movement.

Index Structuring: A Better Way

by Tyler Olsen

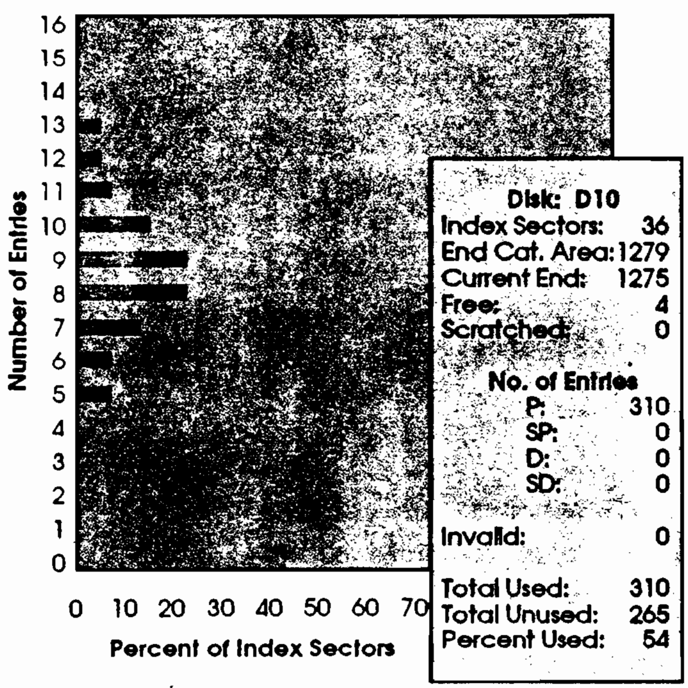
The CS/2200 computer uses disks as a primary media for program and data storage. Its BASIC-2 operating system has a built-in catalog mode for accessing logical disk surfaces so that users can find program and data files automatically. For each logical disk surface, the assigned catalog disk area includes space for index sectors and and space for programs and data storage.

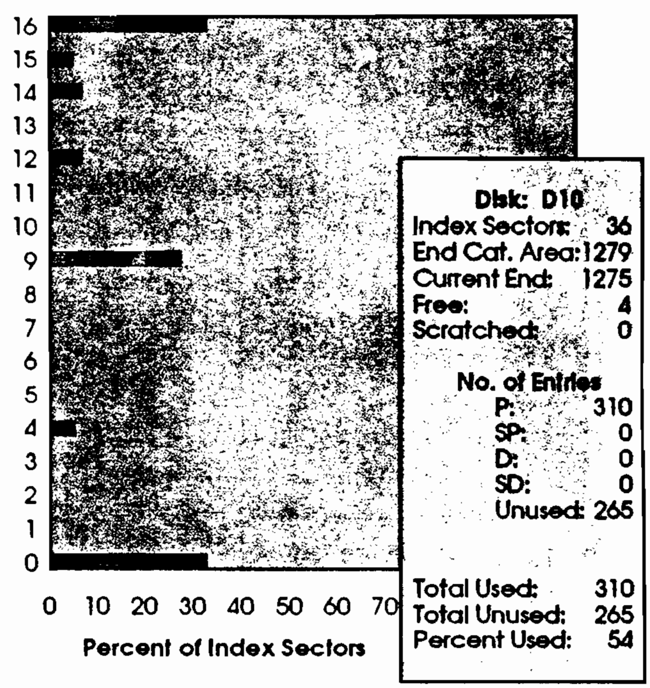
The catalog index lists file names, types, and locations in the storage sectors of the disk surfaces. The index area begins in sector zero and continues through a user-requested number of sectors; that is, the user chooses how much of the catalog disk area will be dedicated to index sectors as well as how large the catalog disk area for that surface will be. The balance of the catalog disk area is left over for program and data storage.

ASSIGNING SPACE

By applying the SCRATCH DISK command to a specific disk surface, the user establishes the form and size of the catalog index area as well as the maximum size of the program and data area. The initial assignment is by a BASIC-2 command such as

SCRATCH DISK T /D25 , LS = 50, END = 38000





New versus old index structuring: The new CS/2200 hashing method (left) distributes entries among index sectors more efficiently than the old (right).

Since the 2200 series was introduced in the early 1970s, two hashing (index-structuring) techniques based on BASIC-2 commands such as this one have been introduced. The new hash method was introduced as a more efficient way of assigning index sector usage and managing disk accesses as an option in release 2.5 of the CS/2200 BASIC-2 operating system. All subsequent releases of the operating system support both the old and new hashing methods.

THE CRITICAL DIFFERENCE

The first index sector contains disk-surface data and up to 15 file pointers; subsequent index sectors contain up to 16 file pointers. The hashing algorithm used by the operating system points to a specific sector in the index area.

The old hashing method is invoked with this command:

