LA-UR-80-2741

MASTER

TITLE: I/O PERFORMANCE MEASUREMENT ON CRAY-1 AND CDC 7600 COMPUTERS

AUTHOR(S): Ingrid Y. Bucher
Ann H. Hayes

SUBMITTED TO: Computer Performance Evaluation Users Group (CPEUG)

Orlando, FL October 20-23, 1980

University of California

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545
An Affirmative Action/Equal Opportunity Employer

Form No. 836 R3
St. No. 2629
12/78

UNITED STATES
DEPARTMENT OF ENERGY
CONTRACT W-7405-ENG. 36

I/O PERFORMANCE MEASUREMENT ON CRAY-1 AND CDC 7600 COMPUTERS

Ingrid Y. Bucher
Ann H. Hayes

Computer Science and Services Division
Los Alamos Scientific Laboratory
Los Alamos, NM 87544

Disk I/O transfer rates and overhead CPU times were measured as functions of buffer size and number of logically independent I/O channels for several operating systems and 16 I/O routines on the Cray-1 and CDC 7600 computers. By parameterizing the codes for a variable number of channels, buffer sizes, and words transmitted, the effect of these variables is observed for buffered, nonbuffered, and random-access I/O transmissions. To measure CPU-overlapped performance, I/O was performed concurrently with a pretimed compute loop. Rates, sector overhead, and CPU transmission speeds were calculated upon completion of I/O. Effects of memory blocking due to vector operations were observed. Methods and results are presented in this paper.

Key words: I/O performance; CPU transfer rates; overhead CPU; compute-and-test loop.

1. Introduction

Due to the high computational speeds of large scientific computers, I/O rates may be the factor limiting execution speeds of certain application programs. It is desirable, therefore, to

- provide users with criteria for the selection of I/O procedures most suitable for their programs;
- determine how well existing operating systems and I/O routines approach the maximum capabilities of the hardware; and
- learn where improvements might be possible.

For these reasons, a study was undertaken at the Los Alamos Scientific Laboratory (LASL) to investigate I/O performance on the Cray-1 and CDC 7600 computers.

Disk I/O rates, as well as the times during which the CPU was unavailable for computing while I/O was being performed, were measured. The measurements were taken as functions of buffer size and the number of logically independent I/O channels used in performing the operations. The tests were executed on Cray-1 and CDC 7600 computers at the following installations: LASL, Lawrence Livermore Laboratory (LLL), and Cray Research Incorporated (CRI). Sections 2 and 3 describe the methods of measurement and analysis of the resulting data. In Sec. 4, results obtained for various operating systems and I/O routines are discussed.

2. Measurements

The following processes were measured: unformatted reading or writing from and to disk; reading or writing with concurrent computing; and concurrent reading, writing, and computing.

The test programs measured two quantities as functions of buffer size *B* and the number of logically independent I/O channels N used to perform the I/O operations: transfer rates R(B,N) in words per second per channel, and overhead CPU times per sector $T_{OH}(B,N)$. The latter are defined as the times when the CPU is unavailable for computing while a sector (512 words) of data is being transferred to or from disk. Buffer sizes were multiples of 512 computer words, except for BUFFER IN/BUFFER OUT operations for which program buffers were multiples of 511 words. Four logically independent channels were available at all Cray-1 systems at LASL and LLL. The CDC 7600s were equipped with three logically independent channels at LASL, two at LLL.

Each test program reads and/or writes N one-million word files by repeatedly filling (emptying) a program buffer of preset length B. The process is repeated for each buffer size. For several routines, I/O can be performed either sequentially or by choosing disk addresses at random. Rates were measured for single channels in two ways;

(1) The non-overlapped (or synchronous) part of the test called for reading (writing) a buffer and waiting for I/O completion, then repeating this sequence until the entire file was read (written).

(2) The overlapped (or asynchronous) part executed a pretimed compute-and-test loop while waiting for I/O completion. The duration of the compute-and test loops had to be short enough not to slow down I/O operations. For several cases, overlapped rates exceeded the non-overlapped rates considerably, indicating that the system's frequency of testing for I/O completion was not high enough or that the time required to return from the interrupt was too long.

