United States Patent

[11] 3,587,057

[72] Inventor **Philip N. Armstrong**
17331 Keegan Way, Santa Ana, Calif. 92705

[21] Appl. No. **830,225**
[22] Filed **June 4, 1969**
[45] Patented **June 22, 1971**

[54] **DATA SORTING SYSTEM**
14 Claims, 40 Drawing Figs.

[52] **U.S. Cl.** **340/172.5**
[51] **Int. Cl.** **G06f 7/22**
[50] **Field of Search** 340/172.5

[56] **References Cited**

UNITED STATES PATENTS

3,015,089	12/1961	Armstrong	340/172.5
3,029,413	4/1962	O'Connor et al.	340/172.5
3,399,383	8/1968	Armstrong	340/172.5

Primary Examiner—Paul J. Henon
Assistant Examiner—R. F. Chapuran
Attorney—Jessup & Beecher

ABSTRACT: A data sorting system for multibit binary records is provided which is capable of responding to a control field in each of the records in order to sort the records into an ascending or descending progression. The sorting system to be described has the feature that it is capable of sorting a file of the aforesaid records which contains more records than the capacity of the sorting system itself. The system is controlled so that the file is repeatedly circulated therethrough until a complete sort is achieved, with the capacity of the system being reduced for each successive pass so that optimum efficiency is maintained at all times.

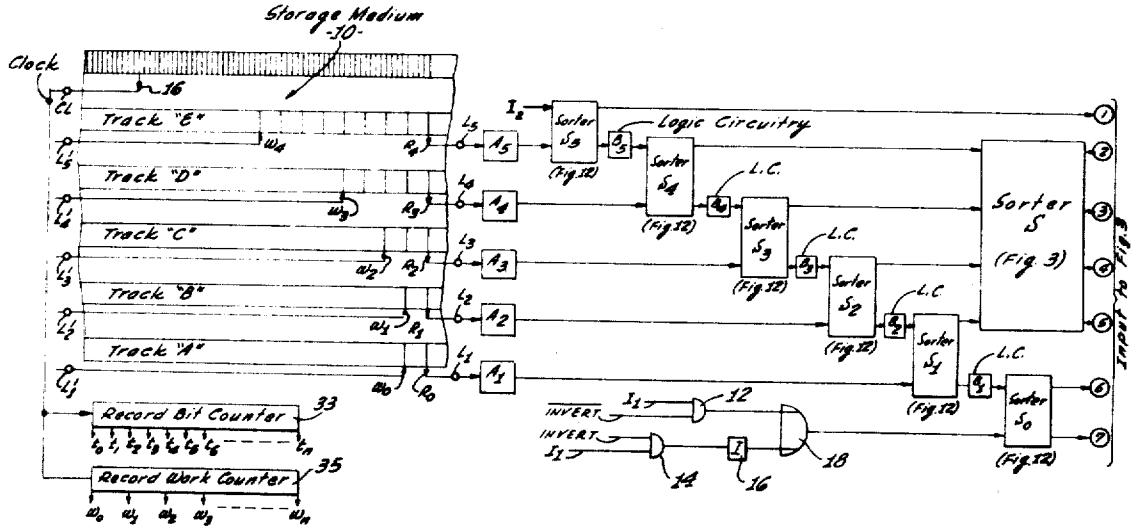


Fig. 1 (Record Format)

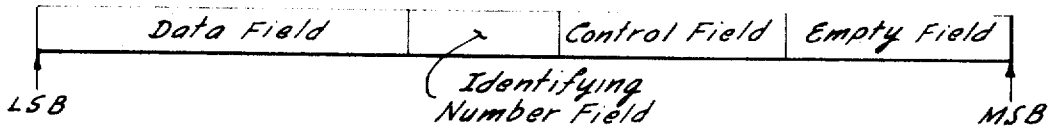


Fig. 2B

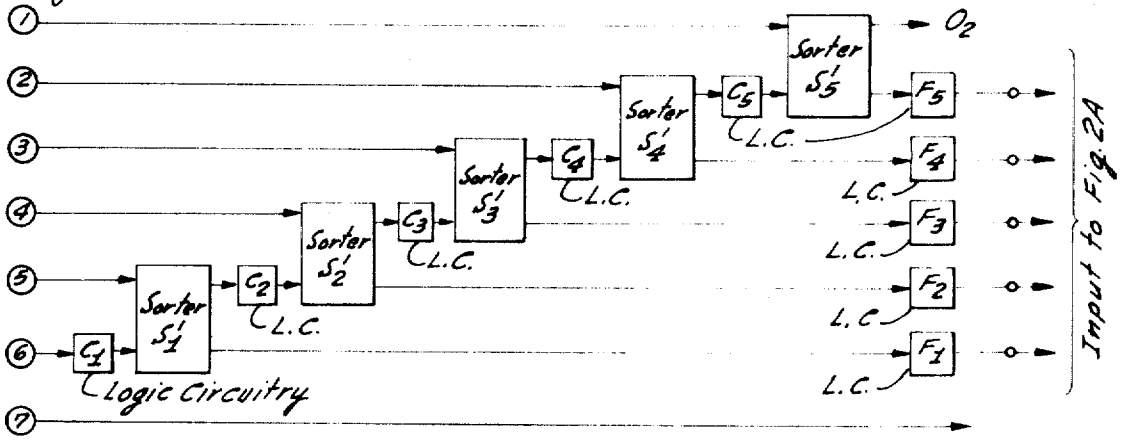
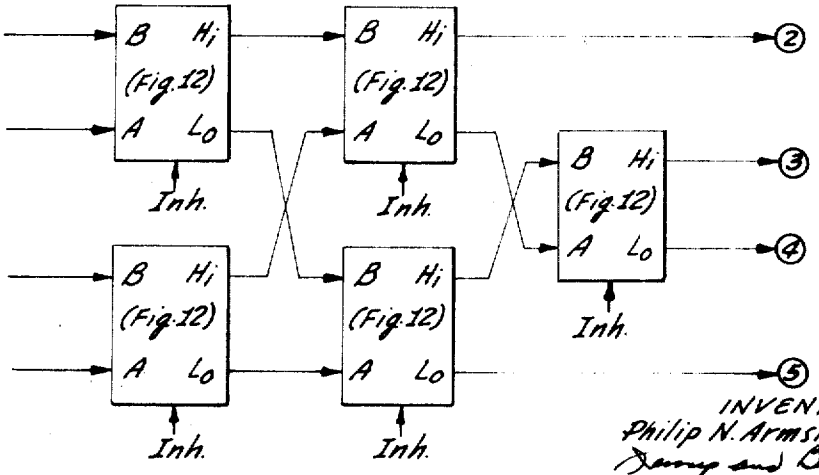


Fig. 3 (Sorter S)



INVENTOR:
Philip N. Armstrong
Dennys and Beadle
By Keith D. Beadle
ATTORNEYS

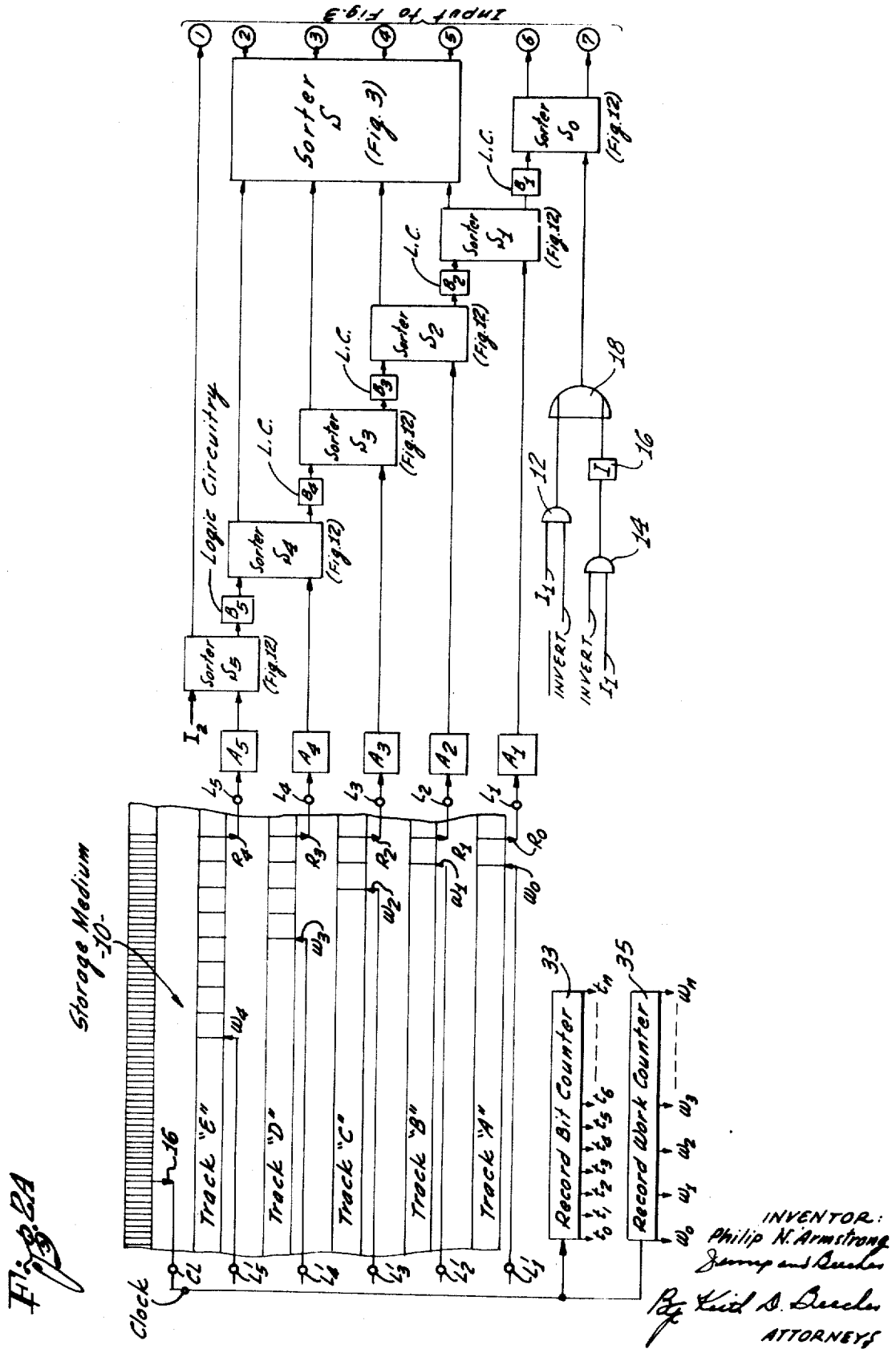


Fig. 2A

INVENTOR:
Philip N. Armstrong
Jessup and Beecher
By Reid A. Beecher
ATTORNEYS

Fig. 4

Record Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Input (I_1)	13	8"	16	10	Z	8"	4	3	7	8"	8"	8"	1	2	5	6	12	11	Z	Z	Z	8
Input (I_2)	Z	Z	Z	Z	Z	Z	8"	Z	Z	Z	Z	Z	8"	8"	Z	Z	Z	Z	Z	Z	Z	Z
Output (O_1)	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"	8"
Output (O_2)	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z

Fig. 5

	Memory				
STEP	A	B	C	D	E
1	13"	X	X X	X X X X	X X X X X X X X
2	13"	X	X X	X X X X	X X X X X X X X
3	16	13"	X X	X X X X	X X X X X X X X
4	13	16	10" X	X X X X	X X X X X X X X
5	Z	13	16 10"	X X X X	X X X X X X X X
6	Z	10	13 16	X X X X	X X X X X X X X
7	4	10	16 13	X X X X	X X X X X X X X
8	4	10	13 16	3" X X X	X X X X X X X X
9	7	10	16 13	4" 3" X X	X X X X X X X X
10	Z	7	10 16	13 4" 3" X	X X X X X X X X
11	Z	Z	7 10	16 13 4" 3"	X X X X X X X X
12	Z	Z	3 7	10 16 13 4"	X X X X X X X X
13	Z	1	4 5	7 10 16 13	X X X X X X X X
14	1	2	3 4	16 10 7 13	X X X X X X X X
15	2	4	5 3	16 13 7 10	1" X X X X X X X
16	3	4	6 5	10 16 13 7	2" 1" X X X X X X
17	4	5	7 6	12 10 16 13	3" 2" 1" X X X X X
18	5	6	11 7	13 12 10 16	4" 3" 2" 1" X X X X
19	Z	6	7 11	16 13 12 10	5" 4" 3" 2" 1" X X X
20	Z	Z	6 7	10 16 13 12	11 5" 4" 3" 2" 1" X X
21	Z	Z	Z 6	7 10 16 13	12 11 5" 4" 3" 2" 1" X
22	Z	Z	6 Z	8 7 10 16	13 12 11 6" 4" 3" 2" 1"