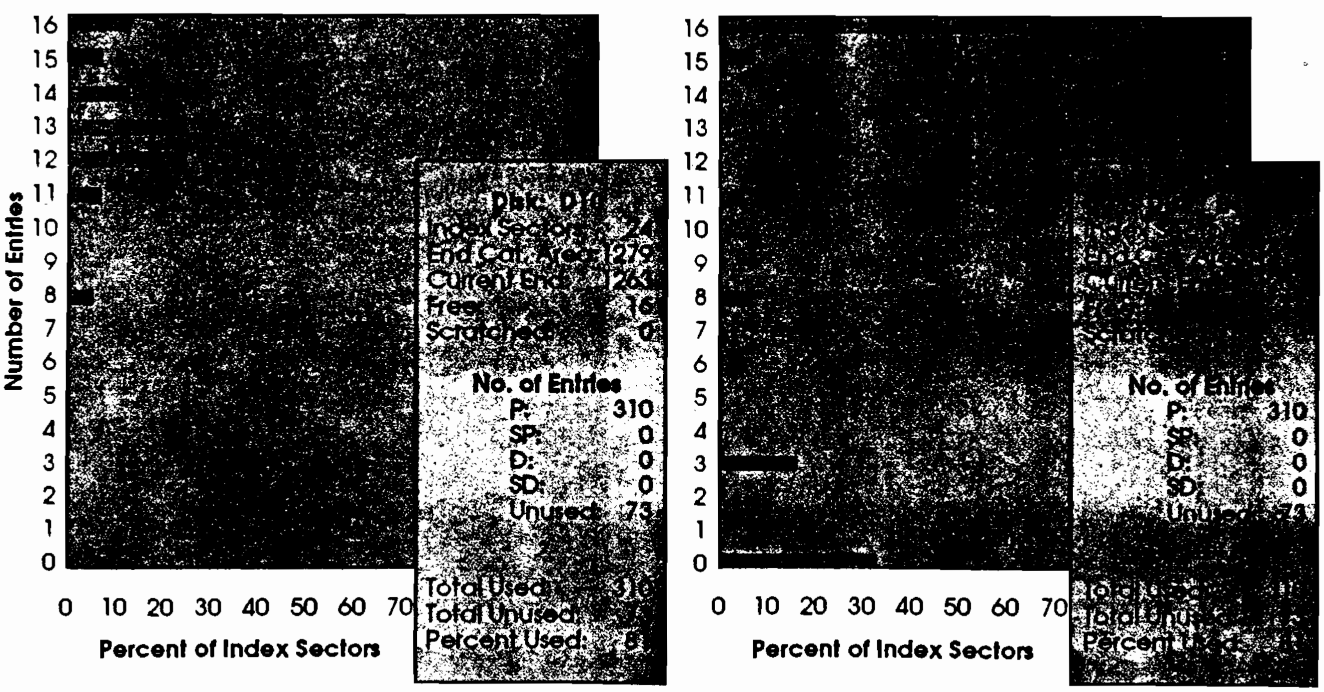
SCRATCH DISK T /D25 , LS = 50, END = 38910

while the new hashing method is invoked with

SCRATCH DISK ' T /D25 , LS = 50, END = 38910

where the prime symbol invokes the new hashing method, /D25 identifies the specific logical disk surface, LS = ii designates the size in sectors of the index area, and END = nnnnn designates the maximum number of sectors in the catalog area.

The hashing algorithm used by the operating system quickly points to a specific sector in the



Though both the old (right) and new (left) hashing methods use 81 percent of the index area, the old hashing algorithm leaves 66 percent filled to capacity, suggesting substantial backward overflow. What little overflow the new method yields is forward overflow, facilitating caching by the data processing unit to reduce the number of performance-crippling disk accesses.

index area based upon a mathematical calculation on the eight-byte file name being referenced. The difference between the old and new hashing methods is in this search algorithm and what happens when there is a file-reference overflow from one index sector to another. An overflow occurs when a hashing calculation points to an index sector that already contains its full complement of 16 items. The old hashing method looked to the prior sector for the overflow; the new method looks to the next sector. To understand how the two methods achieve such different results, let's look at how the CS/2200 manages forward data overflow storage.

The new Wang data storage (DS) cabinets [see March 22, 1987 (711-1311) *TechKnowledge*, p. 10] come with a data processing unit (DPU) which has a given amount of fast storage buffer, or cache, memory. Up to 1,024 sectors of cache memory are available. If a requested sector is within the DPU cache memory, no real disk access has to be made for the data; consequently, information in cache can be transferred to the CPU much faster. The DPU caches to requested disk sectors using a modified least-recently-used algorithm. Such sector caching, therefore, reduces disk accesses. Running the DS utility @HITRATE will show the cache hit efficiency.

GRAPHIC DIFFERENCES

Two sets of index-usage results illustrate the efficiency differences between the old and new hashing methods. For both sets, a program was written to randomly name and open 310 files on a single diskette.

In the first case, the old hashing algorithm left 33 percent of the index sectors empty and filled another 33 percent to capacity. By contrast, the new hashing method distributed data much more evenly and logically among the index sectors, yielding a consolidated bell curve when usage was plotted against number of entries per sector.

The difference is even more striking in the second set of index-usage results. In this set, both the old and new hashing algorithms used 81 percent of a 24-sector index area. The old hashing method filled 66 percent of the sectors to capacity, suggesting substantial backward overflow. Again, the new hashing method yielded a bell-curve distribution around 12 to 16 items per sector, suggesting little forward overflow. What overflow there is, is handled by the DS cache technique so that performance is not greatly affected.

BEST USE

The savings that can be realized using the new hashing method are especially noticeable when programs or program overlays are loaded. A disk surface that is primarily for program storage might be organized to use the new hashing method in a minimum number of index sectors. When programs are moved to this surface, the most-often referenced programs and program overlays might be moved first, grouping them nearer to the index.

If data files are opened and closed repeatedly, they might also benefit by being assigned to a location on the disk surface by the new hashing method. With release 3.1 of the CS/2200 operating system, the number of files that can be opened concurrently within a partition has increased from 16 to 256 so that conversion of data-file disk surfaces to the new hashing method might not be necessary to increase access efficiency. Some applications, however, may have been written requiring the old hashing method. Code can be added to these applictions to make them compatible with either method.

Expanded main CPU memory in the MicroVP and CS/2200 computers [see October 1987 (711-1324) *TechKnowledge*, p. SW-12] put a potenital of 8MB of CPU RAM disk at the user's disposal. The disk is accessible as disk 340, a totally separate disk address. While the CPU RAM memory is fast, it is also volatile and easily lost on a power outage. Still, it remains the ideal medium for easily recovered files such as programs or sort work areas.

k

Tyler Olsen
is a principal software engineer for Wang Laboratories' (Lowell, MA) CS/2200 Product Group.