The tests were run on dedicated system time, with other users and system diagnostics blocked out. Great care was exercised to select timing routines that measured wall clock time and that had high enough precision. Experience showed that this was a non-trivial problem. Rates R(B,N) were obtained by dividing the total number of words transferred W (an integer multiple of 512 * B approximately equal to 10^6N) by the measured time T_t required for the transfer and the number of channels N:

$$R(B,N) = \frac{W(B,N)}{T_t(B,n) * N} \quad . \qquad (1)$$

Overhead CPU times $T_{OH}(B,N)$ were measured in the following way: After initiating the transfer of 512 * B words on each of N channels, a compute-and-test loop was started that performed a series of multiplications and then tested each channel for I/O completion. If I/O was complete on any channel, it was reinitiated immediately before the compute-and-test loop resumed. The process was repeated until all files were transferred. The number of times the compute-and-test loop was executed during the complete file transfer N_{loop} was measured as well as the duration of one compute-and-test loop t_{loop}. The overhead CPU time per transfer of 512 words is given by

$$T_{OH}(B,N) = \frac{T_t(B,N) - N_{loop} * t_{loop}}{W(B,N)/512} \quad . \qquad (2)$$

In many cases, the numerator in Eq. 2 is a small difference of two large numbers, i.e., the overhead times are small compared to the transfer times of 512 words. For these cases the resulting values of T_{OH} are extremely sensitive to even small errors in any one of the numerator terms, which therefore had to be measured with high precision. The total transfer times $T_t(B,N)$ were of the order of several seconds and could easily be measured with high accuracy (better than 10^{-6}) by calls to the cycle counter or microsecond clock. The integer loop count N_{loop} is free of error. However, the measurement of the duration of the compute-and-test loop t_{loop} (which ranged from 20 µs to 1000 µs) required special attention. Three calls to the timing routine were made to accurately time and deduct the duration of the timing call itself. All I/O status tests were included in the timed loop, and parameters were set in such a way that the branches of the loop transferred to were the same and that I/O status checking was done in the same way as when I/O was busy. The essential section of code that includes the timing of the compute-and-test loop for each number of channels and the timing of the file transfer overlapped by the compute-and-test loop is given in Appendix A. Compute-and-test loops were timed several times, and occasional skewed values caused by system disturbances were discarded. The remaining values agreed to better than 1 µs. To assure that no systematic errors were overlooked, tests were run with compute-and-test loops of several lengths differing by factors of about 2 for each routine. No systematic deviations were found.

To prevent certain hardware problems on the CDC 7600 caused by accessing the same memory location too often, the calculations were performed on subscripted variables. These loops automatically vectorized on the Cray-1 and were subsequently replaced by scalar loops to obtain longer compute loops. Overhead CPU results obtained from tests run with vector compute-and-test loops on the Cray-1 showed considerably higher overhead than those using scalar arithmetic, due to memory lockout during a vector operation. This indicates the sensitivity of the operations involved. Some tests were run with compute-and-test loops that required no memory access at all. The results were identical to those with loops performing only scalar operations.

Measured values of transfer rates and overhead CPU times were subject to some random fluctuations, which were especially pronounced for ALAMOS, a LASL-produced operating system for the Cray-1. Part of these variations is due to fluctuations of the rotation rate of the disks that is nominally ± 2%.

3. Analysis of Data

The disk units attached to the Cray-1 (DD-19) and the CDC 7600 (819) are very similar. Each unit consists of 40 recording surfaces subdivided into 411 cylinders for recording data. A read-and-write head is associated with each recording surface. The 40 heads are divided into 10 head groups of 4 heads each. The four heads of a group are used in tandem to transfer data to and from disk in such a way that parts of a single computer word will reside on four recording surfaces. During one disk revolution of 1/60 seconds, a head group will pass over and therefore be able to read or write one track of data. For the Cray-1, a track contains 18 sectors; for the CDC 7600, a track contains 20 sectors of 512 computer words each. Switching from head group to head group is accomplished electronically and rapidly; therefore, a maximum of 10 tracks, constituting one cylinder, can be transferred to or from disk without mechanically repositioning the read-and-write heads or missing a disk revolution. Repositioning of the heads is a mechanical and therefore slow process. For sequential access, one disk revolution is missed at each cylinder boundary. This results in a maximum transfer rate of R_{max} of one cylinder per 11 disk revolutions for sequential access of large files extending over many cylinders. For the Cray-1

$$R_{max,Cray} = \frac{180*512 \text{ words}}{11*1/60 \text{ second}} = 502.7 \text{ kword/s}; \quad (3)$$

for the CDC 7600

$$R_{max,7600} = \frac{200*512 \text{ words}}{11*1/60 \text{ second}} = 558.5 \text{ kword/s}. \quad (4)$$