INVENTOR
Philip N. Armstrong
Jerry and Burke
By Kurt D. Burke
ATTORNEYS

Fig. 6

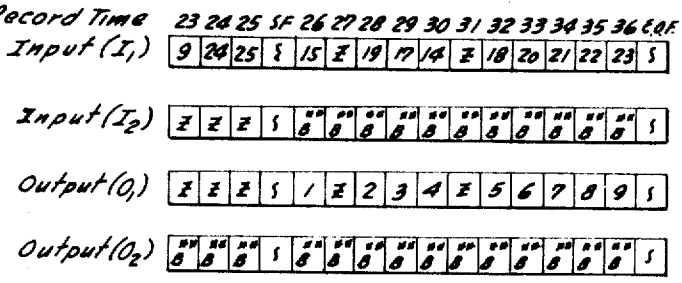


Fig. 7

STEP	Memory A	B	C	D	E
23	Z	Z	1 6	9 8 7 10	16 13 12 11* 5* 4* 3* 2*
24	Z	2	6 1	10 9 8 7	24 16 13 12 11* 5* 4* 3*
25	1	2	3 6	7 10 9 8	25 24 16 13 12 11 5* 4*
Sorter Output Begins					
26	2	4	6 3	8 9 10 9	15 25 24 16 13 12 11 5*
27	2	3	4 6	5 8 7 10	9 15 25 24 16 13 12 11
28	3	6	10 4	11 5 8 7	19 9 15 25 24 16 13 12
29	4	6	7 10	12 11 5 8	17 19 9 15 25 24 16 13
30	6	8	10 7	13 12 11 5	14 17 19 15 25 24 16 13
31	5	6	7 10	8 13 12 11	16 14 17 19 9 15 25 24
32	6	10	11 7	18 8 13 12	24 16 14 17 19 9 15 25
33	7	10	12 11	20 18 5 13	25 24 16 14 17 19 9 15
34	10	11	13 12	15 20 18 8	21 25 24 16 14 17 19 9
35	9	10	11 13	12 15 20 18	22 21 25 24 16 14 17 19
36	10	13	18 11	11 12 15 20	23 22 21 25 24 16 14 17

INVENTOR:
Philip N. Armstrong
Dorsey and Beadles

By Keith D. Beadles
ATTORNEYS

Fig. 8

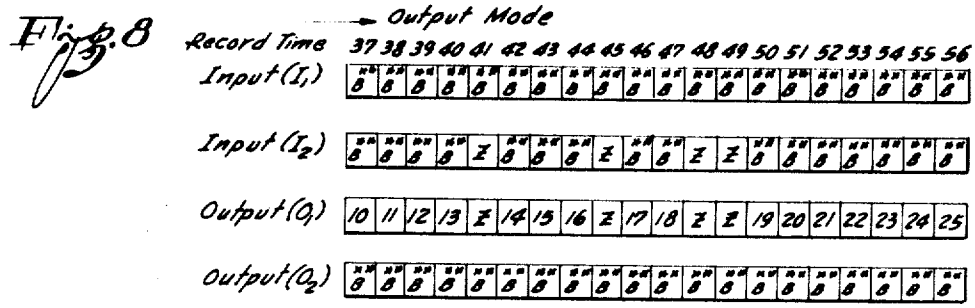


Fig. 9

	Memory				
STEP	A	B	C	D	E
37	11	13	17 18	20 19 12 15	X 23 22 21 25 24 16 14
38	13	14	15 17	18 20 19 12	X X 23 22 21 25 24 16
39	13	14	16 15	17 18 20 19	X X X 23 22 21 25 24
40	14	15	19 16	24 17 18 20	X X X X 23 22 21 25
41	14	15	16 19	20 24 17 18	25 X X X X 23 22 21
42	15	18	19 16	21 20 24 17	X 25 X X X X 23 22
43	16	17	18 19	22 21 20 24	X X 25 X X X X 23
44	17	19	23 18	24 22 21 20	X X X 25 X X X X
45	17	18	19 23	20 24 22 21	X X X X 25 X X X
46	18	21	23 19	X 20 24 22	X X X X X 25 X X
47	19	21	22 23	X X 20 24	X X X X X X 25 X
48	19	21	23 22	24 X X 20	X X X X X X X 25
49	19	20	21 23	22 24 X X	25 X X X X X X X
50	20	23	X 21	X 22 24 X	X 25 X X X X X X
51	21	23	X X	X X 22 24	X X 25 X X X X X
52	23	24	X X	X X X 22	X X X 25 X X X X
53	23	24	X X	X X X X	X X X X 25 X X X
54	24	X	X X	X X X X	X X X X X 25 X X
55	X	X	X X	X X X X	X X X X X X 25 X
56	X	X	X X	X X X X	X X X X X X X 25

INVENTOR:
Philip N. Armstrong
Jerome and Baudler
By Keith D. Baudler
ATTORNEYS

Fig. 10 (Sorting Network)

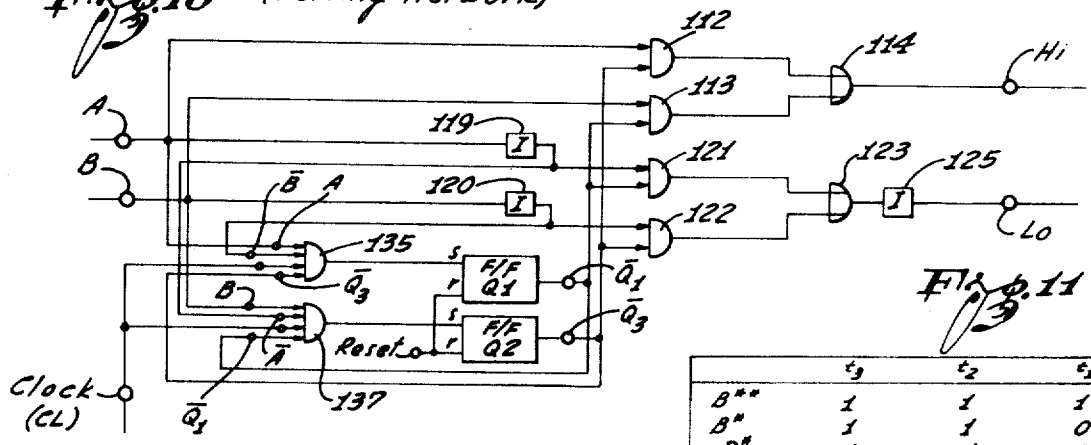


Fig. 11

	t_3	t_2	t_1
B^{***}	1	1	1
B^{**}	1	1	0
D^{**}	1	1	0
B^{*}	0	1	0
D^{*}	0	1	0
D	1	0	0
Z	0	0	0

Fig. 12

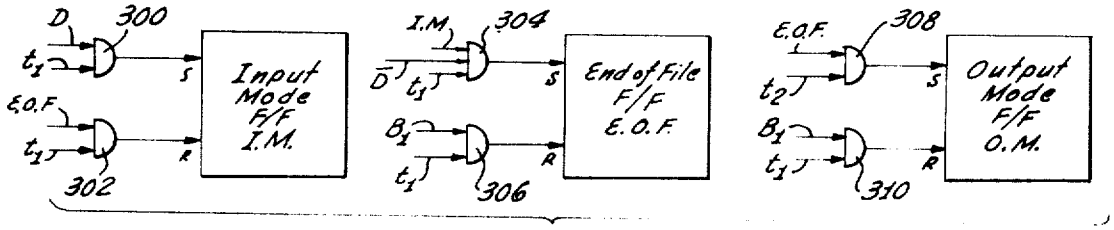
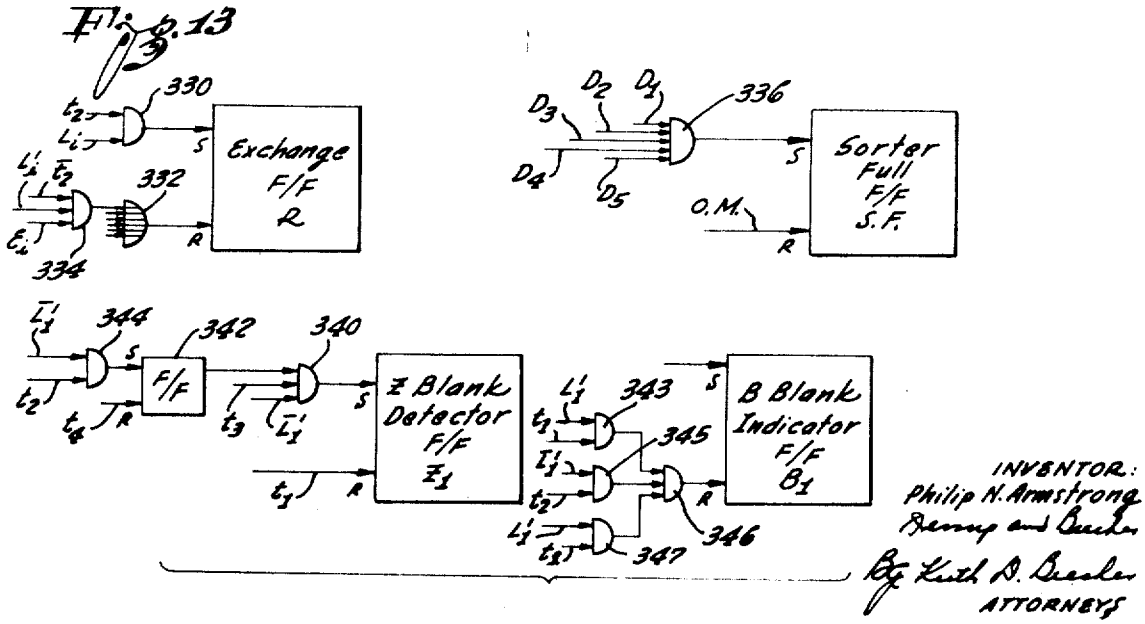
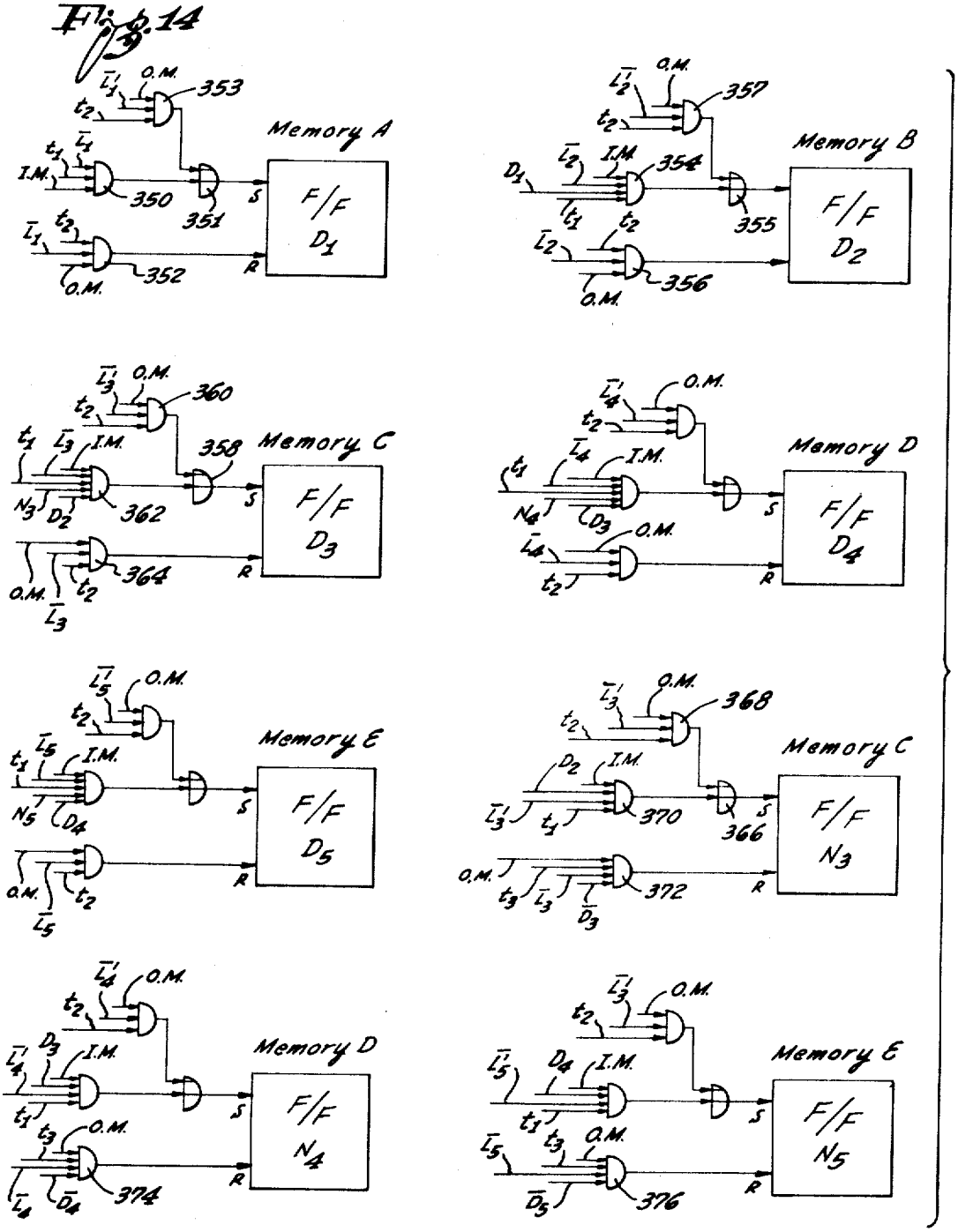


Fig. 13



INVENTOR:
Philip N. Armstrong
Dennis and Bueler
By Keith D. Bueler
ATTORNEYS

Fig. 14



INVENTOR:
Philip N. Armstrong
Dennison and Meserole
By Kurt A. Meserole
ATTORNEYS