CS/2200 Disk access comparisons
All times in milliseconds

Disk Model	Sequential Access Times	Random Access Times over entire Platter	Sequential access Times over 1st 2000 Sectors	Typical Sequence Average Access Time
2270 Diskette	52.7	391.6	329.0	98.5
2275 Diskette	24.6	329.4	313.3	132.9
DS 320KB	19.6	110.3	84.9	17.7
LVP diskette	18.9	282.3	190.0	66.5
DS 1.2 MB	15.2	161.6	93.3	36.3
2275 10MB	7.9	133.0	67.0	15.8
2275 32MB	6.8	66.3	48.5	14.6
LVP 32MB	6.6	89.4	45.5	14.0
DS 10MB	6.0	141.5	36.3	18.1
DS 20MB	5.0	90.8	23.4	10.1
DS 64MB	4.9	42.5	15.1	7.7
DS 32MB	4.8	59.5	21.1	8.0
DS 140MB	4.7	46.3	14.3	7.7
2280 80MB	4.2	43.1	23.8	10.2
DS RAM disk	3.4	3.4	3.4	3.4
CS RAM disk	1.5	1.5	1.5	1.5
<u>Diskettes</u>				
2270 Diskette	52.7	391.6	329.0	98.5
2275 Diskette	24.6	329.4	313.3	132.9
DS 320KB	19.6	110.3	84.9	17.7
LVP diskette	18.9	282.3	190.0	66.5
DS 1.2 MB	15.2	161.6	93.3	36.3
<u>Hard disks</u>				
2275 10MB	7.9	133.0	67.0	15.8
2275 32MB	6.8	66.3	48.5	14.6
LVP 32MB	6.6	89.4	45.5	14.0
DS 10MB	6.0	141.5	36.3	18.1
DS 20MB	5.0	90.8	23.4	10.1
DS 64MB	4.9	42.5	15.1	7.7
DS 32MB	4.8	59.5	21.1	8.0
DS 140MB	4.7	46.3	14.3	7.7
2280 80MB	4.2	43.1	23.8	10.2
DS RAM disk	3.4	3.4	3.4	3.4
CS RAM disk	1.5	1.5	1.5	1.5

Disk Drive Specifications

	320-KB Diskette	1-MB Diskette	20-MB Hard Disk	32-MB Hard Disk	64-MB Hard Disk	140-MB Hard Disk
Disk Platters	1	1	2	2	4	14 or 7
Capacity/ Platter	320-KB (360-KB for PC)	1-MB (1.2-MB for PC)	10-MB	16-MB	16-MB	10-MB or 16-MB
Sectors/ Platter	1,280 (1,440 for PC)	4,160 (4,800 for PC)	38,912	65,024	65,024	38,912 or 65,024
Bytes/ Sector	256 (512 for PC)	256 (512 for PC)	256	256	256	256 256
Average Access Time	100 ms	100 ms	68 ms	45 ms	27 ms	27 ms 27 ms
Data Transfer Rate	250 KB/sec	500 KB/sec	5 MB/sec	5 MB/sec	5 MB/sec	5 MB/sec 5 MB/sec

Release 3.0 for Multiuser BASIC-2 Available; Answers to Data Storage

by Tyler Olsen
Wang Labs, Inc.

The latest version (Release 3.0) of the Wang Multiuser BASIC-2 Operating System is now available. Release 3.0 OS will operate on any CS or any 2200 CPU with 28K of control memory. The software is shipped with the new CS units; it is available as a WSS automatic update; or may be ordered individually by calling WangDirect. The release contains the software on diskette, and four documents: "2200 Programming in BASIC," "BASIC-2 Utilities Reference Manual," "Customer Service Release Notice for 3.0," and "Multiuser BASIC-2 Reference Manual."

Model Number	**Media Code**
195-0049U-X	3, 5, 9
Product Name	
Wang Multiuser BASIC-2 Operating System 3.0	

There have been many questions about the Wang Data Storage (DS) cabinet for the Wang CS or 2200 system. The DS can be cabled to either a disk, disk/printer, or triple (disk/printer/terminal) controller in an I/O slot of the CPU. If a DS is multiplexed with other CS or 2200 CPUs, it can plug into the 2275MUX controller board.

The DS cabinet contains four compartments, each designed to hold two half-height or one full-height devices. The upper two visible compartments are accessible from openings in the front panel; the lower two interior compartments are covered by the front panel. (Figure 1.)

The DS cabinet contains cables for a streaming tape cassette drive (STCD), a floppy drive, a Winchester removable, and a daisy chain cable to handle up to four Winchester drives (four fixed or three fixed and one removable). Two eight position switches are used for assignment of