In practice, the maximum transfer rates will not be reached if additional disk revolutions are missed or if the density of data written on the disk is less than optimal. If B sectors are transferred to or from disk per I/O call, M disk revolutions are missed per call in addition to those at cylinder boundaries, and S sectors are transferred per revolution, then the number of disk revolutions N_B needed to transfer B sectors is given by

$$N_B = B/S + B/(10*S) + M \quad (5)$$

The number of disk revolutions per sector $N_{revs}(B) = N_B/B$ is

$$N_{revs}(B) = 1.1/S + M/B \quad , \quad (6)$$

and the associated transfer rate is

$$R(B) = \frac{512 \text{ words}}{N_{revs}(B)*1/60 \text{ second}} = \frac{512*60 \text{ words}}{(1.1/S + M/B) \text{ seconds}} \quad . \quad (7)$$

If more than one logically independent channel is employed for data transfer, Eqs. 3-7 should be applicable to each channel independently.

Equation 6 indicates the number of disk revolutions $N_{revs}(B)$ per sector should be a linear function of 1/B, the reciprocal of the buffer size. To analyze the experimental data, the measured values of $N_{revs}(B)$ were plotted for each I/O routine as a function of 1/B. Most plots were indeed linear, and the constants S and M could be determined from the zero intercept and the slope of each line. As an example, Fig. 1 represents data measured by the routines BUFFER IN/BUFFER OUT on the Cray-1. The number of disk revolutions per sector for BUFFER OUT as plotted as a function of the reciprocal of the buffer size is a straight line, the slope of which corresponds to M = 3; that is, three disk revolutions are missed per I/O call. The zero intercept corresponds to a transfer of 18 sectors per revolution. The data for the BUFFER IN operation can be fitted by two straight lines, with the same zero intercept. For B $\geq$ 18 sectors, two disk revolutions are missed; for B $\geq$ 16 sectors, only one disk revolution is missed per I/O call. These results are interesting but not characteristic of most I/O routines on the Cray-1 and CDC 7600. The most frequently encountered sets of coefficients were M = 0 and M = 1 (one disk revolution missed per I/O call) and S = 18 for the Cray-1 or S = 20 for the CDC 7600 (maximum number of sectors per disk revolution).

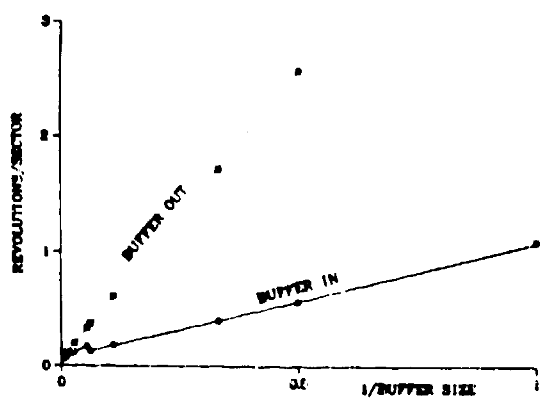

FIGURE 1. Disk revolutions per sector for BUFFER IN/OUT as a function of the reciprocal of the buffer size.

An example of results obtained on the CDC 7600 is shown in Fig. 2. Routines performing random-access I/O transmission were used for this test. The solid line represents data for sequential access and corresponds to one disk revolution missed per I/O call and a transfer of 20 sectors per disk revolution.

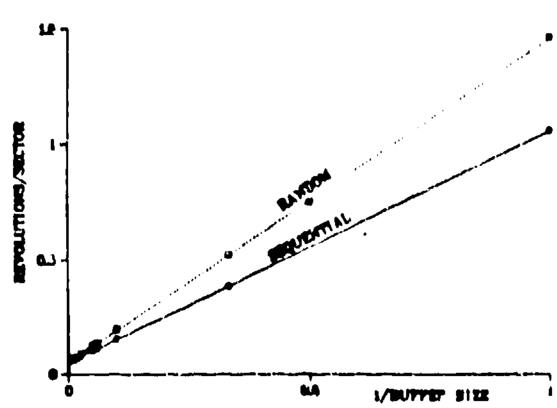

FIGURE 2. Revolutions per sector for transfer by WDISK/RDISK (7600) as a function of the reciprocal of the buffer size.