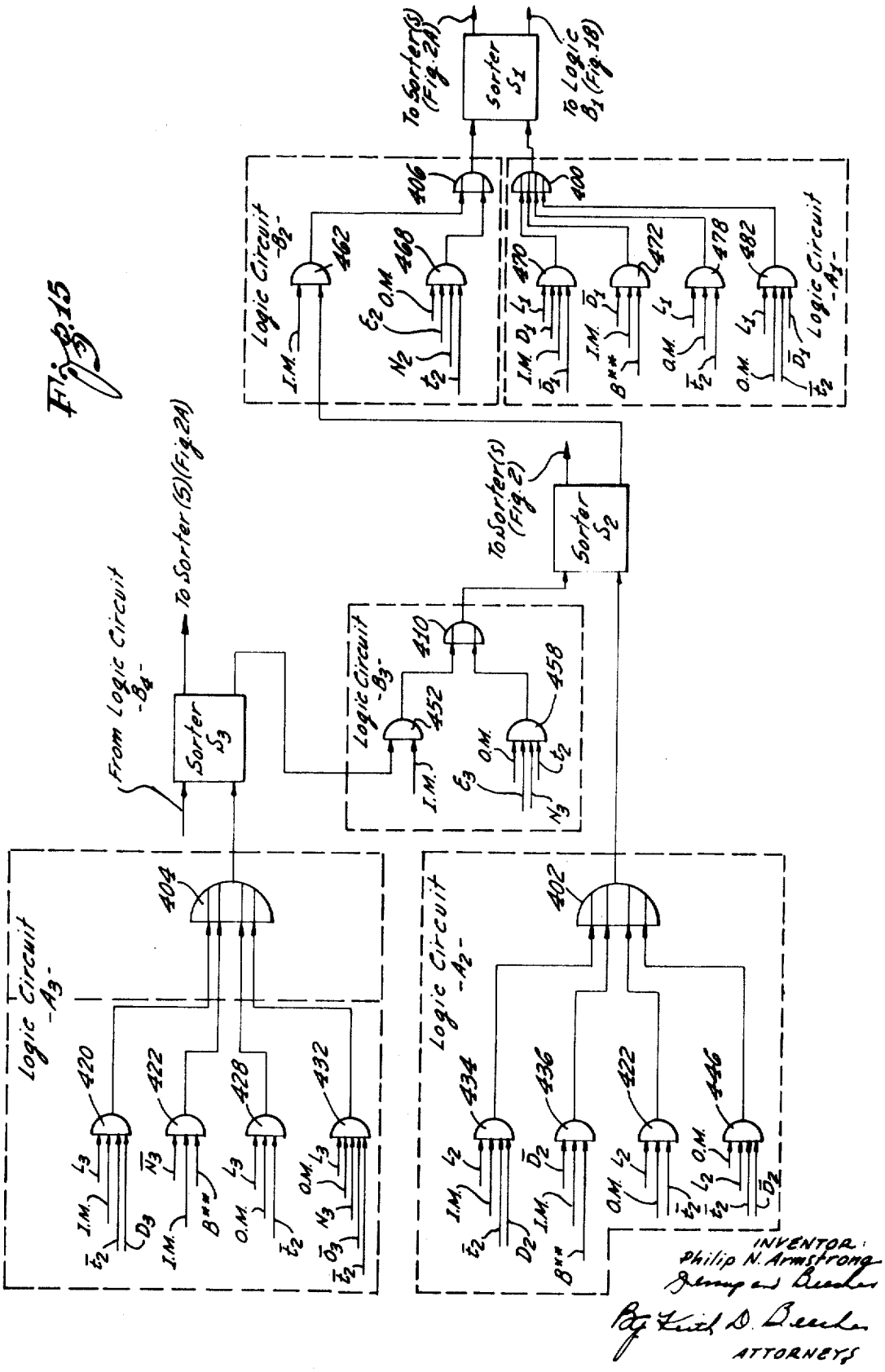


Fig. 15
To Sorter(s) (Fig. 2A)
Sorter S1
To Logic B1 (Fig. 1B)
406
400
Logic Circuit -B2-
462
468
470
472
478
482
Logic Circuit -A1-
I.M.
E2 O.M.
N2
I.M. D1
I.M. D1
O.M.
To Sorter(s) (Fig. 2)
Sorter S2
From Logic Circuit -B4-
To Sorter (S) (Fig. 2A)
Sorter S3
Logic Circuit -B3-
410
452
458
I.M.
E3 O.M.
N3
Logic Circuit -A3-
404
420
422
428
432
Logic Circuit -A2-
402
434
436
422
446
INVENTOR:
Philip N. Armstrong
By Kirk D. Beecher
ATTORNEYS

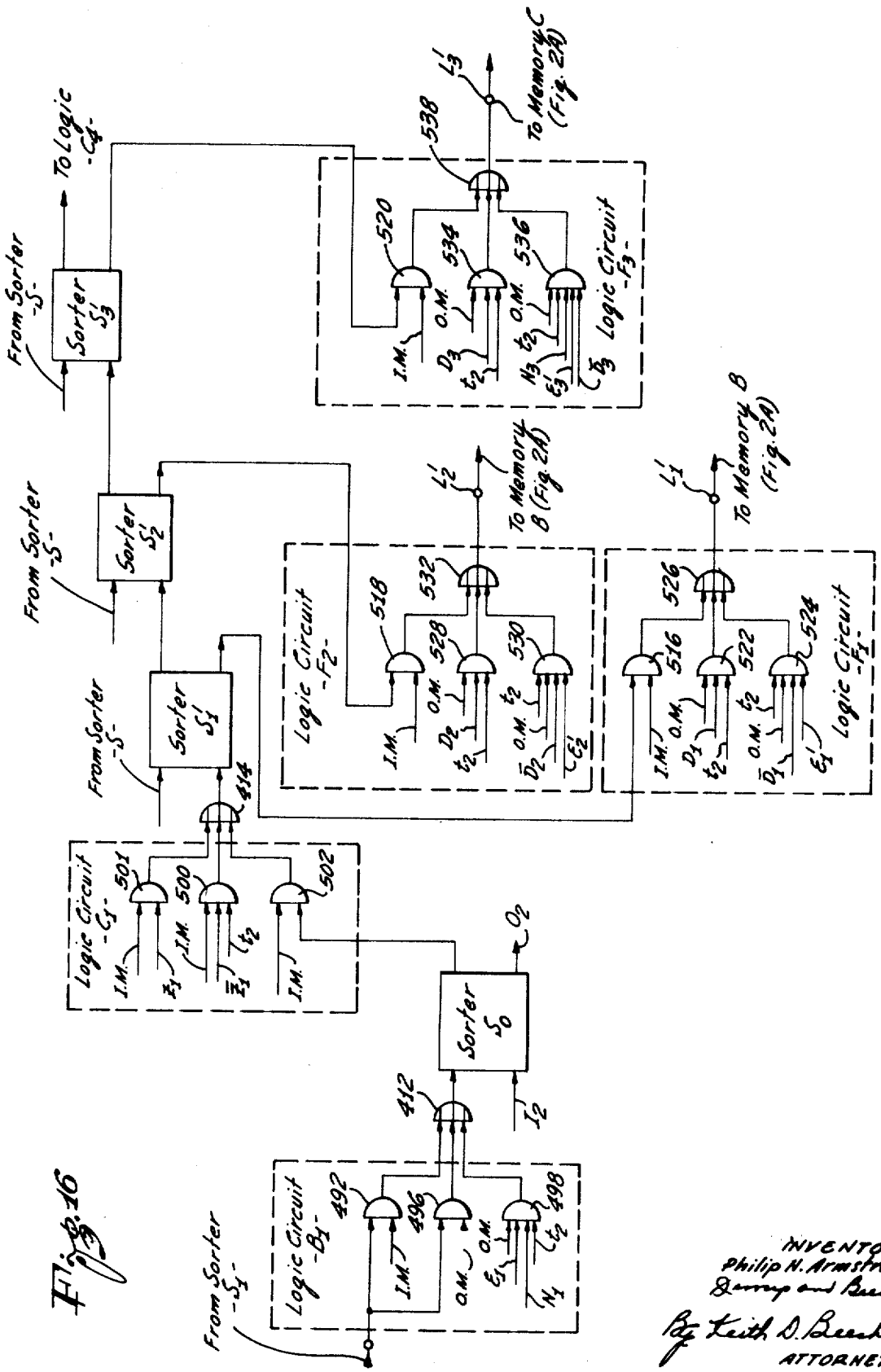


Fig. 1C

INVENTOR:
Philip N. Armstrong
Dennis and Brooks
By Keith D. Brooks
ATTORNEYS

Fig. 17

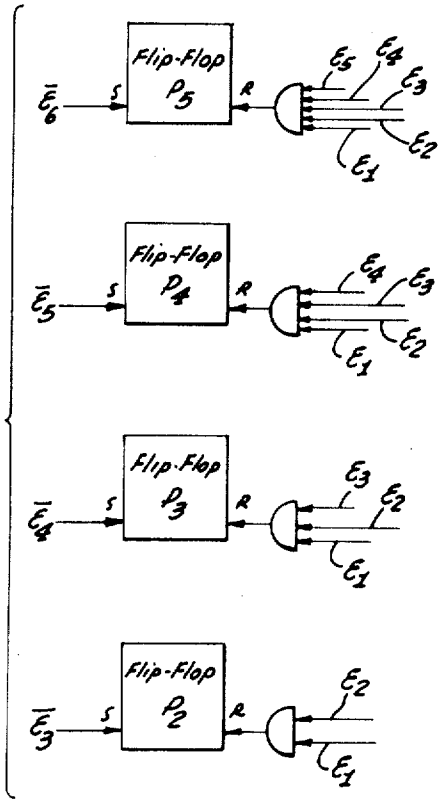


Fig. 18

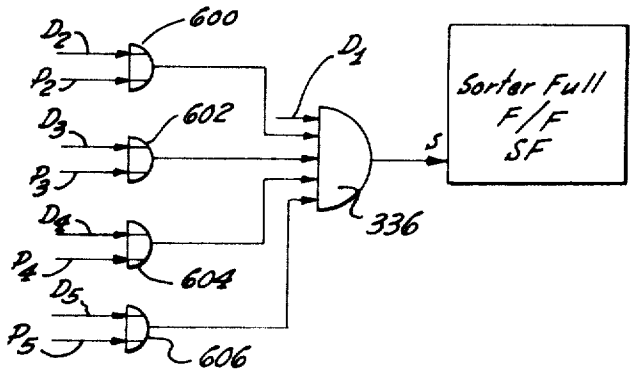
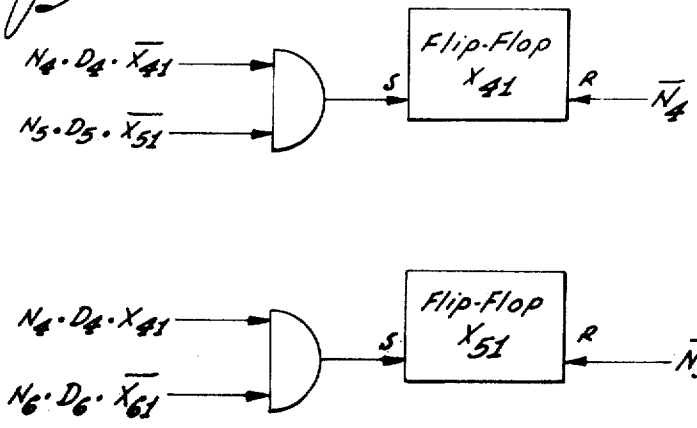


