May 7, 1959

MEMO TO: Mr. H. G. Jones

SUBJECT: Binary Serial Files

Attached is a comparison of the 3 unit and 4 unit binary serial files. It attempts to answer some of the questions you raised in our discussion last week.

My conclusion, that the 3 unit file should be pushed, rests mainly on the relative speed with which such a file could be produced. Satisfaction of SABER requirements is the main objective.

Your comments on the comparison and the possible value of the modal approach would be appreciated.

E. W. Coffin

E. W. Coffin
Planning Representative
Project 7000

EWC:jcj
Attachment
cc: Mr. D. W. Pendery
Mr. D. W. Sweeney
Dr. H. G. Kolsky ←
Mr. W. J. Vitek
Mr. J. R. Lyons

Comparison of Two Types of Binary Serial File

I Physical Characteristics

	A	B
Modules or units per file	3	4
Disk plates per unit	22	17
Disks/module for information bits	21 or 22	16
Total information disks/file	64	64
Total disk plates/file	66	68
Tracks per module	5376 or 5632	4096

File rotation		1800 rpm
File rotation time		33-1/3 ms/rev.
Tracks per disk plate		256
Total tracks per file		32,768
Information bits per track	approx.	18,432
704-709-7090 words/track		512
STRETCH (ECC mode)words/track		256
Total words per file (7090)		16,777,216
Total words per file (STRETCH)		8,388,608
Data Transmission rate (72 bit word)		144 us

II Addressing Characteristics

Assuming 15 bits used for this purpose

File A 8 bits vertical cylinder, 6 bits access arm, 1 bit head

File B 8 bits vertical cylinder, 2 bits access unit, 4 bits access arm and 1 bit head

The attached figures 1 (A) and 1 (B) show the physical picture of these files. Figures 2 (A) and 2 (B) show the arrangement, by unit or module, of the access arm for each file.

III Modal Operation

The physical nature of a disk file allows partially random logical usage. Two levels of randomness can be achieved. These lie between the strictly sequential logical nature of tapes and the fully random logical nature of core memory.

Using disk files a modified sequential transmission of long streams of data is possible. Also random access to the start of a long block of information is possible. Once started the transmission of information should be continuous. Mechanically an access mechanism will be set to the track specified by the address. Thereafter, data will be sent from successive tracks. The entire disk file may be considered as a set of 256 concentric cylinders standing on end. Each cylinder has 128 tracks of information in the vertical plane. To achieve sequential data transmission succeeding access mechanism are positioned to the same cylinder until we go from the last to the first. Here the access mechanism steps in to the next cylinder. By this process we may step through all the modules of the file with effectively no track to track access.

The other level of randomness is obtained by being able to suppress the automatic access mechanism sequencing. Two logical concepts of independent random accessing should be considered. First is random accessing over the contents of the entire file. Second is random access within particular modules of the file. The first type of accessing occurs with Project SABER requirements, storage of libraries of subroutines, data tables, inventory information, distribution accumulation processes. The second type occurs when one module may be used for Reading and another for Writing. Independent modules may be used to collect particular types of information for subsequent processing. This type of accessing is the basic nature of the Advanced Disk File. The nature of this type of randomness makes cylinder to cylinder access gaps unavoidable within the references to a given unit. This problem may be solved by providing multiple access arms for each module. However, the nature of the control required at the program level, makes track addressing for each track used, a feasible requirement.

Therefore, a modal form of operation is proposed for binary serial files. The mode should be alterable by programming.

Sequential Mode - The address will cause the access mechanism specified to be set at the cylinder specified by the high order 8 bits. The low order seven bits will specify the arm and head to be used for the start of data transmission. Once started data flow will be continuous in the usual sense applicable to the using system. i. e., 704 - data continues until "copy" orders are suspended. For 709-7090 and STRETCH the data flow continues until broken by the control word processing.

Random Mode - Each access mechanism will act independently of any other access mechanism. Continuous data flow will occur for a maximum of one track's worth of information. Separate addresses will be supplied for each track from which information is desired.

It is worth pointing out that separate blocks of cylinders could be used independently with respect to mode. Thus, for example, the first n cylinders may be used sequentially while the next block might be considered as 3 m cylinders, m for each access mechanism. This would imply a higher level of functional utility than is possible with the standard Advancdd Disk File.

IV Programming Constraints

Ultimately the usefulness of any disk file will be determined by the amount and complexity of the programming required to use it. Two distinct conditions of use must be considered. The first is the use of unique requests i. e., a new request will be made only after the previous request has been completed. Regardless of the mode of operation the use of the straight forward binary address developed by the program is handled only by the access decoder. No program problems occur. The second condition is the occurrance of random requests which must be analyzed by the program before being processed. One or two levels of control must be considered depending on the mode of operation.