Using a random number generator to determine the disk addresses at which to start transmissions, the test was performed a second time using the same routines. Data for these results are also fitted by a straight (dotted) line, with the same zero intercept, corresponding to 20 sectors transferred per disk revolution. Due to more frequent seek operations, 1.4 disk revolutions are missed per I/O call.

To interpret results obtained for overhead CPU times, it is reasonable to assume that the overhead CPU time per sector $T_{OH}(B,N)$ consists of two contributions, the CPU time T_{trans} required to actually transfer the data, and 1/B times the CPU time T_{call} needed to initiate and complete the system call for I/O:

$$T_{OH}(B,N) = T_{trans} + 1/B*T_{call} \quad . \qquad (8)$$

Experimental results indicate that the assumptions leading to Eq. 8 are valid in most cases. Plots of $T_{OH}(B,N)$ versus 1/B are straight lines with a zero intercept of T_{trans} and a slope of T_{call}.

Figure 3 shows data obtained for the Cray-1 using random-access I/O routine RDISK The lower curve represents data obtained using a scalar compute-and-test loop, with T_{call} = 660 ± 10 µs and T_{trans} = 6 ± 2 µs. The tests were also run using a vectorizable compute-and-test loop. Because of memory lockouts during the vector calculations, the CPU overhead times are slightly higher, as observed in the upper curve.

Data obtained from CDC 7600 tests is shown in Fig. 4. Both sequential and random tests were performed as previously described. The zero intercept is the same for both tests; the overhead CPU times per sector are slightly larger for the random test due to more frequent seeking.

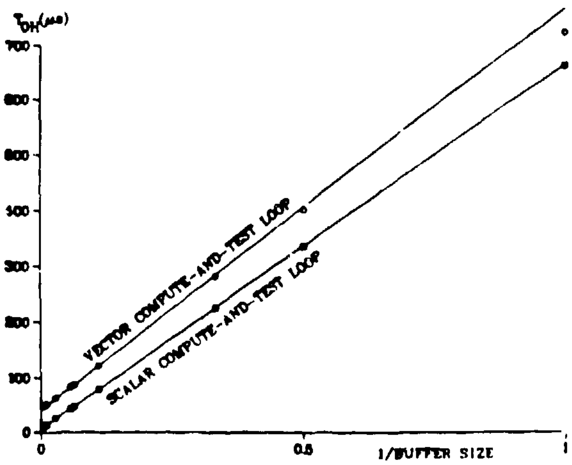

FIGURE 3. Overhead CPU times per sector for RDISK (CFTLIB) as a function of the reciprocal of the buffer size.

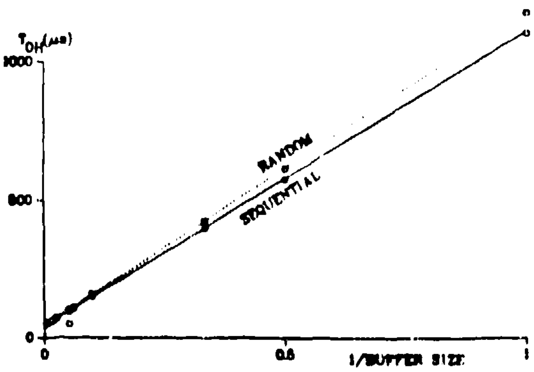

FIGURE 4. Overhead CPU times per sector for WDISK/RDISK (7600) as a function of the reciprocal of the buffer size.

4. Notes on Results

Measurements made on both the Cray-1 and the CDC 7600 employed a wide variety of operating systems, libraries, and I/O routines. Both sequential and random writing/reading, buffered and nonbuffered I/O, Fortran versus assembly language (CAL), and overlapped/non-overlapped tests were run and analyzed.

Tests performed on the Cray-1 used program buffer sizes of 1, 2, 3, 9, 18, 36, 90, and 180 sectors to ensure that any peculiarities occurring at track and cylinder boundaries would be observed.

In almost all cases, overhead CPU times can be represented by Eq. 8. The random access routines IZDKIN/IZDKOUT, RDABS/WRABS, and RDISK/WDISK are all very efficient with very small overhead CPU times. In some cases, the implementation of a very short compute-and-test loop (< 30 μs) ensured that no disk revolutions were missed, even at buffer size B = 1.

Buffered operations were measured using BUFFER IN/BUFFER OUT and BINARY READ/WRITE statements. Some of the tests were run with system buffer sizes of 1/4, 1/2, 1, and 2 times that of the program buffer size to determine the effect on rates. It was apparent that in all cases where this approach was taken, the resultant rates were a direct function of the system buffer size.