Fig. 22 (First Pass) Memory D



INVENTOR:
Philip N. Armstrong
Dennys and Beecher
By Keith D. Beecher
ATTORNEYS

Fig. 19

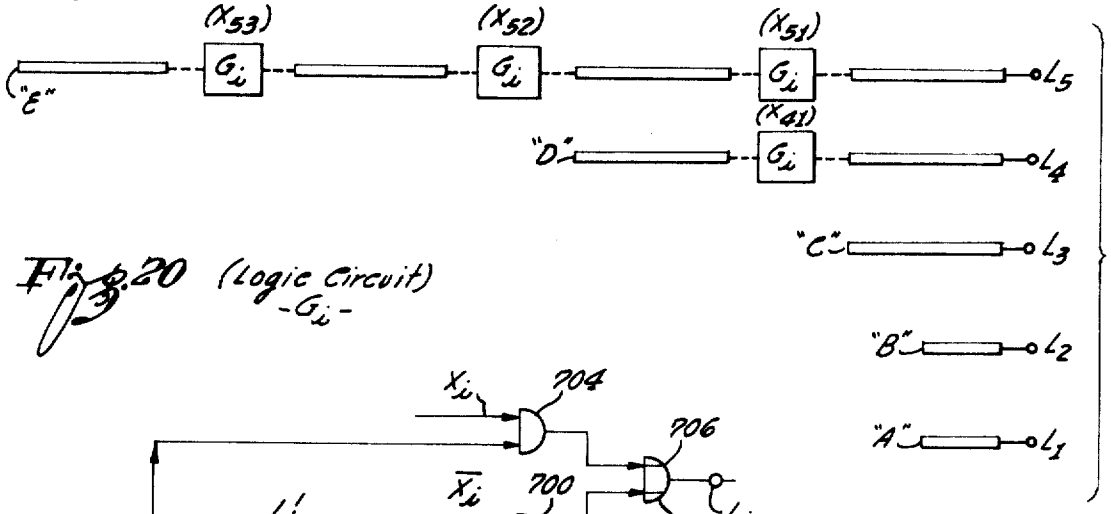


Fig. 20 (Logic Circuit) -Gi-

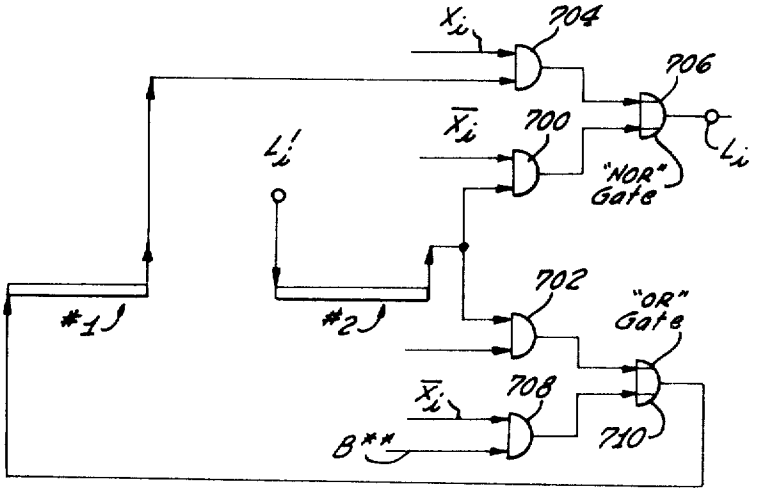
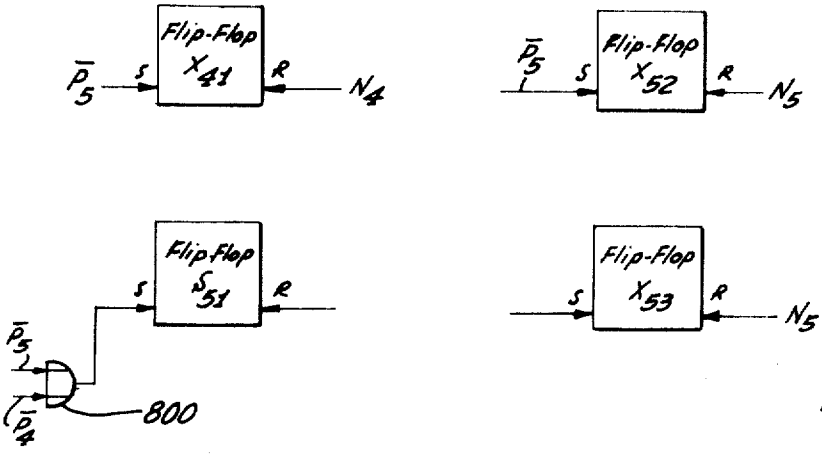


Fig. 23

Control of Flip-Flops Xab for Succeeding Passes



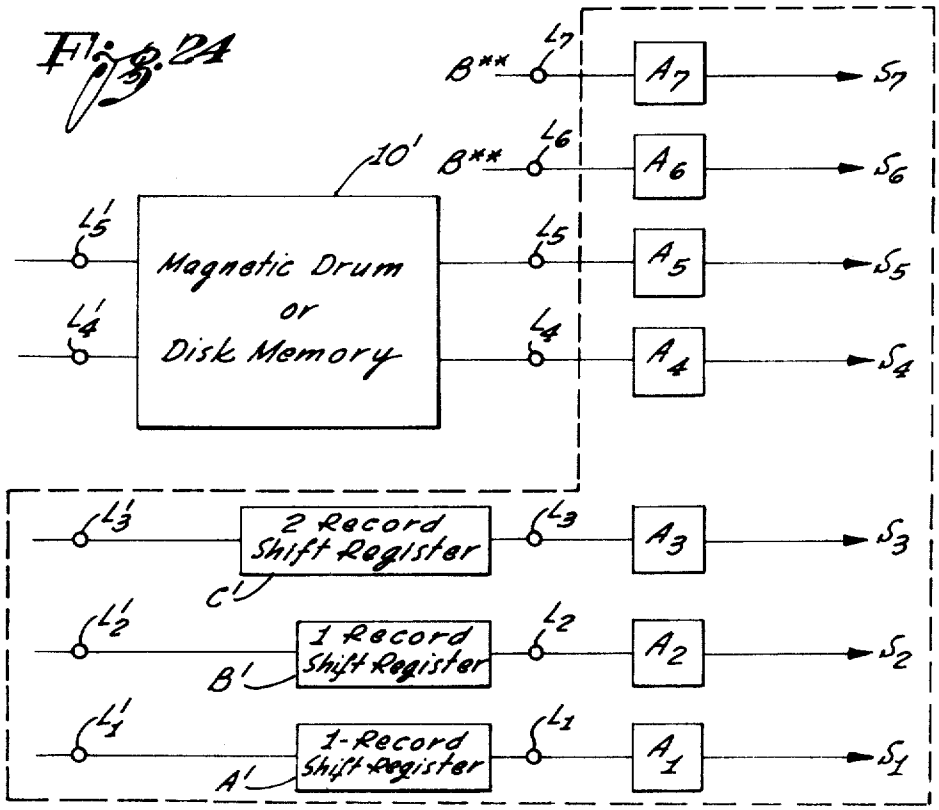
INVENTOR:
Philip N. Armstrong
Jump and Bucher
By Keith D. Bucher
ATTORNEYS

Fig. 21

Step	Memory A	B	C	D	E	Input
1	13*	X	X X	X X X X	X X X X X X X X	(13)
2	13*	X	X X	X X X X	X X X X X X X X	(8**)
3	16	13*	X X	X X X X	X X X X X X X X	(16)
4	13	16	10* /	X X X X	X X X X X X X X	(10)
5	7	13	16 10*	X X X X	X X X X X X X X	(7)
6	7	10	13 16	X X X X	X X X X X X X X	(8**)
7	4	10	16 13	X X X X	X X X X X X X X	(4)
8	4	10	13 16	9* X X X	X X X X X X X X	(9)
9	7	10	16 13	6* 9* X X	X X X X X X X X	(7)
10	3	7	10 16	13 6* X X	X X X X X X X X	(8**)
11	3	4	16 10	7 13 X X	X X X X X X X X	(8**)
12	3	4	10 16	13 7 X X	X X X X X X X X	(8**)
13	3	4	7 10	16 13 X X	1* X X X X X X X	(1)
14	3	4	10 7	13 16 X X	2* 1* X X X X X X	(2)
15	3	4	5 10	16 X 7 13	1* 2* X X X X X X	(5)
16	3	4	6 5	13 16 10 7	2* 1* X X X X X X	(6)
17	4	5	12 6	16 10 7 13	3* 2* 1* X X X X X	(12)
18	5	6	11 12	13 16 10 7	4* 3* 2* 1* X X X X	(11)
19	5	6	12 11	7 13 16 10	1* 4* 3* 2* X X X X	(7)
20	5	6	11 12	10 7 13 16	2* 1* 4* 3* X X X X	(7)
21	5	6	12 11	7 10 16 13	1* 2* 3* 4* X X X X	(7)
22	6	8	11 12	10 7 13 16	5* 1* 4* 3* 2* X X X	(8)

INVENTOR:
Philip N. Armstrong
Jerry and Beech
By Keith D. Beecher
ATTORNEYS

Fig. 24



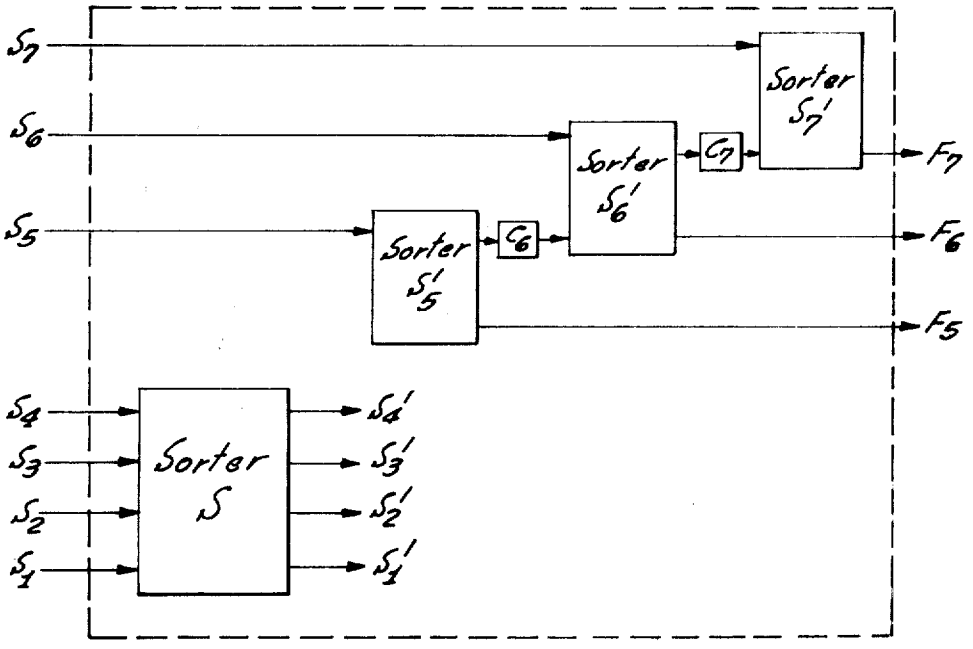


Fig. 25

INVENTOR:
Philip N. Armstrong
Jennings and Beecher
By Keith D. Beecher
ATTORNEYS

Fig. 26

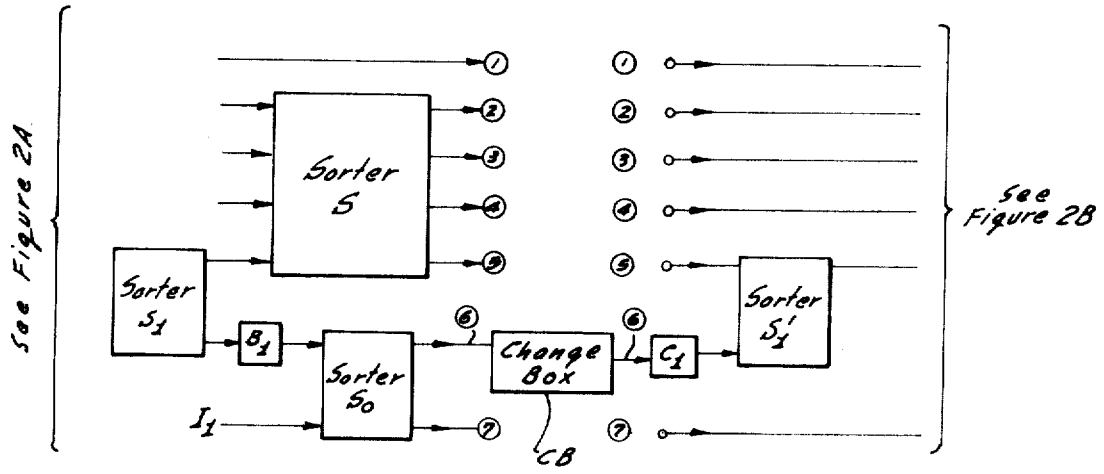
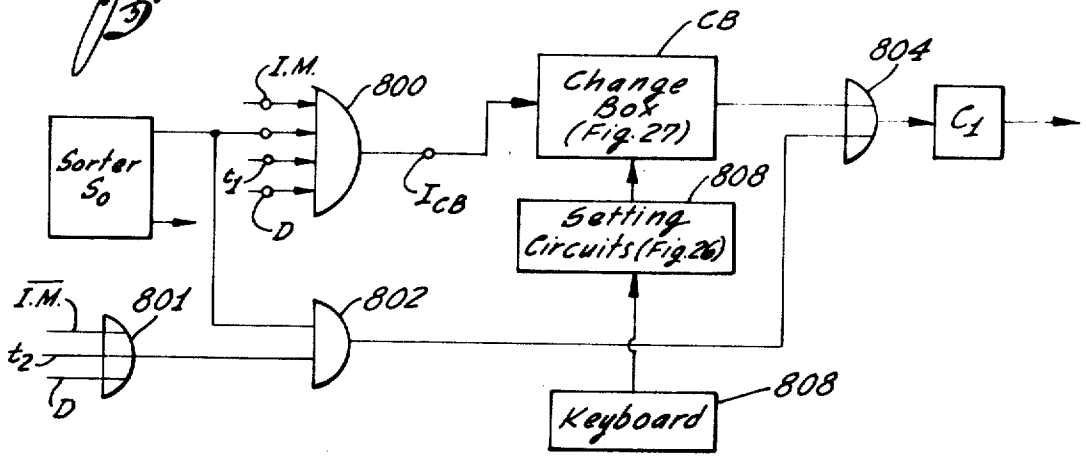