Sequential mode - The new request must be "stacked" in sequence and a subroutine serves to start a new disk reference when the old request is completed. For the control program a "bit" is required to indicate that the file is busy or not busy. A control count indicates the number of requests already in the stack. This must be increased as requests are added and diminished as they are serviced. For 7030 systems such a program consists of manipulation of a control word coupled with an occasional "transmit" to shift the latest requests downward. For 709-7090 systems a program of approximately 16 instructions, of which either 6 or 12 are executed is required.

Random mode - Random requests developed while operating in the random mode appear at first glance to make the 3 unit file hard to use. The following four programs indicate that the steps required to decode for either the 3 unit or 4 unit file are of equal magnitude.

These programs operate on the bits marked "PLATE" in figure 2 (A) for the 3 unit file. They operate on the bits marked "Unit" in figure 2 (B) in the 4 unit file. Each program leads to a set of control subroutines determined by the number of units in the file. Each control subroutine consists of approximately 16 steps as indicated in the earlier discussion.

3 Unit File 1st Program Required 1 index register

```
ANA   MASK                              MASK = 000000001111110
ARS 1
PAX 0,1
TXL   ARM 1, 1, 20                        Value of 6 bits    20 to Mech. 1.
TXL ARM 3, 1, 42                          6 bits     43  to Mech. 3
TRA   ARM 2                               21    6 bits    42   to Mech. 2.
```

3 Unit File 2nd Program No index register

```
ANA   MASK                              MASK = 000000001111110
SUB CI                                  CI = 0101001
TMI   ARM 1
SUB  C2                                 C2 = 0101100
TPL   ARM 3
TRA   ARM 2
```

4 Unit File 1st Program Using 1 index register

```
ANA   MASK                              MASK = 000000001100000
ARS    5
PAX    0,1
TXL    ARM 1, 1,0
TXL    ARM 2,1,1
TXL    ARM 3,1,2
TRA    ARM 4
```

4 Unit File   2nd Program   Using no index register

```
ANA    MASK                      MASK = 000000001100000
TZE    ARM 1
SUB    CI                        CI = 1000000
TMI    ARM 2
TZE    ARM 3
TRA    ARM 4
```

V Construction

Decoding schemes for the hardware of the 4 unit file are straightforward. The corresponding functions for the 3 unit file require a slight increase in the amount of hardware required. A review of proposed decoding schemes, by Mr. John Lyons of Project 7000 Engineering, indicates that about 40 transistors are required for access mechanism decoding. Current switching alloy junction transistors were the transistor type considered. These are already being used in the Advanced Disk File. On a full file this adds about 70-88 transistors to existing electronic controls.

Discussions between Mr. Lyons and San Jose representatives have brought out a possible problem. This is the number of disks which can be physically contained in a file. The present upper bound is 63 disk plates but this point should be more fully explored. The 3 unit file would require 66 disks and the 4 unit file; 68 disks. The 22 arm access mechanism seems feasible. The 16 arm access mechanism might present manufacturing difficulties.

The binary hydraulic adder should be available since it is being developed for the STRETCH High Speed File which will be delivered in late 1959. Therefore it would appear feasible to construct a 3 unit binary serial file to meet SABER deadlines.

VI Applications

Binary Serial Files will provide the compact large capacity auxiliary storage required for storing large blocks of tabular and library information. Requirements for this type of storage can be found in wind tunnel data reduction problems and weights and parts listings requirements for aircraft manufacturers. Large scale matrices associated with operations research problems, Large mesh arrays useful in some partial differential equation procedures, gravometric studies and magnatometer calculations from petroleum calculations should find this type of storage desirable.

Basically the ability to obtain blocks of information randomly plus the ability to alter and rewrite elements of a long file of information is vital to scientific computation procedures. Programming systems such as "Monitor", "Mock-Donald" etc., would find the random accessibility of library and master system subroutines a major source of operational improvement. Where possible, "SPOOL" techniques would provide an equivalent operation to "card to tape" permitting selective program processing on a relative priority free from the constraint of tape sequencing. Other "SPOOL" type operations would permit one file, 3 units, to handle reading cards, punching cards, and printing from one device as opposed to preparing relatively short tapes and manually transferring to other tape drives for processing.

VII Conclusions

Using the Advanced Disk File on binary systems with decimal addressing seems like a very poor functional concept. STRETCH has the capability of directly manipulating decimal data but this does not make it desirable. Decimal addressing would appear to require major alterations of the 709-7090 instruction set.

While the 4 unit binary serial has many desirable properties, development of the 3 unit file seems to be more feasible. This feasibility stems from the lesser number of changes from the Advanced File program. The modal operation proposed will result in increased flexibility of disk usage, hence increased usefulness. The potential market area of about 120 files justifies the effort required.