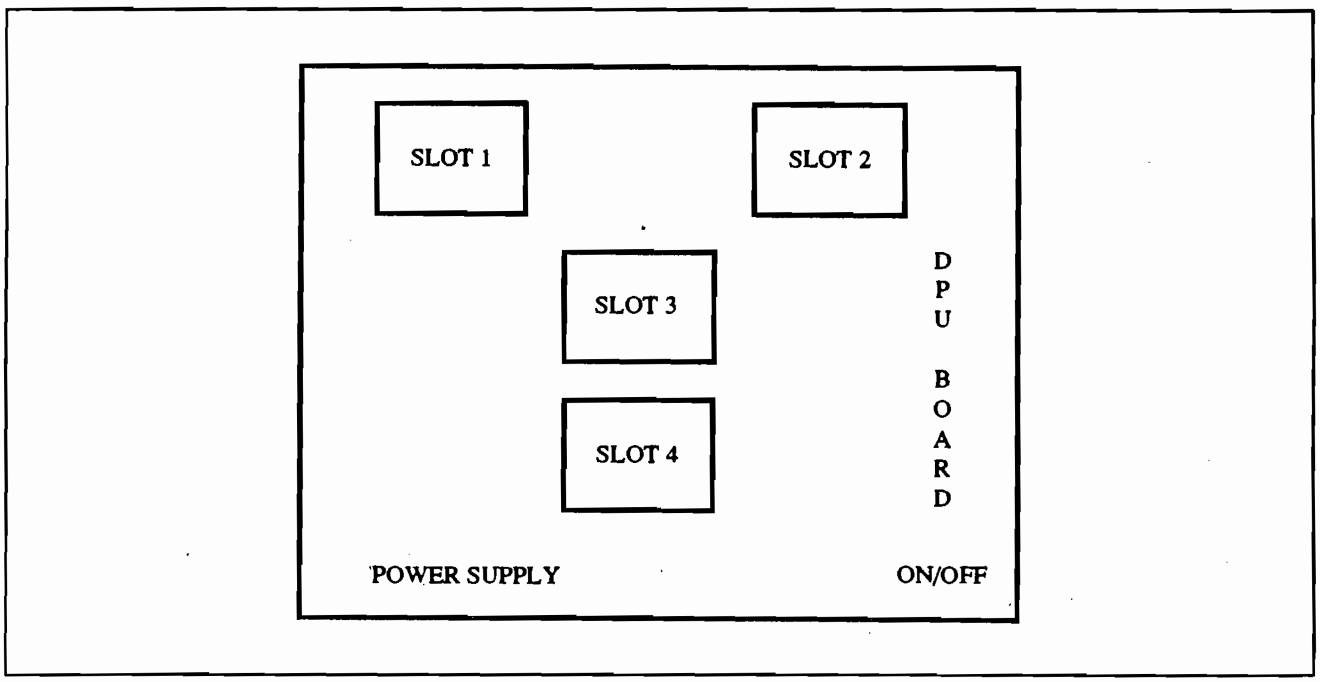


Figure 1. The Data Storage Cabinet for the CS/2200

addresses to the fixed disks. SW.1 is used for drives 1 and 2; SW.2 for drives 3 and 4. The maximum number of logical disk surfaces for SW.1 and SW.2 is 14 each. Table 1 shows how device addressing on the CS works.

The fixed drive in the data storage cabinet uses one address per logical disk surface; e.g., four surfaces would use four addresses. Table 2 shows the amount of storage available for the fixed Winchester disks for the Data Storage Cabinet.

Table 1. DS Addressing on the CS

DEVICE		UNIT 10	UNIT 20	UNIT 30
Master Devices - 40 bit off				
Diskette		/D10	/D20	/D30
SW.1	Fixed	/D11	/D21	/D31
	Hard	thru	thru	thru
	Disk	/D1E	/D2E	/D3E
Removable				
	Hard	/D1F	/D2F	/D3F
	Disk			
Slave Addresses - 40 bit on				
DPU RAM Disk		/D50	/D60	/D70
SW.2	Fixed	/D51	/D61	/D71
	Hard	thru	thru	thru
	Disk	/D5E	/D6E	/D7E
Streaming				
Tape	(STCD)	/D5F	/D6F	/D7F
Cassette	Disk			

Table 2. Table of fixed Winchester storage capacities

Data Storage Type	#Available Logical Disk Surfaces	Amount of Available Storage Per Surface	Total Available Storage
DS-20	2	10 mg	20 mg
DS-32	2	16 mg	32 mg
DS-64	4	16 mg	64 mg
DS-112*	7	16 mg	112 mg
DS-140	14	10 mg	140 mg

*DS-112 is a DS-140 with SW.1 or SW.2 switches set by installing CE for DS-112 operation.

5 1/4-inch Diskette: 320/360KB vs. 1/1.2MB

by Tyler Olsen

The various diskette media and drive options now available to CS/2200 users provide flexible storage options. The diskettes provide options of "porting" data and software between and among CS/2200 systems as well as with the Wang PC. This article dwells on the 5 1/4-inch diskettes: available types, how they are formatted; compatibility with the Wang PC; and compatibility between CS/2200 drives.

TWO TYPES OF DISKETTES AND DRIVES

There are two types of diskette media, DSDD or DSHD, which must be considered when using a 5 1/4-inch diskette drive. The CS/2200 double-sided, double-density (DSDD) diskettes store 320KB or 360KB of data. The double-sided, high-density (DSHD) diskettes introduced with the Wang Data Storage Cabinet store 1MB or 1.2MB of data.

There are two types of 5 1/4-inch diskette drives available with the CS/2200, either 320/360KB or 1/1.2MB. The 2200/2275 drive introduced the 320/360KB 5 1/4-inch diskette. The CS Data Storage Cabinet requires a 5 1/4-inch diskette drive, which may be either a 320/360KB drive or a 1/1.2MB drive.

The DSHD 1/1.2MB formatted diskettes cannot be read by a 320KB drive. The 320/360KB DSDD diskette can be used with both types of drives—320/360KB or 1/1.2MB—but the read/write results vary depending on the diskette/drive combination.

A WORD ABOUT FORMATTING

Before a diskette can store information, it must be formatted for the system with which it will be used. Formatting leaves flux markings (binary 0s and 1s) at locations determined by the drive on the diskette's oxide media surface. Formatting creates circular tracks (analogous to phonograph tracks) on a diskette. In addition to creating the tracks, formatting allocates fixed-length storage areas, called "sectors," along the tracks.