Fig. 27



INVENTOR:
Philip N. Armstrong
By Kemp and Beecher
By Keith D. Beecher
ATTORNEYS

Fig. 28

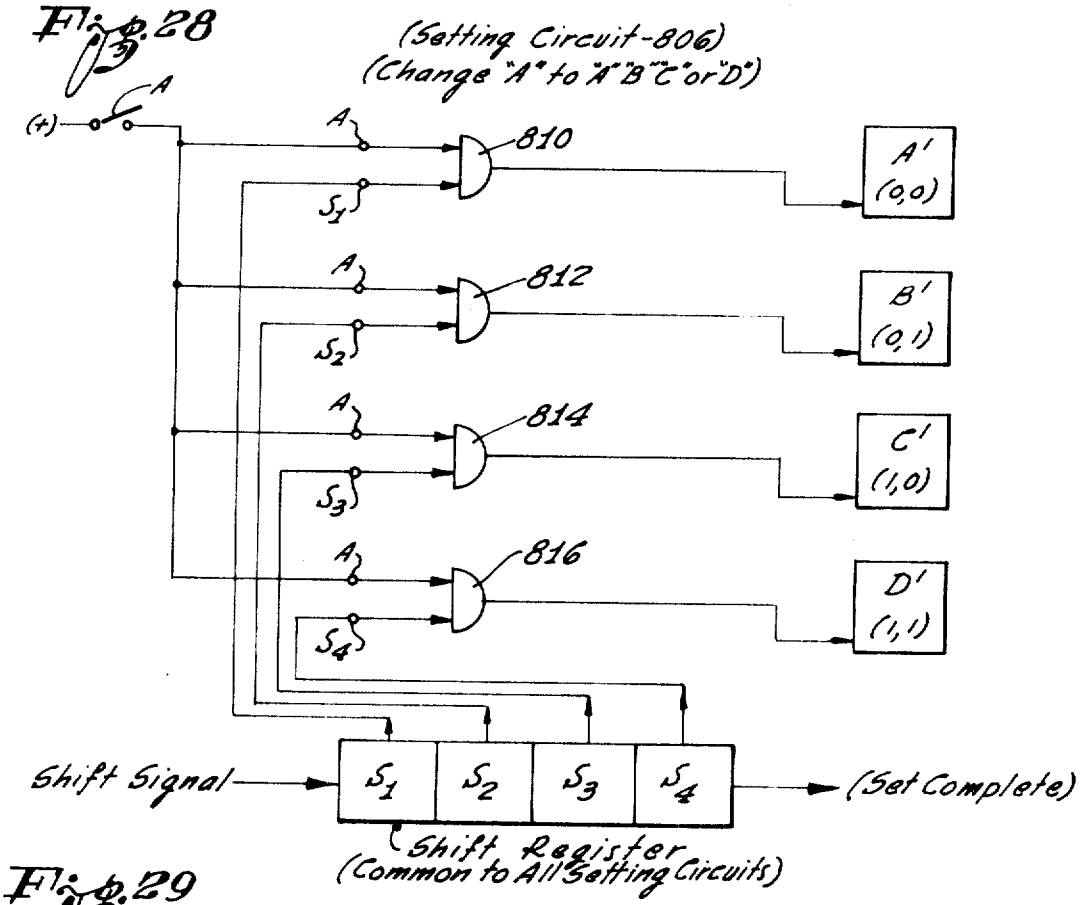
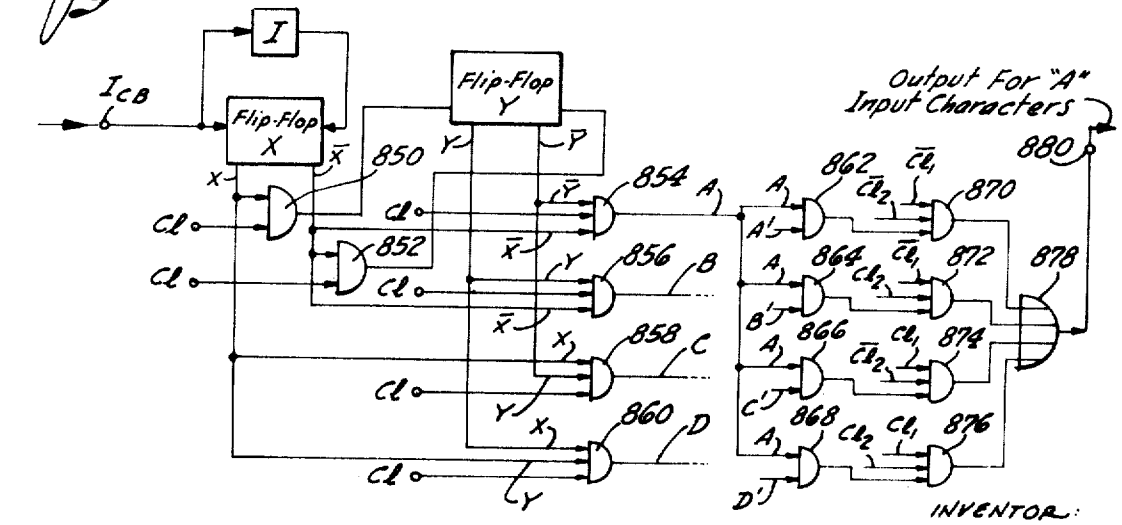


Fig. 29



INVENTOR:
Philip N. Armstrong
Jennings & Beecher

By Keith A. Beecher

ATTORNEYS

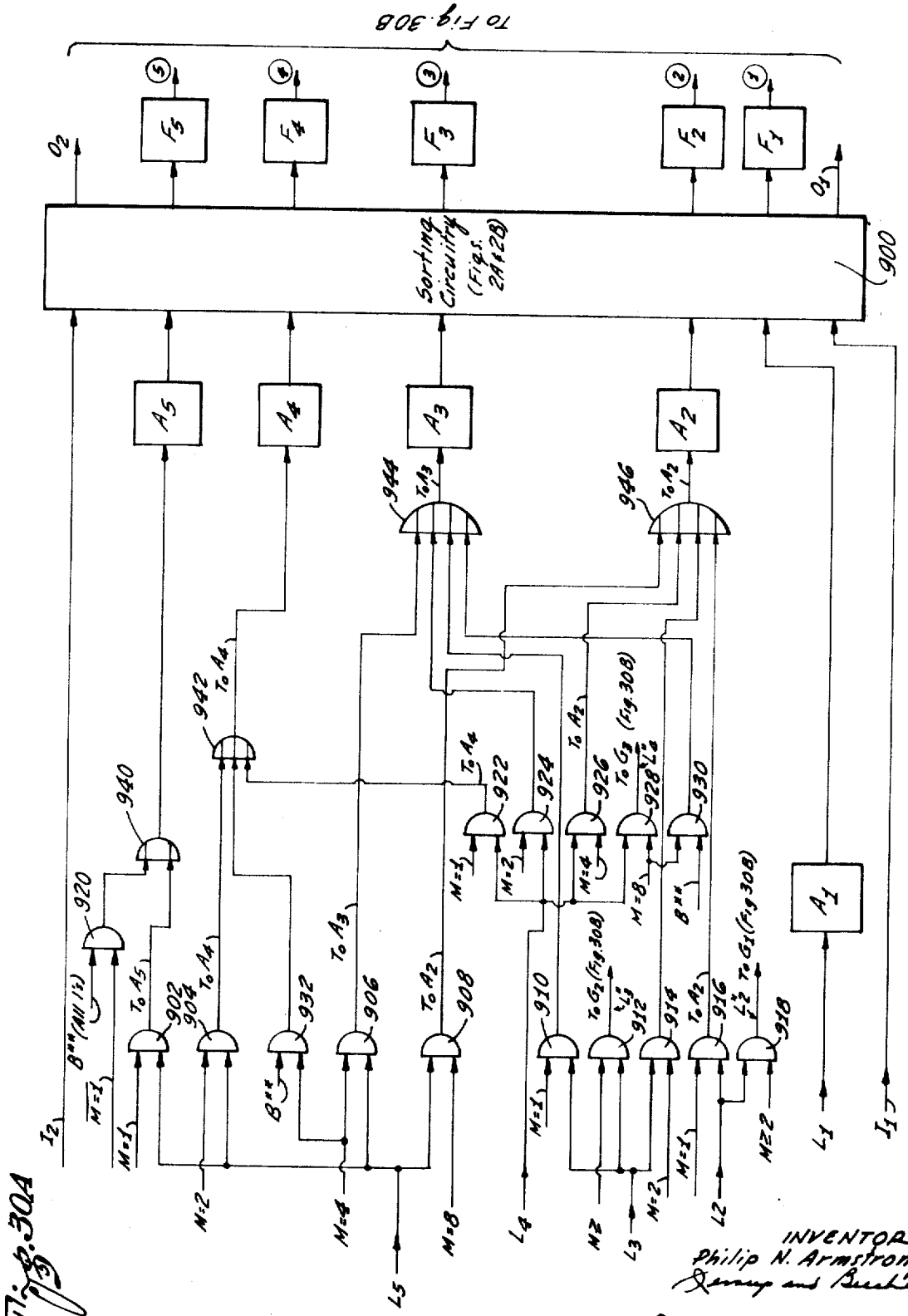


Fig. 30A

INVENTOR:
Philip N. Armstrong
By Kidd, Gamon and Berch
ATTORNEYS

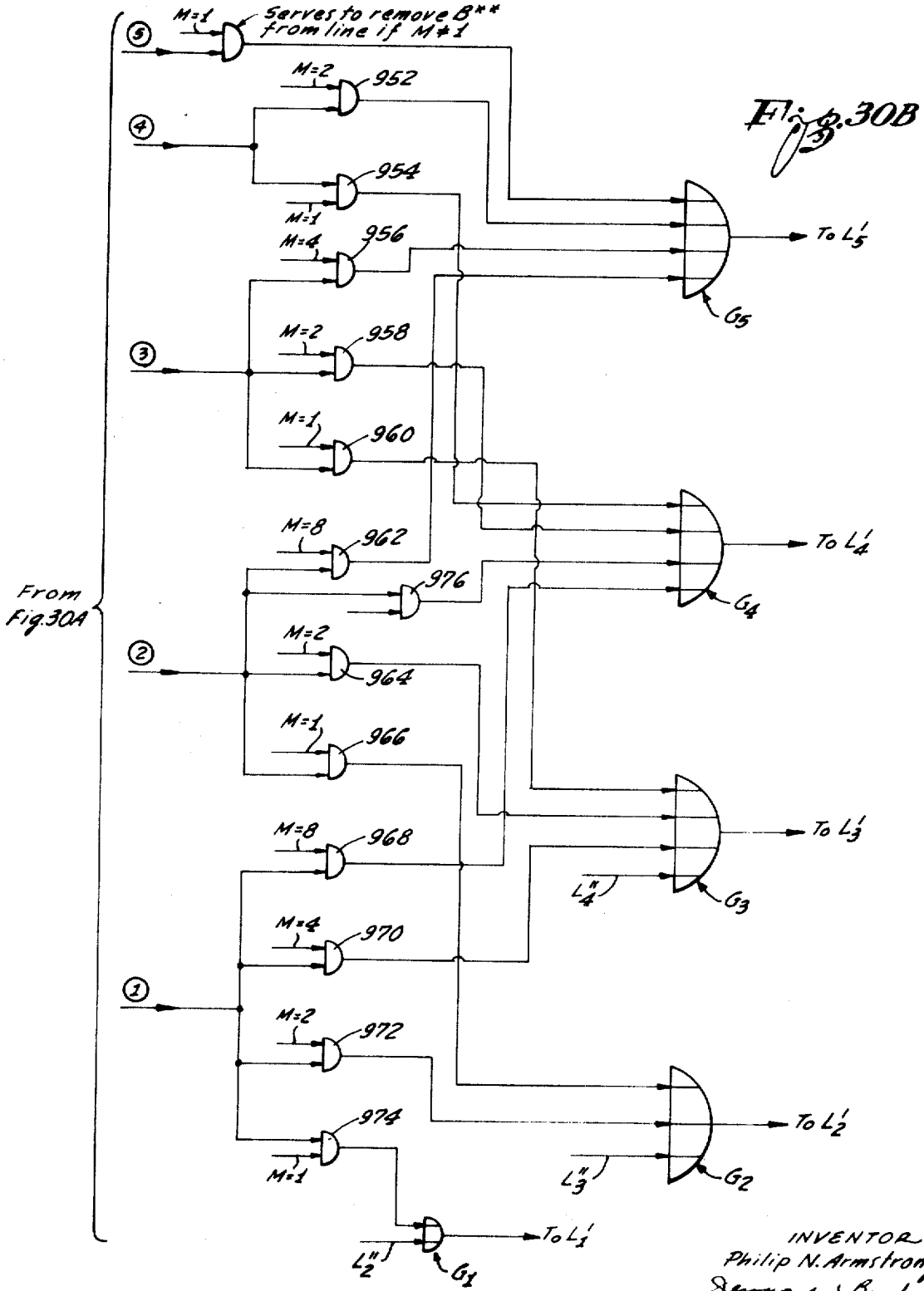
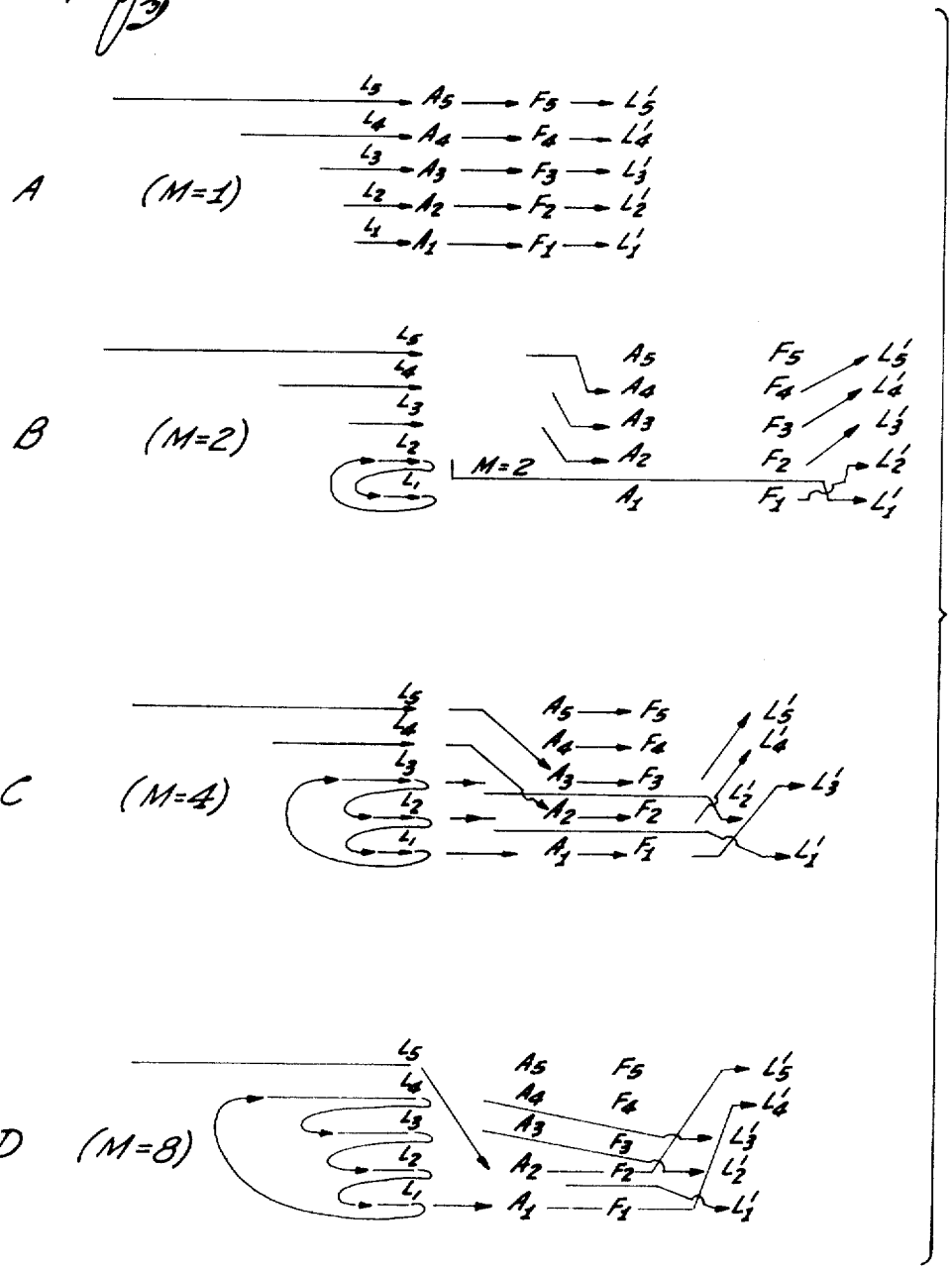