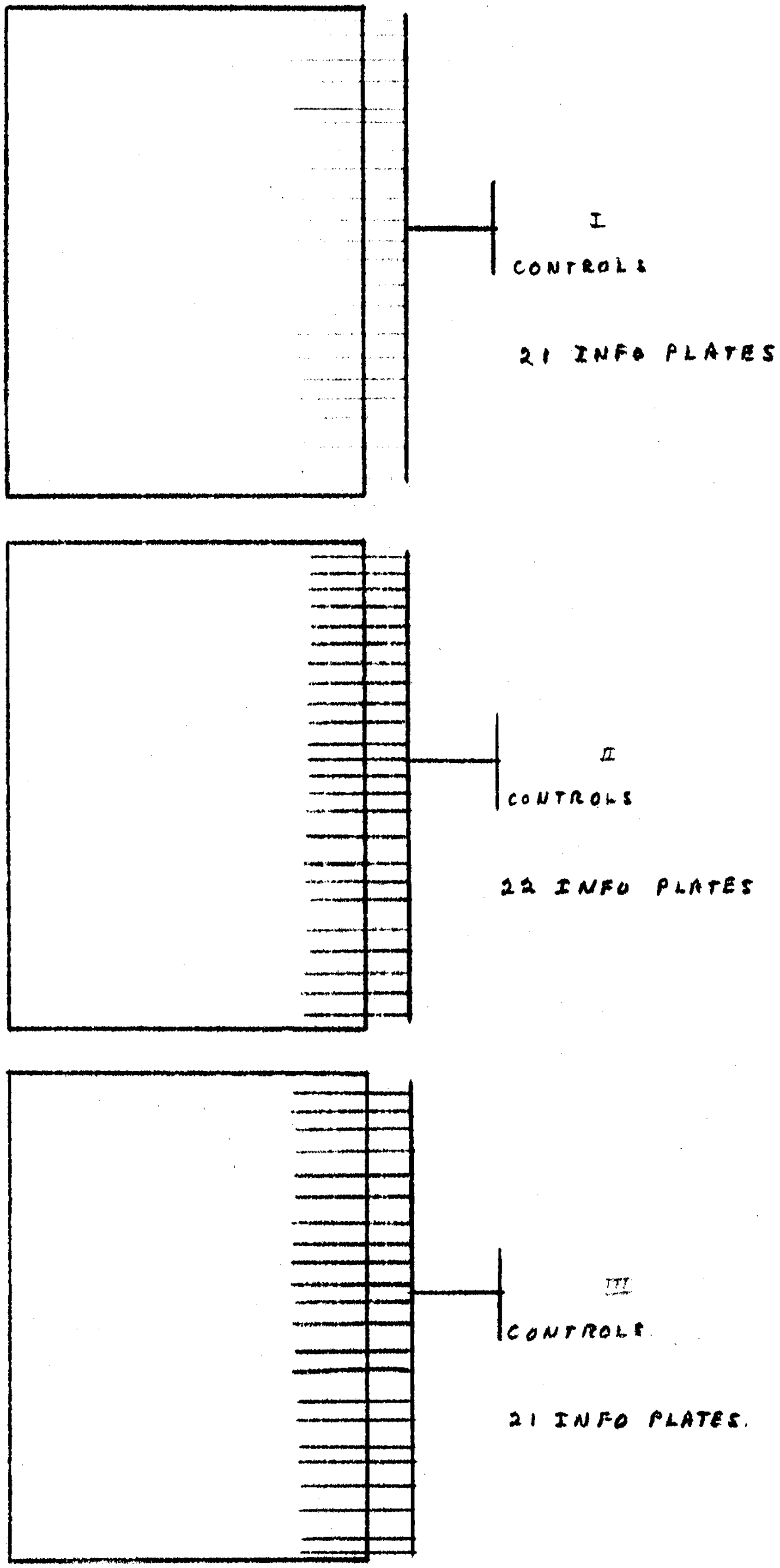

FIGURE 1(A)