A diskette fresh out of the cellophane pack is not formatted and, thus, has no track or sector markings. Conversely, a diskette that has been used will probably have clearly defined track and sector markings. These last diskettes may cause troubles if used interchangeably between DSDD and DSHD drives.

Tracks. A 320/360KB drive will format 40 tracks at 48 tracks per inch. A 1/1.2 MB drive will format 80 tracks at 96 tracks per inch. In other words, the relatively wide read/write head of the 320/360KB drive describes a track wider than the thin path described by the relatively narrow 1/1.2MB drive.

Sector size. The standard format for CS/2200 BASIC-2 disk operations uses 256-byte sectors. This format should be used for normal CS/2200 and 2200 MicroVP operations. Diskettes formatted

Read/Write Comparisons

Capability		320/360KB Drive	1/1.2MB Drive
Read	320/360KB diskette	Yes	Yes
Write	320/360KB diskette	Yes	No*
Format	320/360KB diskette	Yes	No
Read	1/1.2MB diskette	No	Yes
Write	1/1.2MB diskette	No	Yes
Format	1/1.2MB diskette	No	Yes

*No because flux changes from 320KB format may not be erased.

by a 320/360KB drive with 256-byte sectors have 320KB of storage. Diskettes formatted by a 1/1.2MB drive with 256-byte sectors have 1MB of storage.

An alternative of 512-byte sectors may be specified. Processing of 512-byte sectors is normally used only for PCs or transfer of diskette information between the CS/2200 and PCs. Diskettes formatted by a 320/360KB drive with 512-byte sectors have 360KB of storage. Diskettes formatted by a 1/1.2MB drive with 512-byte sectors have 1.2MB of storage.

[For more information on diskette formats, see the "DS Data Storage Cabinet User's Manual" (715-0740), from which excerpts have been taken for this article.]

512-BYTE SECTOR DISKETTES

The 5 1/4-inch diskette also supports the Wang PC 512-byte sector format so that data can be easily interchanged between the Wang CS/2200 and PC systems. DSDD diskettes formatted by a 320/360KB drive with 512-byte sectors have a capacity of 360KB; DSHD diskettes formatted by a 1/1.2MB drive with 512-byte sectors have a capacity of 1.2MB. The 1MB drive cannot format 320KB or 360KB diskettes.

The 512-byte formatting is transparent to the CS/2200 operating system because it actually maps two 256-byte logical sectors into one 512-byte physical sector. CS/2200 BASIC-2 can then access diskettes with the 512-byte PC formats as if the diskette were formatted with 256-byte sectors, and all BASIC-2 operations can be performed. Even through the 512-byte format provides more diskette capacity, it it not recommended for normal CS/2200 operations because disk write is considerably slower than with the 256-byte format.

DSDD [illegible] vs. DSHD [illegible]

Characteristic	[illegible]	[illegible]
Number of heads	[illegible]	[illegible]
Read/write head width	[illegible]	[illegible]
Number of tracks formatted	40	80
Tracks per inch formatted	48	96
Number of 512-byte sectors per track	8/9	15
Number of 256-byte sectors per track	16	30
Diskette type accepted	[illegible]	DSHD, DSDD*

*Read only recommended.

DRIVE-TO-DRIVE CONSIDERATIONS

Fresh Wang diskettes formatted and written on similar drives should pose no problems in reading and writing. Fresh Wang diskettes formatted and written on a 320/360KB drive and read on a 1/1.2MB drive should pose no problem in reading.

Where a diskette has been formatted with a wide track (on a 320/360KB drive) and is then written on a 1/1.2MB drive, there will surely be problems.

Compatible Drive and Media (1/1.2MB). If a 1MB formatted diskette is used with a 1MB drive, read/write is no problem because there is proper alignment between the diskette tracks and the read/write head path. Data resides on the media in 80 tracks, or at a density of 96 tracks per inch.

Compatible Drive and Media (320/360KB). Likewise, if a 320KB formatted disk is used with a 320KB drive, read/write is no problem. Data resides on the media in 40 tracks, or a density of 48 tracks per inch.

Differing Drive and Media (1/1.2MB drive and 320/360KB media). When a 320KB formatted diskette is used with a 1MB drive, the difference in read/write head path becomes important. With data residing on the diskette in 40 tracks at a density of 48 tracks per inch, the wide 320KB diskette tracks align easily with the thin 1MB drive path for reading. Reading should present no problem.

Writing to a 320/360KB diskette by a 1/1.2MB drive is not advised because a good portion of the diskette sector lies outside the drive-head write path.

Differing Drive and Media (320/360KB drive and 1/1.2MB media). The worst-case situation is realized in the combination of a 1MB formatted diskette used with a 320KB drive. In this situation, there is virtually no alignment between the diskette tracks and the wide path described by the 320KB drive head. Data would reside on the diskette in 80 tracks, or at a density of 96 tracks per inch. It could not be written or read properly by the wide read/write heads.

CONCLUSION

Two different 5 1/4-inch diskette drives are available for CS/2200 systems. The advantage of the 1/1.2MB drive is greater media storage. If used for back-up or portability to other systems with 1/1.2MB drives, the 1/1.2MB drive with formatted DSHD diskettes is the logical alternative.

The advantage of the 320/360KB drive is a potential for greater media portability. If used for portability of media to other systems with either 320/360KB of 1/1.2MB drives, the 320/360KB drive is the logical alternative.

For normal CS/2200 operations, format diskettes to a 256-byte format. For PC compatibility, format diskettes to 512-byte sectors—but remember that doing so may slow down writing speed.