Fig. 30B

INVENTOR:
Philip N. Armstrong
Jennings and Breeden
By Keith S. Breeden
ATTORNEY

Fig. 31



INVENTOR
Philip N. Armstrong
Strong and Brooks
By Keith D. Brooks
ATTORNEYS

Fig. 32

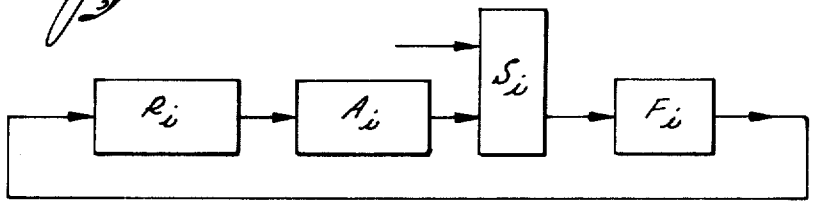


Fig. 33

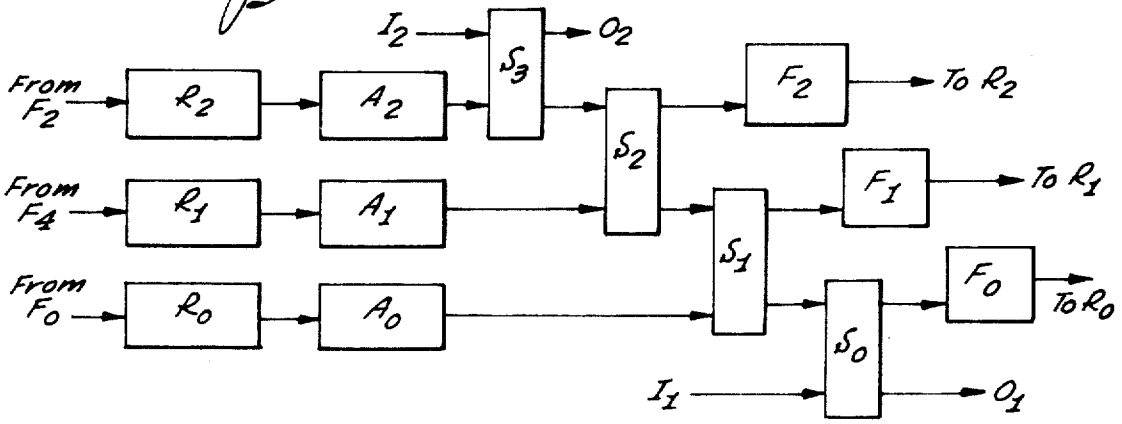
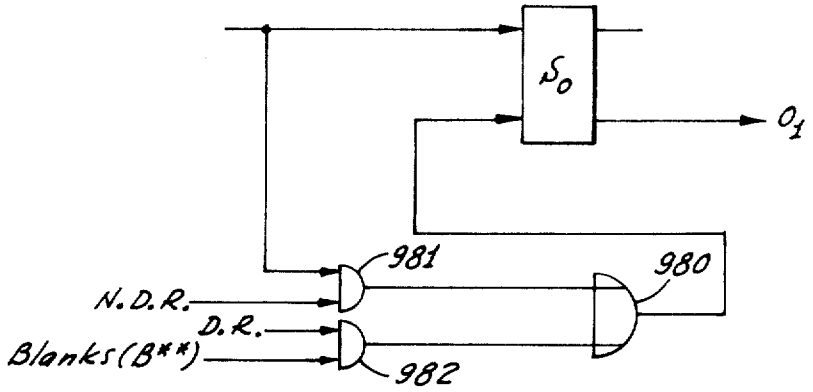


Fig. 35



INVENTOR:
Philip N. Armstrong
Dennys and Beecher

By Keith D. Beecher
ATTORNEYS

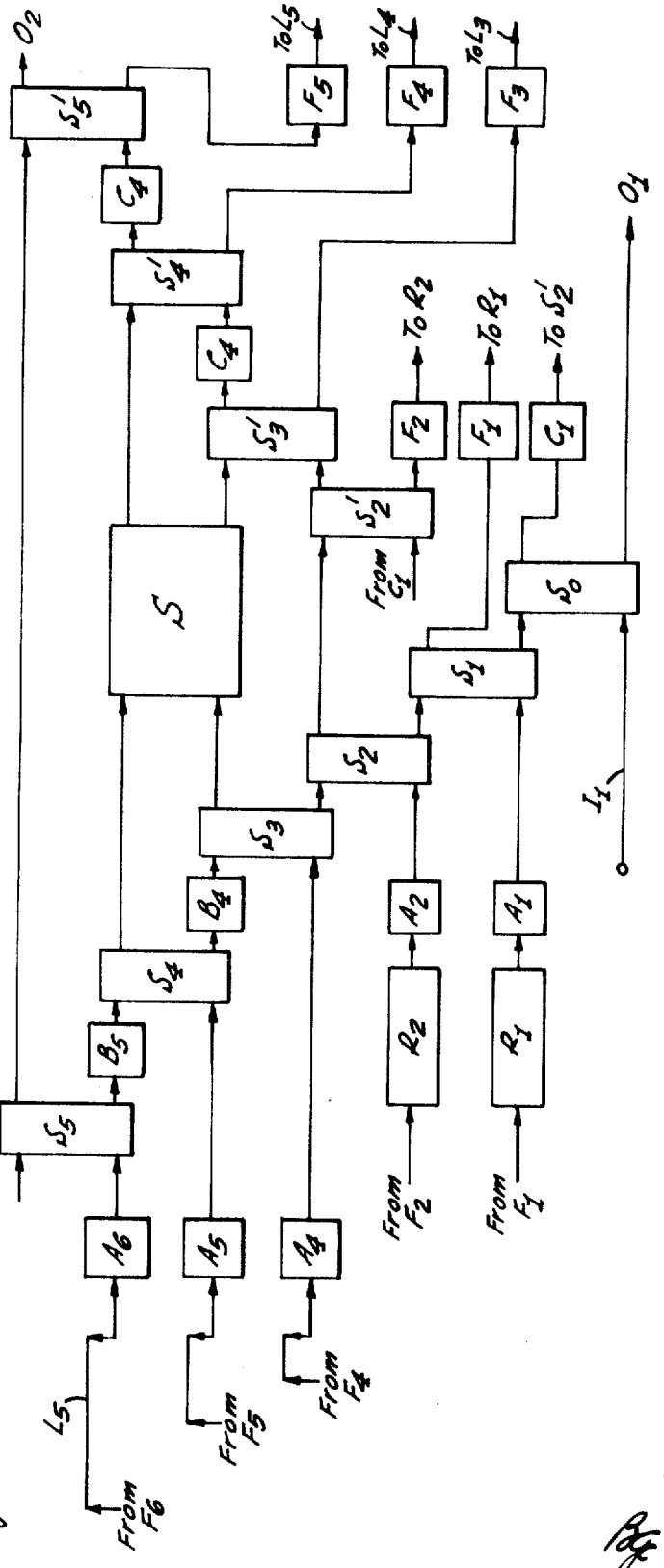


Fig. 34

INVENTOR:
Philip N. Armstrong
Kemp and Beecher
By Keith D. Beecher
ATTORNEYS

DATA SORTING SYSTEM

BACKGROUND OF THE INVENTION

The sorting system of the present invention is of the same general type as the sorting system described in U.S. Pat. No. 3,399,383, which issued Aug. 27, 1968, to Philip N. Armstrong. As is the case with the system described in the patent, the improved sorting system of the present invention is capable of rearranging binary records into a desired sequence, and it is intended to be used as an adjunct to a general purpose computer. The primary objective of the invention is to provide a system which is capable of rearranging records in a file, in which the number of records exceeds the normal capacity of the system, and of achieving this sort in a rapid manner and on an efficient basis.

A feature of the improved sorting system of the present invention, as was the case with the sorting system described in the aforesaid patent, is that it can be coupled to a wide variety of present day digital computers, including those using magnetic tape buffering equipment. For example, the uses to which the system of the invention may be applied include magnetic tape sorting; large volume inventory control; billing; mailing list preparation; library retrieval, and so on. In general, the sorting system of the invention is useful in all numerical computations in which placing data in a particular order is an essential part; and it also finds utility in the general storage of data.

The records which are to be sorted in the system of the invention are represented by equal length multibit binary numbers. Each record includes a control field bearing an identifying number which, in turn, is represented in usual manner in multiple bit binary digital form. In the embodiment to be described, it is assumed that the identifying number for each of the records is an integer. The sort is accomplished when the identifying numbers in the control fields of a series of records represent a monotonic sequence of numbers which, in the embodiment to be described, will be considered as increasing, although the system can just as easily be constructed to provide a decreasing series. For example, the sequence may be changed from increasing to decreasing, or vice versa, by the provision of complementing inverters at the input and output of the system. Such a complementing system will be shown and described subsequently herein.

The sorting system of the invention, as mentioned above, is intended to be used in conjunction with a file of records greater than the internal capacity of the system. The system is controlled to undergo an input operation, during which the records of the file are fed in an unsorted sequence into the sorting system until the system is full. The unsorted sequence of input records are then continued to be fed into the sorter, and together with the records already in the sorting system appear at its output in a generally sorted condition. The operation is continued until all the records of the file fed into the sorter have been produced at its output. If the indication is that the file of records is not completely sorted at the end of the first pass, the records are again passed through the sorting system. This is continued until a complete sort has been achieved. However, during each successive pass through the sorting system, its capacity is either unchanged or reduced to that required to achieve the limited sorting function needed for each successive pass, so that the time of each successive pass may be reduced, and so that the sorting system may operate at optimum efficiency at all times.

The aforesaid decrease of the capacity of the sorting system for each successive pass may continue, for example, until only two records are contained in the sorter at any particular time. The sort will be completed at the end of the pass in which no record exchanges have been made by the sorting system.

It will become evident as the description proceeds that the particular sorting system of the invention may incorporate the logic circuitry of the system described in the aforesaid patent, so that it may additionally operate in the same manner as the previous system to sort files of records which are fewer in number than the capacity of the sorter.