EWC 5/1/59

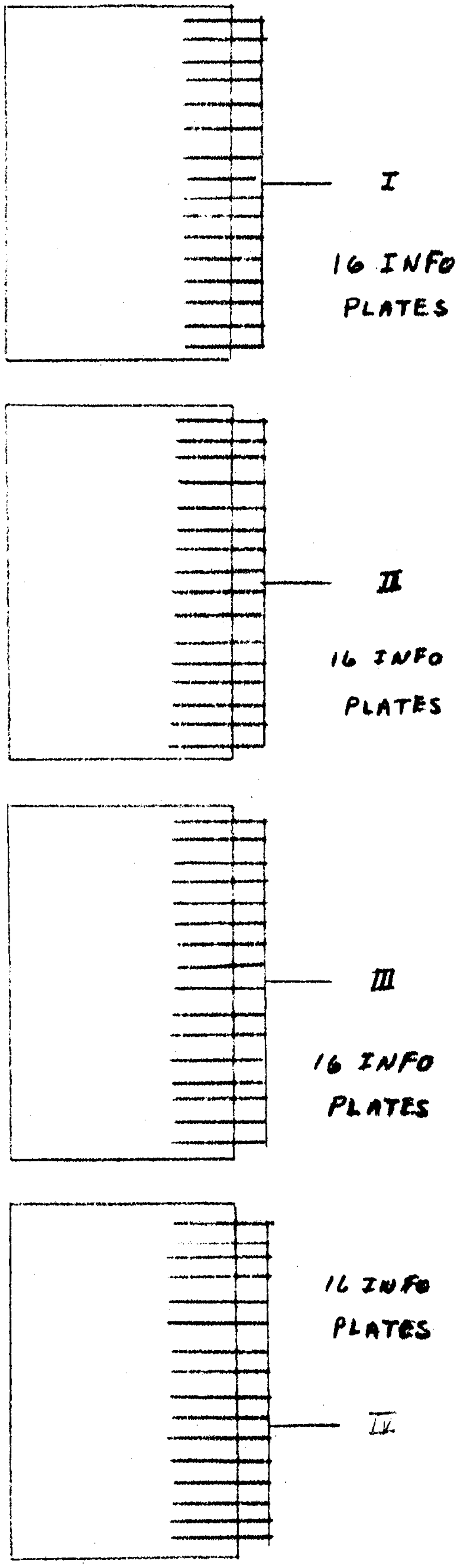

FIGURE 1 (B)

Ewc 5/1/59

PLATE ADDRESSES

I	II	III
0 0 0 0 0 0	0 1 0 1 0 1	1 0 1 0 1 1
0 0 0 0 0 1	0 1 0 1 1 0	1 0 1 1 0 0
0 0 0 0 1 0	0 1 0 1 1 1	1 0 1 1 0 1
0 0 0 0 1 1	0 1 1 0 0 0	1 0 1 1 1 0
0 0 0 1 0 0	0 1 1 0 0 1	1 0 1 1 1 1
0 0 0 1 0 1	0 1 1 0 1 0	1 1 0 0 0 0
0 0 0 1 1 0	0 1 1 0 1 1	1 1 0 0 0 1
0 0 0 1 1 1	0 1 1 1 0 0	1 1 0 0 1 0
0 0 1 0 0 0	0 1 1 1 0 1	1 1 0 0 1 1
0 0 1 0 0 1	0 1 1 1 1 0	1 1 0 1 0 0
0 0 1 0 1 0	0 1 1 1 1 1	1 1 0 1 0 1
0 0 1 0 1 1	1 0 0 0 0 0	1 1 0 1 1 0
0 0 1 1 0 0	1 0 0 0 0 1	1 1 0 1 1 1
0 0 1 1 0 1	1 0 0 0 1 0	1 1 1 0 0 0
0 0 1 1 1 0	1 0 0 0 1 1	1 1 1 0 0 1
0 0 1 1 1 1	1 0 0 1 0 0	1 1 1 0 1 0
0 1 0 0 0 0	1 0 0 1 0 1	1 1 1 0 1 1
0 1 0 0 0 1	1 0 0 1 1 0	1 1 1 1 0 0
0 1 0 0 1 0	1 0 0 1 1 1	1 1 1 1 0 1
0 1 0 0 1 1	1 0 1 0 0 0	1 1 1 1 1 0
0 1 0 1 0 0	1 0 1 0 0 1	1 1 1 1 1 1
	1 0 1 0 1 0	

15 8	7 2	1
CYLINDER OR TRACK	PLATE	U L

15 bit ADDRESS

FIGURE 2 (A)

Ewc 5/1/59

I

0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	0	1
0	0	1	1	1	0
0	0	1	1	1	1

II

0	1	0	0	0	0
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	0	1
0	1	0	1	1	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	0	1
0	1	1	0	1	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	0	1
0	1	1	1	1	0
0	1	1	1	1	1

III

1	0	0	0	0	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	0	1
1	0	0	1	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	0	1
1	0	1	1	1	0
1	0	1	1	1	1

IV

1	1	0	0	0	0
1	1	0	0	0	1
1	1	0	0	1	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	0	1
1	1	0	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	0	1
1	1	1	1	1	0
1	1	1	1	1	1

15 8	7 6	5 2	1
CYLINDER OR TRALK.	UNIT	ARM	H

FIGURE 2 (B)

Ewc 5/1/59