If you are trying to upgrade to the latest, use the 1/1.2MB alternative. If you must retain a tie to the past perhaps you should stick with the 320/360KB alternative. If you need compatibility with the past and a connection to the future, why not two Wang DS cabinets, one with a 320/360KB drive and one with a 1/1.2MB drive? k

Tyler Olsen is a principal software engineer for Wang Laboratories (Lowell, MA) CS/2200 Product Group.

CS/2200 T.C. Emulations

ASC

Teletype Emulator
IBM 2741 (Selectric)

BSC

Multileaving Hasp
IBM 2780
IBM 3780
IBM 3741
Wang 2200 to 2200
2200-WPS mode

Burroughs TC500

Burroughs Poll/Select

BSC 3275

IBM 3275 BSC emulation

BSC 3271

IBM 3271 BSC emulation
(9 device streams total of mix below)
8 3277 terminals, 4 3288 printers.

RCM

Remote Control Maintenance

VS/2200 via 2258 LCO link

VS/2200 via BSC package and VS TC COPY.

PC/2200

via 2236DE terminal emulators.

Modem connections:

2227N	Null modem	Async only.
2228N	Null modem	Async or Bisync.
Wang Telemodem		0-2400 baud
WA3451		0-1200 baud async only.

NOTE: 2200 Terminals require 11 bit mode, when talking as program terminals.

NOTE: Wang TeleModem is not compatible with WA-3451 in 11 bit mode.

Table 6. Controllers for Communications

Emulation	MXE	2227B	2228B	2228C	2228D D-4	2258 LCO
Asynchronous						
RCM	Yes	Yes	Yes	Yes	-	-
ASC	Yes	Yes	Yes	Yes	-	-
ASC With Flow Control	Yes	-	-	-	-	-
Asynchronous/Synchronous						
Burroughs Poll Select			Yes	Yes	-	-
Synchronous						
2780/3780/3741	-	-	Yes	Yes	-	-
2200-2200	-	-	Yes	Yes	-	-
2200/WP	-	-	Yes	Yes	-	-
2200/VS TCCopy	-	-	Yes	Yes	-	-
IBM 3275	-	-	-	Yes	-	-
IBM 3271 BSC	-	-	-	-	Yes	-
2200/VS	-	-	-	-	-	Yes

Table 7. Line Speeds of the Communication Controllers

Device	Line Speed (bps)	Mode
MXE	110-19,200	Terminal Mode
	110-9,600	Asychronous TC
2227B/28B/28C	110-9600	Asychronous mode
2228B/28C	0-4800	Synchronous
2228D-4	0-9600	Synchronous

CS / 2200 Resource sharing

Disk Multiplexing

- Each CPU can have a maximum of 3 disk controllers.
- 2275-MUX.
  allows a DS or 2280 drive to be shared with up to 3 other CPUs.
- 2275-MUXE
  Extends the 2275-MUX range to 3 more CPUs.
- A single DS cabinet can thus be shared by 1-16 CPUs.

User Partitions.

- Each CPU can support 1-16 terminals.
- Each CPU can support 1-16 partitions.
- Each partition has a maximum size of 56K. (61K in memory bank 1).
- Any partition may run in foreground or background.
  Foreground partitions have access to a terminal.

  Background partitions do not require terminal access.

  Background partitions may be started in the foreground and then sent into the background.

  Simple programming inclusions can be made to allow partitions to run in the foreground if a terminal is available or in the background if it is not.

This example was incorporated in the BSC communications package.

Module "BSC*010M" lines 5620-5780.

```
5720 F8=-F8 : ON F8+2 GOTO , , 5760
5725 Set items for foreground

5760 Set items for background and $RELEASE TERMINAL
```

Variable F8	PRINT to	$IF ON test #9	Terminal Mode
-1	/005	/001	Foreground, background possible.
0	/005	/001	Foreground ONLY.
+1	/000	/005	Background, foreground possible.

/000 as an output device is a bit bucket.
/001 is the input keyboard.
/005 is the CRT, in the background it can be tested for availability.

CS/2200 Tool Box Option Updated

by Tyler B. Olsen

[Editor's note: Tyler Olsen is the primary author of the CS/2200 Tool Box Option.]

An updated version of the CS/2200 Tool Box Option, offered through the ISWU Software Library, is now available. The Tool Box Option is a subset of a collection of BASIC-2 programs that have been developed over the lifespan of the CS/2200 line to facilitate the development and maintenance of system software.

These programs are collected and maintained in a loose-leaf notebook. The programs are created by a variety of people but are collected and sometimes modified by one.

This update includes several changes from earlier versions. The cross-reference lister has been modified to include most BASIC-2 release 3 language enhancements. Disk prompting has been modified to include addresses of the Data Storage Cabinet. New features have been added to the disk catalog sort. Other program functions have been added, some enhanced.

TOOL BOX GROUND RULES

My ground rules for developing and maintaining this collection are:

- The programs must reside on single diskette surface. The maximum number of sectors is 1232 on a 2270A-type drive. These criteria keep the number of programs to a manageable number.
- The basic operating instructions are to mount the diskette, then key RESET LOAD RUN RETURN and follow the prompts. There is documentation with the package which provides program abstracts in addition to operating instructions and descriptions of some functions.
- The functions must be useful to the CS/2200 system user. System diagnostics enable you to look at the hardware to see what is connected. Interrogation tools enable you to look at programs and data files. Documentation aids provide helpful tools for development or system maintenance. And some of the programs are for the creation, listing, and maintenance of telecommunications-formatted data files.

Tool Box Option diagnostic programs include "Monitor Partition Status," "What Disks Are On the System?" "Show Status Of Printers, TC Boards, and Terminal," "Analyze Disk Index," and "Show CRT Character Set and Keyboard Values." Interrogation tools include "Sort the Disk Catalog Area," "Disk Catalog Examination," and "File ZAP."