Program buffer sizes for tests run on the CDC 7600 were set at 1, 2, 3, 10, 20, 40, 60, 100, and 200 sectors. The results of these tests were generally analogous to those seen on the Cray-1. Overhead CPU times were somewhat larger, with the smallest exhibited by RDISK/WDISK routines with $T_{call} = 1050 \pm 50$ μs and $T_{trans} = 46 \pm 3$ μs. Again, on random access routines, transfer rates are consistent with a transfer of 20 sectors per disk revolution and one disk revolution missed per I/O call. Tests that generated random-disk addresses by a random-number generator showed an increase to 1.4 in the number of disk revolutions missed per I/O system call, due to more frequent and longer seek operations.

For BUFFER IN/BUFFER OUT tests, as with the Cray-1, the rates were solely dependent on the size of the system buffer used in the transfer. For read requests, two revolutions were missed per processing of a system buffer, except for the minimum system buffer size of B = 2 sectors, for which four disk revolutions were missed.

The situation for write operations was more pronounced: four disk revolutions are missed per system buffer processing for B > 2. For B = 2, it was found that nine disk revolutions are missed per call. This is caused by excessively high overhead times associated with this buffer size.

The raw data from these tests can be obtained from the Computer Science and Services Division's Research and Applications Group at the Los Alamos Scientific Laboratory.

5. Summary

Characteristics of all routines tested are summarized in Table 1. The following general observations can be made.

With the exception of some BUFFER IN/BUFFER OUT routines, the maximum number of sectors is transferred per disk revolution: 18 for the Cray-1, 20 for the CDC 7600.

For the Cray-1, maximum transfer rates of 503 kword/s for both overlapped and non-overlapped reads and writes were achieved on the COS operating system. The overhead CPU time for an I/O system call T_{call} = 480 μs is smallest among the operating systems tested; the CPU time required for the transfer of one sector T_{trans} = 42 μs is slightly higher than that observed on the CTSS operating system.

The CTSS operating system has library routines that achieve maximum transfer rates for both overlapped and non-overlapped writes. The reads are more sensitive to proper timing of I/O requests than the writes. Non-overlapped reads always miss one disk revolution per call. For overlapped I/O, using the most efficient routines available, maximum transfer rates can be achieved only by testing I/O completion at < 100-μs intervals. Observed overhead CPU times are T_{call} = 650 μs for the I/O system call and T_{trans} = 7 μs for a one-sector transfer. On a single channel, these are short enough not to degrade I/O performance; for multichannel tests and small buffer sizes, some rate degradation will occur.

At least one disk revolution is missed per I/O call for all routines on the ALAMOS operating system. This is attributable to the high overhead CPU time of about 4000 μs for the I/O call. This is considerably higher than the 926 μs required for one sector to pass under the read/write head.

CDC 7600/LTSS operating system routines also miss at least one disk revolution per I/O call. The minimum overhead CPU times for initiating an I/O call are T_{call} = 1050 μs for RDISK/WDISK routines, just slightly longer than the 833 μs required for one sector to pass under the read/write head. The minimum measured transfer CPU time per sector is T_{trans} = 46 μs.

It is apparent that maximum I/O rates can be achieved only if one chooses I/O routines carefully and performs frequent enough testing for I/O completion on small buffer sizes.