The operation of the sorting system of the invention, as will be described, and as described in the aforesaid patent, depends in part on the facility of the system for separating the records to be sorted into delay line, or other type memories of lengths corresponding to 1, 1, 2, 4, 8...2^n records respectively. The number of delay line memories chosen in any particular system depends upon the desired capacity of the system. As pointed out in the patent, extreme flexibility can be achieved in the type of sorting system under consideration, since any commercial embodiment of the sorting system can in general be tailored to fit the number of records to be processed.

For purposes of simplifying the description of the invention as much as possible, the embodiment to be described is assumed to include five of the aforesaid delay line type memories, and these memories have respective lengths corresponding to 1, 1, 2, 4 and 8 records. The selection is such that two of the memories each holds one record, one of the memories holds two records, one of the memories holds four records, and one of the memories holds eight records. Whenever a record is placed in any one of the memories, it may be removed from that memory a number of record times later, as determined by the capacity of the particular memory.

In the embodiment to be described, for example, any record placed in either of the two least memories can be recovered during the next record time, any record placed in the third memory can be recovered after a delay of one record time, any record placed in the fourth memory can be recovered after a delay of three record times, and any record placed in the fifth memory can be recovered after a delay of seven record times.

The aforesaid delay line type memories are synchronous, and in the embodiment to be described, the memories are provided by writing the records on different tracks of a magnetic tape, and by subsequently reading the information from each track on the tape at the required different record time delays. This permits the various records to be placed simultaneously in all the memories, and it also allows the records to be recovered simultaneously from the memories. It should be evident, however, that many types of delay line and other memories are suitable for the purpose of the system of the invention.

During the operation of the system, the received records are placed in the different memories during successive record times in a serial manner, so that the first record is compared with the contents of the empty system, and each successive record is compared with the previously inputted records. The contents of the empty system are represented by "blanks" which bear control numbers larger than the identifying number of any record to be sorted. The first record received is placed in the first least memory. Then, when the second and successive records are serially received, they are compared with the preceding records, and the records are placed in a sorted condition in the different memories.

As mentioned above, the sorting system of the present invention is one capable of handling a file of records in which the number of records is greater than the internal capacity of the system, and it serves to recirculate the records through the system for a succession of passes, until a complete sort has been achieved. For each successive pass, the number of delay line memories either remains the same or is reduced accordingly, and the length of the remaining delay line memories may also be reduced, so that the capacity of the system is never any larger than that required to perform the sort operation for the successive pass, and so that the time required for each successive pass is reduced accordingly and is no longer than absolutely necessary.

For practical purposes, and in order to adapt the system for use in conjunction with a variety of sizes and types of memories, the shorter delay line memories may be replaced by shift registers. These shift registers may be integrated circuit types, and they may be combined in a single unit with other logic cir-

cuits in the system. The result is a standard sorter network with a predetermined number of inputs. The sorter network may then be used in conjunction with standard magnetic drum or disc memories, with tracks thereon serving as the longer delay line memories. If the shorter memory line lengths, as represented by the shift registers, are so selected that the longer memory line lengths consist of integral numbers of disc drum tracks, standard magnetic memory drum or disc equipment may be used.

If the particular drum or disc magnetic memory which is used in conjunction with the standard sorter network described in the preceding paragraph has fewer memory lines than inputs to the network, then appropriate circuitry is used to assure that only a corresponding part of the memory network will be used.

The apparatus to be described also incorporates a logic control unit, designated a "change box," through which the records to be sorted are passed, and which is capable of changing the identification field of selected records in a predetermined way. This permits any desired group of records in the file to be shifted to the front of the file, for example, for the retrieval of data in that group.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic representation of the format of a typical record as processed by the sorting system of the invention;

FIGS. 2A and 2B are schematic block representations of the sorter system described in the aforesaid patent, and which may be modified to perform the sort function of the present invention;

FIG. 3 is a representation of the interconnections of a plurality of sorting blocks, which are included in one of the components of the overall system of FIGS. 2A and 2B;

FIGS. 4, 6 and 8 are representations of a file of records which are successively fed into the system on an interrupted basis;

FIGS. 5, 7 and 9 are tables showing how the aforesaid file is processed in the sorting system;

FIG. 10 is a logic diagram of a sorting network which may be used in the system;

FIG. 11 is a table showing the control bits for different records and blanks used in the system;

FIGS. 12, 13 and 14, 17, 18, 22 and 23 show the logic associated with flip-flops used in the various embodiments of the system;

FIGS. 15 and 16 are logic diagrams of sorter networks included in the system;

FIG. 19 is a second embodiment in which the individual memory lines have controllable lengths;

FIG. 20 is a logic diagram of circuitry associated with the embodiment of FIG. 19;

FIG. 21 is a table showing how the aforesaid file of records is handled by the embodiment of FIG. 19;

FIGS. 24 and 25 are block diagrams showing how the sorter may be adapted for use with standard memory equipment;

FIGS. 26, 27, 28 and 29 are block diagrams showing the inclusion of the aforesaid "change box" in the sorter system;

FIGS. 30A, 30B and 31*a*—31*d* are diagrams showing how the system may be modified to handle records of multiple lengths; and

FIGS. 32—35 are block diagrams of a portion of the system further modified to incorporate one-record register and sorter modules.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENT

The format of the records processed by the sorting system of the invention is shown as FIG. 1, and as shown, each of the records includes a data field in which the data pertaining to the particular record is contained in binary coded form. Adjacent the data field is an identifying number field which contains the identifying number for the record, this being in the form of a multibit binary number representing a particular integer. A control field is placed adjacent the identifying number field, and this control field may be treated during the processing by the system so as to alter the identification of the record. An empty field is provided at the end of the record, and which is utilized to permit time for switching the control operations in the system.

The most significant bit of the record is to the right in FIG. 1, and the record is sensed from the right to the left, that is from the most significant bit to the least significant bit. Therefore, each record is sensed by the system first through the interval of the empty field, then the control field is sensed from the most significant to the least significant bit, and the identifying field is next sensed from the most significant to the least significant bit. This permits the identifying number of the identifying number field to be increased to any desired extent depending upon the more significant bits which may be inserted into the control field.

As will be described, multibit binary blanks are also passed through the system from time to time, these being designated as "blanks" (B) in which the data field and identifying field are all 1's; or "blanks" (Z) in which the data field and identifying number field are all 0's. In addition, further "blanks" (B* and B**) are used in the system, and which are distinguished by the binary coded numbers in their control fields. Also, there are several types of records, which likewise are distinguished from one another by the binary coded numbers in their respective control fields.

The different records and blanks handled by the sorting system to be described may be represented by the following table:

Label	Data and identifying fields	Control field
B** (blank)	All 1's	111
B* (blank)	do	110
D* (altered record)	Record data and identifying number	110
B (blank)	All 1's	010
D+ (altered record)	Record data and identifying number	010
D (record)	do	100
Z (blank)	All 0's	000

The identifying field of each of the record words labeled D, D+ or D* includes a binary representation of an integer identifying the particular record. The most significant bit of the number in the identifying field is adjacent to the control field of the particular record. Therefore, as mentioned above, the identifying number which identifies and controls the sort of the particular record appears adjacent the control field, as shown in FIG. 1. Then, the bits in the control field serve to increase the significance of the identifying number in the identifying field.

At least one bit position in the data field of the D, D+ or D* records must be a 0. It follows that if the contents of the combined data identifying and control fields are considered as binary integers, they are naturally ordered if a count is taken of the restriction of the data field to control at least one zero. Then, at all times:

$$Z<D<D+<B<D^*<B^*<B^{**}$$

In the operation of the system, and as mentioned above, a succession of records are caused to flow into the system until it is filled, at which time the records appear at the output of the system while the input operation is continued.

The system may have the general form shown in FIGS. 2A and 2B. The system shown in FIG. 2A, for example, includes a storage medium 10 which may be in the form of a magnetic tape, on which different tracks are provided to form different delay-line type memories for the records of FIG. 1 which are to be processed by the system. As mentioned above, these memories have different lengths in a predetermined progression corresponding to the different numbers of records to be respectively stored therein. In the illustrated embodiment, and as also noted above, the memories represent a progression 1, 1, 2, 4, 8 insofar as the individual record storage capabilities are concerned. However, the system is not limited to powers of two progressions, and other progressions may be used.

The magnetic tape, which constitutes the storage medium **10** in the system of FIG. **2A**, is shown in fragmentary form and is assumed to be moving from the left to the right in FIG. **2A**. The tape includes a series of different tracks which extend along its length. Corresponding read and write electromagnetic transducer heads are provided for writing the records and blanks in the different tracks, and for subsequently reading the records and blanks from the tracks. The read and write heads are designated by the arrows in FIG. **1**.

The magnetic tape of the storage medium **10** also includes a timing track which has regularly spaced magnetic recordings, these being used for bit timing clocking purposes. A transducer read head **16** is magnetically coupled to the timing track, and it responds to the recording on the track to provide clock pulses CL at its output. These clock pulses represent the bit timing in each of the records processed in the system.

The clock pulses are also used to synchronize the operation of a record bit counter **33** and of a record word counter **35**. The bit counter **33** produces timing pulses t_0, t_1, t_2...representing the different bit times in each word time in the system; and the word counter produces timing pulses w_0, w_1, w_2...designating successive word or record times in the system.

The magnetic tape of FIG. **2A** includes a track designated A, and this track serves as one of the least memories for the system. A write head w_0 is magnetically coupled to the track A, and this head responds to binary signals applied to the input terminal L'$_1$ to write the signals in that track. A read head R_0 is also magnetically coupled to the track A, and the read head is displaced from the write head along the track a distance corresponding to one record time. Therefor, any record or blank stored in the least memory of the track A is read from the memory during the following record time.

A second least memory is provided by a track B on the magnetic tape, and an appropriate write head w_1 and read head R_1 are magnetically coupled to the track B. The write head w_1 responds to binary signals applied to the input terminal L'$_2$ to write the corresponding binary signals on the track B. The read head R_1 is displaced along the track B a distance corresponding to one record from the write head w_1. Therefore, in the second least memory, any record entered therein during a particular recorded time is recovered during the following record time.

The magnetic tape also includes a track designated C. A write head w_2 and a read head R_2 are magnetically coupled to the track C. The read head R_2 is displaced along the track C from the write head w_2 a distance corresponding to two records. This latter track forms the third memory, and the write head w_2 responds to input terminals introduced to the input terminal L'$_3$ to produce corresponding recordings in the track C. In this third memory, a record introduced during any record time may be recovered two record times later. In addition, the third memory is capable of storing two records, rather than one, as was the case with the two least memories.

A fourth memory is also provided on the tape by a track D, and a write head w_3 and a read head R_3 are magnetically coupled to that track. The write head w_3 responds to binary signals applied to the input terminal L'$_4$ to produce corresponding recordings in the track D. The read head R_3 is displaced from the write head w_3 along the track D a distance corresponding, for example, to four records. Therefore, the fourth memory is capable of holding four separate records, and any record introduced to that memory during any particular record time can be recovered after an elapse of three record times.

A fifth memory, capable of storing eight records, is also provided on the tape in a track E. The write head w_4 and a read head R_4 are magnetically coupled to the track E. The write head responds to signals received from an input terminal L'$_5$ to record corresponding signals in the track E on the tape. The read head R_4 is displaced from the write head w_4 along the track E a distance corresponding to eight records. Any record introduced to the fifth memory may be recovered after an elapse of seven record times, and the fifth memory is capable of storing eight records at any time.

As mentioned above, the number of individual memories illustrated in FIG. **2A** is for purposes of illustration only. More or less tracks on the magnetic track may be used, depending upon the required capabilities of the system. As mentioned, the number of memories follows the progression 1, 1, 2, 4, 8,...2^n insofar as records stored and record times required to pass therethrough are concerned.

The read heads R_0—R_4 are connected to corresponding output terminals L_1—L_5 which, in turn, are respectively connected to logic circuits represented in FIG. **2A** by the blocks A_1, A_2, A_3, A_4 and A_5. These logic circuits will be described in detail subsequently herein. The blocks A_1—A_5 are respectively connected to the lower input terminals of a series of sorters designated by the blocks S_1, S_2, S_3, S_4, S_5, which also will be described.

As shown, each of the sorters S_1—S_5 includes two input terminals and two output terminals. The individual sorters respond to records serially applied to their two input terminals to process such records in a manner such that the record with the higher identifying number appears at one of the output terminals and the record with the lower identifying numbers appears at the other output terminal.

An input terminal I_1 for the system is connected through the illustrated logic circuitry to the lower input terminal of a sorter S_0, which is similar to the sorters S_1—S_5. A further input terminal I_2 of the system is connected to the upper input terminal of the sorter S_5. The logic circuitry includes a pair of "AND" gates **12** and **14**, an inverter **16**, and an "OR" gate **18**, connected as shown.

The lower output terminals of the sorters S_1—S_5 are respectively connected to logic circuits represented in FIG. **2A** by the blocks B_1—B_5 which also will be described in detail subsequently. These logic circuits are respectively connected to the upper input terminals of the sorters S_0—S_5, and the upper output terminals of the sorters S_1—S_4 are connected to the input terminals of a sorter represented by the block S in FIG. **2A**. The sorter S is made up of a plurality of individual blocks, such as shown in FIG. **3**. Each of the individual blocks making up the sorter S may be similar to the individual sorters S_0—S_5, and each functions in the same manner.