Documentation and development aids include "Sort the Disk Catalog Area," "Program or System Comparison," and "Cross-Reference Listing."

Other programs represent experiments that might be useful or of interest; e.g., "Analyze Disk Index," "Flow Chart Maker," "Data File Comparison," "Number Conversions," "Analyze $GIO Statements," "@CLOCK" a clock and calendar, and many others.

Some of the programs contain one or more subroutines to perform a particular function. You may want to study these routines for code that could easily be included in your own system software. TBODISKS, for example, has code to look for all possible disk addresses; TBO.XDAD is used to determine the disk type; and @CLOCK creates large block letters and digits on the CRT.

If I find myself on an unknown system, I might run "Monitor Partition Status," "Show Status of Printers," and "What Disks Are On the System?"

If I'm looking at software packages, I am liable to use "Sort the Disk Catalog Area," "Cross-Reference Listing," or "Program or System Comparison." If used with all their bells and whistles, these three functions are adequate to document and maintain a software system.

A FEW PROGRAM DETAILS

The "Cross-Reference Listing" creates a cross-referenced listing of one or more programs from one or more disk surfaces. This program has been upgraded to process most of the language enhancements of release 3 of the Wang Multi-User BASIC-2 operating system.

The cross-reference program creates a listing of each program module that is page-numbered and titled. The first section is a decompressed listing of each program line; the summary section describes file references and variables used, prime subroutine locations, then statement and prime subroutine references.

The program uses tables that enable the user to describe BASIC-2 variables, prime functions, and file references with mnemonic descriptors of up to 16 characters. These mnemonic descriptors may be output in the margin of the listing as well as in the summary description. Use of the mnemonic descriptors is optional, but they can be extremely useful in documenting programs and systems.

"@CLOCK" uses the MXE board to provide a data and time clock. The clock has a large digital face and monthly calendar or message display area. A small file of operator-entered reminder messages is maintained for each terminal.

"Sort the Disk Catalog" sorts the items in the catalog index area for viewing. Data or program file names may be ordered by sector number, reverse sector number, or name. If ordered by name, either all or a common one- to eight-character ID may be requested. The display will show name and sector information only as with LIST DCT, but sorted. Alternatively, it will show the sector information with the descriptive remark or the image statement normally found as the first line of any module. Provision is also made for output of descriptions of data files.

The sort has an option of looking for programs modified after a certain date. (I put a MM/DD/YY stamp on the front of each program module.)

Another option lets you look only for the presence or absence of an item from a list of names; this feature lets you see if all modules of a particular system reside on a particular surface. The list of names found is saved in a common array, which may be used as input by other programs. An exit is provided for this feature.

"Program or System Compare" compares two programs line by line and displays the differences. Alternatively, it will compare the modules on two different disk surfaces from a list of program names.

"What Disks Are On the System?" shows the cabinet type for each of the three disk controllers. It also shows the catalog index summary for each disk drive.

ORDERING INFORMATION

The CS/2200 Tool Box Option is available to all ISWU members. The package costs $45 and is available on three types of diskette: 8-inch SSSD, 8-inch DSSD, and 5 1/4-inch.

To order the CS/2200 Tool Box Option (or if you have any questions), call the ISWU Software Library at (617) 967-1058. Have your ISWU membership number as well as a purchase order number ready. To order by mail, send a completed order form (following this article) along with a check or purchase order. Indicate the type of media you prefer on your order.

Allow three to four weeks delivery time for any software order. Express mail service is available for an additional $25. Specify express mail service on your order if you wish to take advantage of it.

Tyler Olsen
is a principal software engineer for the Wang Laboratories, Inc. (Lowell, MA) CS/2200 Product Group.

CS/2200 Tool Box Option

Tool Box Option   What do you wish to do?
TO OPERATE -- Press S.F. KEY or DIGIT corresponding to name,
or position # via Alpha, RETURN, SPACE, or BACKSPACE and key RUN

- Tool Box Utilities and program development aids
- Clock and calendar
- Tool Box Diagnostic aids
- LOAD RUN xx from another surface
- Monitor Partition status
- Prepare or list TC format data
- @MENU

Tool Box Utilities

- Sort the Disk catalog area
- Cross-reference listing
- Program or system comparison
- File ZAP
- Analyze data file
- Data file comparison

Tool Box Diagnositics

- What disks are on the system?
- Monitor LIST DT
- Scan disks on the system
- CRT character set and keyboard values
- Show status of Printers, TC boards, & terminal
- Analyze $GIO statements
- Analyze Disk index

TIPS FOR IMPROVING YOUR BASIC-2 PROGRAM'S PERFORMANCE

by CS/2200 Development Team

This article provides some insight into how the CS/2200 BASIC-2 language processor functions, thus enabling you to improve the performance of your programs. In particular, the article addresses:

. How the processor performs line number searches (GOTO, GOSUB)

. How the processor performs a search for the value of a variable (A = A + 1)

. How the processor performs a search for a user-defined subroutine (DEFFN', GOSUB')

. How the processor performs a search for a user-defined function (DEFFN, A = FNB(x))

SEARCH FOR A LINE NUMBER

When the user executes a program in memory, the BASIC-2 language processor performs many functions that are grouped together in a phase known as resolution. One of these functions "threads" your program in line number sequence order. It is because of this function that you do not have to enter your program in line number sequence.

Another function of this phase divides your program into as many as 16 groups of lines. Each group contains INT(L/16+1) number of lines, where L is the total number of lines in memory. The memory address of the first line of each group is stored in a table.

Whenever your program must search for a line number (GOTO, GOSUB, and others), it calculates the group in which the line should be found, and uses the table entry associated with that group to begin its search.

This technique was implemented so that a search for a particular line number does not always need to start at the beginning of the program.