Table 1

Performance Characteristics of Cray-1 and CDC 7600 I/O Routines

Machine	Operating System	Routine	Library	Sectors per disk revolution	Disk revolutions missed per I/O call	T_{call} (us)	T_{trans} (us/sector)
CRAY-1	CTSS	IZDKIN	BASELIB	18	0 or 1[a)]	630	6
		IZDKOUT		18	0	630	10
		RDISK	CFTLIB	18	0 or 1[a)]	660	6
		WDISK		18	0	660	6
		RDABS	FORTLIB	18	0 or 1[b)]	650	7
		WRABS		18	0	650	9
		BUFFER IN	CFTLIB	18	1 or 2	1580	50
		BUFFER OUT		18	3	14000	150
		BUFFER IN	FORTLIB	18	1	---	---
		BUFFER OUT		18	0 or 1	---	---
		READ	CFTLIB		0 or 1	730	10
		WRITE			0	730	10
		READ	FORTLIB	18	≥ 1	---	---
		WRITE		18	2	---	---
	ALAMOS	System calls (read)		18	1[c)]	4050[c)]	155[c)]
		System calls (write)		18	1[c)]	4050[c)]	155[c)]
		BUFFER IN		9	2	---	---
		BUFFER OUT		9	1	---	---
	COS	System calls (read)		18	0	480	42
		System calls (write)		18	0	480	42
CDC 7600	LTSS	IZDKIN	BASELIBF	20	1	1600	50
		IZDKOUT		20	1	1700	50
		RDISK	FTNLIB	20	1	1050	46
		WDISK		20	1	1050	46
		BUFFER IN	FTNLIB	20	2	~1750[d)]	~100[d)]
		BUFFER OUT		20	4	*[d)]	*[d)]
		READ	FTNLIB	20	1	*[d)]	*[d)]
		WRITE		20	1	*[d)]	*[d)]
		BUFFER IN	ORDERLIB	20	5	---	---
		BUFFER OUT		20	2	---	---

a) Depending on length of compute-and-test loop for overlapped I/O. Always 1 for non-overlapped I/O.
b) 0 for B<24, 1 for B≥36.
c) for B≥9
d) Equation 3.5 does not apply.

Appendix A

Example of Code for Timing of the Compute-and-Test Loop
and File Transfer

```
C      SET NUMBER OF CHANNELS
       DO 1000 NCHAN=1,MAXCH
            ICOMP=0
            DO 110 N=1,NCHAN
                 NEXT(N)=0
                 DONE(N)=.TRUE.
110              CONTINUE
            CALL IDLE
C
C      TIME COMPUTE-AND-TEST LOOP FOR EACH NUMBER OF CHANNELS
           T0=TIMEF(DUM)
           T1=TIMEF(DUM)
C
115        CALL COMPUTE(NCOMP)
           ICOMP=ICOMP+1
C
           DO 160 N=1 NCHAN
                IOC=N+4
                IND=DSP(4,N)
                IF((IFTBL(3,IND).GE.0).OR.DONE(N))GO TO 160
                IF(NEXT(N).LT.MAX) GO TO 150
                DONE(N)=.TRUE.
                GO TO 160
150             CALL RDISK(IOC,BUFF,NWDW,NEXT(N)*NWDW)
                NEXT(N)=NEXT(N)+1
160             CONTINUE
C
           ALDONE=.TRUE.
           DO 170 N=1,NCHAN
170             ALDONE=ALDONE.AND.DONE(N)
           IF (.NOT.ALDONE) GO TO 115
C
           T2=TIMEF(DUM)
           TLOOP(NCHAN)=1000.*(T2-2*T1+T0)
C
C
C      SET BUFFER SIZE, ARRAY ISECT CONTAINS BUFFERSIZES
            DO 900 NBF=1,9
                 NWDW=ISECT(NBF)*512
                 MAX=NWDS/NWDW
                 ICOMP=0
                 DO 310 N=1,NCHAN
                      NEXT(N)=0
                      DONE(N)=.FALSE.
310                   CONTINUE

                 CALL IDLE
C
C
C
C      GET TIMINGS FOR OVERLAPPED I/O
C
                T0=TIMEF(DUM)
C
                GO TO 355
340             CALL COMPUTE(NCOMP)
                ICOMP=ICOMP+1
C
```

```
C      TEST EACH CHANNEL WHETHER I/O BUSY, IF NOT REINITIALIZE
355                DO 360 N=1,NCHAN
                          IOC=N+4
                          IND=DSP(4,N)
C                         IF CHANNEL N BUSY OR DONE GO TO 360
                          IF((IFTBL(3,IND).LT.0).OR.DONE(N)) GO TO 360
                          IF(NEXT(N).LT.MAX) GO TO 357
                          DONE(N)=.TRUE.
                          GO TO 360
357                       CALL RDISK(IOC,BUFF,NWDW,NEXT(N)*NWDW)
                          NEXT(N)=NEXT(N)+1
360                       CONTINUE
C
C      TEST FOR FINAL I/O COMPLETION
                   ALDONE=.TRUE.
                   DO 370 N=1,NCHAN
370                       ALDONE=ALDONE.AND.DONE(N)
                   IF (.NOT.ALDONE) GO TO 340
C
                   T1=TIMEF(DUM)
                   TIO(NCHAN,NBF)=1000.*(T1-T0)
C
900                CONTINUE
1000         CONTINUE
```