SEARCH FOR THE VALUE OF A VARIABLE

During the resolution phase, memory is allocated for each variable encountered in your program. During this phase, a table is constructed that contains a pointer to the first variable whose name begins with a particular letter of the alphabet (there are 26 pointers, one for each letter). This pointer is the starting point of a "linked list" of the variables that all have a common first letter.

When a value is needed, the leading letter of the variable name is used to get the address of that letter's linked list. The linked list is then searched for the particular variable. The variables within the linked list are listed in the reverse order in which they are encountered in memory.

To optimize your program, you should make use of all 26 letters of the alphabet to minimize the lengths of the linked lists, thus reducing your search time. Additionally, you should define your variables in reverse order of their frequency of use (most frequently used variable defined last).

SEARCH FOR A USER-DEFINED SUBROUTINE

During the resolution phase, the address of the first 16 user-defined subroutines is stored in a table.

When your program references a user-defined subroutine (GOSUB'), the table is used to quickly retrieve a pointer to the subroutine and program execution begins there. If the user-defined subroutine is not listed in the table, the BASIC-2 language processor uses the address of the last entry in the table as a starting point for a search through memory for the desired user-defined subroutine.

Therefore, to maximize your program's performance, you should do the following:

. Limit the number of defined user-defined subroutines to 16.

. Define the most frequently-used subroutines first within your program.

. Group together in memory the 16th and all succeeding user-defined subroutines, thereby reducing the memory search time.

SEARCH FOR A USER-DEFINED FUNCTION

During Resolution phase, the address of the first user-defined function found in the program is stored in a memory location known to the BASIC-2 language processor. When a reference to a user-defined function is made, the pointer is read out of memory and is used as a starting location for a memory search for the function specified.

Therefore, to optimize your program, you should group your user-defined functions together in your program. The grouping should be in order of frequency of use, with the most frequently used defined first. This grouping minimizes the time needed to locate the desired function.

History lesson: CS / 2200 Disk(ette) Family Tree

. Memorex floppy disk -- Single floppy (incompatible with subsequent models)
          (1023 sectors)   64 tracks of 16 sectors. .

..8 inch SSSD white label diskettes.
. 2270   (Shugart) floppy disk -- One, two, or three floppy drives.
          (1023 sectors)   64 tracks of 16 sectors. .

. 2270A (Shugart) floppy disk -- One, two, or three floppy drives.
          (1231 sectors)   77 tracks of 16 sectors.

. 2270A (Shugart) floppy disk -- One, two, or three floppy drives.
          (1231 sectors)   77 tracks of 16 sectors.

..8 inch SSDD Red label diskettes.
. LVP or SVP internal diskette drive.

..5 1/4" inch DSDD (177-0080) diskettes.
. 2275   diskette drive.   (1279 sectors)
. DS-320 diskette drive.   (1279 sectors)

..5 1/4" inch DSHD (177-   ) diskettes.
. DS-1.2 diskette drive.   (4159 sectors)

CS / 2200 Hard Disk Family Tree

.. SVP / LVP Winchester

.. 2230 Series   (Diablo)

.. 2260 Series   (CDC Hawk)

.. 2280 Series   (Phoenix)
   (950 watts running, 250 watts standing)   BTU 1050 / hour

..DS Winchesters.

.. DS 10R	10 MB	Removable	1 x 10 mb surface.
.. DS-20	20 MB	fixed	2 x 10 mb surfaces.
.. DS-32	32 MB	fixed	2 x 16 mb surfaces.
.. DS-64	64 MB	fixed	4 x 16 mb surfaces.
.* DS-112	112 MB	fixed	7 x 16 mb surfaces.
.. DS-140	140 MB	fixed	14 x 10 mb surfaces.

Disk Multiplexing

. Each CPU can have a maximum of 3 disk controllers.

. 2275-MUX.
   allows a DS or 2280 drive to be shared with up to 3 other CPUs.
. 2275-MUXE
   Extends the 2275-MUX range to 3 more CPUs.

. A single DS cabinet can thus be shared by 1-16 CPUs.

Configurations:

```
                                    2236 MXE board
    4 x RS232 ports
    Channel 1             Channel 2             Channel 3             Channel 4
                             !
                             ! RS 232 cable
                             !
                             !

                          ____________
                          !  Data      !
                          !Concentrator!
                          !____________!
                             !
                             ! (balanced line serial
                             !  multi-drop RS422 line
                             !  1-144 bar code reader wands
                             !--      1500 foot limit.
```

Configurations:

```
                                    2236 MXE board
    4 x RS232 ports
    Channel 1             Channel 2             Channel 3             Channel 4
                             !                   !
                           ! ! Touch pad         ! OCR page reader
                             !
                             !
                             !
                          _____________
                          ! Standard   !
                          ! Terminal   !
                          !____________!
```

Configurations:

```
                                    Multiple CPUs.
    CS/D                  CS/D                  CS                    CS/D
    ! ! !                 ! ! !                 ! ! !                 ! ! !
    ! ! .-x'10            ! ! .-x'10            ! ! .-x'10            ! ! .-x'10
    ! !                   ! !                   ! !                   ! !
    ! .---x'20            ! .---x'20            ! .---x'20            ! .---x-20
    !                     !                     !                     !
    .-----x'30            .-----x'30            .-----x'30            .-----x'30
```

Object: configure trying to mimimize channel bottleneck.
where /340 on each CPU is CPU RAM disk -- avg access time 1.5 ms/sector.
Typical use: program files

x'10 on each CPU is internal disk (20-140mB)
Typical use: program files or system specific data files.

x'20 on each CPU is shared external disk (20-316MB).

x'30 on each CPU is shared external disk (20-316MB).