The blocks of the sorter S, as shown in FIG. **3**, are connected so that the records applied to the input terminals are rearranged by the blocks, and appear in ascending order, insofar as their identifying numbers are concerned, at the four output terminals designated **2**, **3**, **4** and **5** when the $\overline{\text{INVERT}}$ term enables the "AND" gate **12** in FIG. **2A**, and in descending order when the INVERT term enables the "AND" gate **14**. The blocks in the sorter S may be inhibited, so that the records applied to the input terminals are passed through the sorter S in their original order.

The upper output terminal of the sorter S_5 is connected to the output terminal **1** of FIG. **2A**, and the output terminals of the sorter S_0 are connected to the output terminals **6** and **7**. The output terminals of the sorter S are connected to the remaining output terminals **2**, **3**, **4** and **5**, of FIG. **2A**, as indicated above. The output terminals **1**—**7** of FIG. **2A** are connected to correspondingly numbered input numbered **1**—**7** of FIG. **2B**.

The input terminals **1**—**5** of FIG. **2B** are connected to the upper input terminals of respective sorters represented by the block S'$_5$, S'$_4$, S'$_3$, S'$_2$ and S'$_1$. The input terminal **6** of FIG. **2B** is connected to a logic circuit represented by the block D_1 which, in turn, is connected to the lower input terminal of the sorter S'$_1$. The upper output terminals of the sorters S'$_1$, S'$_2$, S'$_3$ and S'$_4$ are connected to logic circuits represented by the blocks C_2, C_3, C_4 and C_5 respectively, which in turn are connected to the lower input terminals of the sorters S'$_2$, S'$_3$, S'$_4$ and S'$_5$. The lower output terminals of the sorters S'$_5$, S'$_4$, S'$_3$, S'$_2$ and S'$_1$ are connected to respective logic circuits represented by the blocks F_5, F_4, F_3, F_2 and F_1. These latter logic circuits are connected to respective output terminals L'$_5$, L'$_4$, L'$_3$, L'$_2$ and L'$_1$, which are connected back to the correspondingly numbered input terminals of FIG. **2A**. The upper output terminal of the sorter S'$_5$ is connected to an output ter-

minal of the system designated **0** , and the input terminal 7 of FIG. 2B is connected to a second output terminal of the system designated 0_2.

Before any records are fed into the system, the memories A—E of FIG. 2A are filled with blanks of any type. The logic circuits A_1—A_5 respond to these blanks, representing the absence of records, to apply the maximum blanks B** to the sorters S_1—S_5. This expedient, as described in the aforesaid patent, whereby the logic circuits A_1—A_5 generate the maximum blanks B** in the absence of records, obviates the necessity of filling the system with the maximum blanks B** before each input operation.

To carry out the sorting operation, and as shown schematically in FIG. 4, the records having the format shown in FIG. 1 are introduced serially to the input terminal I_1, and the minimum blanks Z (all 0's) are introduced to the input terminal I_2. In the tables of FIGS. 5 and 9, the B** blanks have been designated by X's to simplify the drawing.

The action of the sorting system during the operation is such that the maximum blanks B** (all 1's) appear at the output terminal 0_1 of FIG. 2B, and the minimum blanks Z (all 0's) appear at the output terminal 0_2. During this operation, as each record is fed into the sorting system, it is compared with the outputs of the different memories A—E. As explained, until any one of the individual memories is filled with records, its outputs are automatically transformed to maximum blanks B** by the respective logic circuits A_1—A_5.

When records appear at the outputs of the individual memories, they are compared with the successive input records during the successive word times, and the least of these records at each word time is altered by the logic C_1, so that it becomes a maximum record D*. This causes the altered lowest record to be placed in the longest memory exhibiting a record output. The A_1—A_5 logic circuits remove the asterisk (*) from each record as it leaves the corresponding individual memories A—E of FIG. 2B.

In the system of the present invention, and as will be described in detail herein, the records continue to be fed into the system after the system has reached its capacity, with the overflow records being fed out of the system in a generally sorted condition for subsequent recirculation, unless a complete sort is achieved by the first pass.

Should the introduction of the records into the sorting system be interrupted, as shown in the example of FIGS. 4 and 5, the maximum blanks B** normally applied to the input terminal I_1, and the minimum blanks Z are normally applied to the input terminal I_2. This causes the records previously fed into the system to be retained therein, and it prevents such records from being displaced out the output terminals 0_1 and 0_2 before the system has reached its maximum capacity.

During the interruption of the records being fed into the system during the input operation, it is important to assure that as the succession of records shifts into any of the individual memories, there are no interposed B blanks in the series. For that reason, during the interrupted phase of the input operation, the inputs and outputs of each of the individual memories are examined. Whenever a memory is found with a record input and a B* blank output, Z blanks are then introduced to the system by way of both the input terminals I_1 and I_2, and this results in the introduction of Z blanks into the different memories. The aforesaid introduction of Z blanks into the system at both the input terminals I_1 and I_2 is continued from one word time to the next during the interruption until none of the memories exhibits a record input and a B* blank output, which is indicative of a partially filled condition. Then, the system reverts to the introduction of B** blanks to the input terminal I_1, and of Z blanks to the input terminal I_2. During the interruption, the Z blanks from the least memory A are not altered by the C_1 logic circuit, as is normally the case with the least record in the system, as described above.

After an interruption in the input operation, the inputting of the records can be resumed at any word time. If the Z blanks were placed in the system during the interruption of the input mode, they will be displaced out of the system by the records subsequently fed into the system, after the longest record-bearing memory has been filled with records. Therefore, when the input mode is resumed, records are again fed to the sorting system by way of the input terminal I_1, and Z blanks are again introduced into the system by way of the input terminal I_2, this being continued until the longest data-bearing memory is filled with records, at which time records D or D* or Z blanks, will appear at the output of each memory containing records. When this condition is reached, B** BLANKS ARE introduced to the system by way of the input terminal I_2 as long as Z blanks are detected in the memory L_1 or until the inputting of records is again interrupted. Should a subsequent interruption occur, then B** blanks are applied to the input terminal I_1 and Z blanks are applied to the input terminal I_2.

When the absence of Z blanks in the memory A is detected, the system resumes its normal input operation. That is, records continue to be fed into the sorting system by way of the input terminal I_1 and Z blanks continue to be fed into the sorting system by way of the input terminal I_2. As shown in the example of FIGS. 4 and 5, records are fed into the system at successive word times during the input operation, with interruptions occurring during the operation, and with the operation again being resumed at subsequent selected record times. It will be remembered that during each word time during the operation, a B** blank is displaced out from the output terminal 0_1 and a Z blank is displaced out from the output terminal 0_2.

During the first record time of the input operation in the example of FIG. 4, a record identified by the integer "13" is fed into the sorting system by way of the input terminal I_1, and a Z blank is simultaneously applied to the system by way of the input terminal I_2. The record "13" is distinguished to "13*" by the logic circuit C_1 of FIG. 2B. However, since the distinguished record "13*" is still less than the B** blanks generated by the logic circuits A_1—A_5, due to the unfilled condition of all the memories A—E, the record "13*" is placed in the least memory A (Step 1 of FIG. 5). During the second record time, an interruption occurs, as designated I in FIG. 4. Since under the conditions existing at that record time, there are no memories partially filled with records, a B** blank is applied to the input terminal I_1 and a Z blank is applied to the input terminal I_2. The record 13* remains in the least memory during this record time as shown in the Step 2 of FIG. 5.

The input operation in the example of FIG. 4 is resumed at the third record time, and a record identified by the integer "16" is fed into the system by way of the input terminal I_1, and at the same time a Z blank is fed into the system by way of the input terminal I_2. Since the record "16" is less than the record 13*, the former is placed in the memory A and the latter is placed in the memory B as shown in Step 3 of FIG. 5. During the fourth word time, a record identified by the integer "10" is introduced into the system by way of the input terminal I_1, and a Z blank is introduced into the system by way of the input terminal I_2. The record 10 being the least record in the system is altered by the logic circuit C_1 of FIG. 2B, and the resulting record 10* is placed in the memory C, the records 13 and 16 being placed in the memories A and B respectively, as shown in Step 4 of FIG. 4.

Another interruption I occurs at the fifth record time, as shown in FIG. 4. However, a condition existed during the previous word time in which the memory C was only partially filled with records (Step 4 of FIG. 5). Therefore, in order to prevent a B** blank from being shifted into the memory C behind the 10* record, with a resulting deleterious effect on the sorting process, the condition is detected, and for the particular interrupt phase of the fifth word time, Z blanks are applied to both the input terminals I_1 and I_2. This causes the record 16 to shift into the memory C adjacent the record 10*, the record 13 to shift into the memory B, and a Z blank to shift into the least memory A (Step 5 of FIG. 5).

The aforesaid interruption of the fifth record time is continued to the sixth record time. However, during the fifth

record time there were no partially filled memories, so that a normal interruption control is provided for the sixth record time, whereby a B** blank is applied to the input terminal I_1 and a Z blank is applied to the input terminal I_2. This control produces a rearrangement of records, as shown in Step 6 of FIG. 5, but the Z blank remains in the least memory A. It will be remembered that the C_1 logic circuit is ineffective to alter the Z blanks coming out of the least memory A.

During the seventh record time, a record 4 is fed into the system by way of the input terminal I_1. This resumption of the inputting of records occurs at a record time (Step 6 of FIG. 5) when there are no memories partially filled with records. Then, since there is a Z blank in the least memory A, a B** blank is now applied to the input terminal I_2. The record "4" now displaced the Z blank in the least memory A, and the Z blank is forced out of the system by way of the output terminal O_2, since it is less than the record "4."

During the eighth record time, a record "3" is applied to the system by way of the input terminal I_1(FIG. 4). Now, since there are no Z records in the least memory, a Z blank is applied to the input terminal I_2. The record "3" is altered to 3* by the logic circuit C_1, and it is placed in the memory D. The other records are distributed in the memories A, B and C, as shown by the eighth step in FIG. 5.

During the ninth record time, a record "7" is placed in the system by way of the input terminal I_1. The conditions are the same as in the previous word time. That is, there is no Z blank in the least memory (Step 8 of FIG. 5). Therefore a Z blank is applied to the input terminal I_2. The record "4" is now altered to 4*, since the 3* record remains in the D memory. The resulting 4* record is placed in the D memory adjacent the 3* record.

During the tenth record time another interruption occurs, as shown by the designation I in FIG. 4. Again, the condition of a partially filled memory exists, in this case in the D memory (Step 9 in FIG. 5), so that again Z blanks are applied to both the input terminals I_1 and I_2. This again causes a Z blank to appear in the least memory A, and the record "13" is shifted into the D memory.

The aforesaid interruption continues during the 11th record time, as shown in FIG. 4. During this latter record time, the condition of the partially filled D memory persists (Step 10 in FIG. 5), so that again Z blanks are applied to both the input terminals I_1 and I_2. The D memory is now filled (Step 11) as the record "16" is shifted into that memory, and Z blanks now appear in both the A and B memories.

The aforesaid interruption (I) continues during the 12th record time. However, there are now no partially filled memories (Step 11 of FIG. 5), so that there is no need to feed another Z blank into the system. Instead, a normal control is exerted, whereby a B** blank is applied to the input terminal I_1 and a Z blank is applied to the input terminal I_2. The records become rearranged, as shown in Step 12 of FIG. 5, but the Z blanks remain in the A and B memories.

During the ninth record time, a record "7" is placed in the system by way of the input terminal I_1. The conditions are the same as in the previous word time. That is, there is no Z blank in the least memory (Step 8 of FIG. 5). Therefore, a Z blank is applied to the input terminal I_2. The record "4" is now altered to 4*, since the 3* record remains in the D memory. The resulting 4* record is placed in the D memory adjacent the 3* record.

During the 10th record time another interruption occurs, as shown by the designation I in FIG. 4. Again, the condition of a partially filled memory exists, in this case in the D memory (Step 9 of FIG. 5), so that again Z blanks are applied to both the input terminals I_1 and I_2. This again causes a Z blank to appear in the least memory A, and the record "13" is shifted into the D memory.

The aforesaid interruption continues during the 11th record time, as shown in FIG. 4. During this latter record time, the condition of the partially filled D memory persists (Step 10 in FIG. 5), so that again Z blanks are applied to both the input terminals I_1 and I_2. The D memory is now filled (Step 11) as the record "16" is shifted into that memory, and Z blanks now appear in both the A and B memories.

The aforesaid interruption (I) continues during the 12th record time. However, there are now no partially filled memories (Step 11 of FIG. 5), so that there is no need to feed another Z blank into the system. Instead, a normal control is exerted, whereby a B** blank is applied to the input terminal I_1 and a Z blank is applied to the input terminal I_2. The records become rearranged, as shown in Step 12 of FIG. 5, but the Z blanks remain in the A and B memories.

During the 13th record time, the inputting of records is resumed and a record "1" is applied to the input terminal I_1. The condition now exists of a Z blank being contained in the least memory A (FIG. 5, Step 12), but there is no partially filled memory. Therefore, a B** blank is applied to the input terminal I_2. The remaining Z blank is now placed in the least memory A, the record "1" is placed in the memory B, and the other records are rearranged in the manner shown in Step 13 of FIG. 5. It will be observed that as the altered records 4* and 3* leave the D memory they are returned to their unaltered 4 and 3 form, this being achieved by the logic circuit A_1—A_5, as will be described. It will also be observed that the Z blanks are not altered as they leave the Z memory.

During the 14th record time the inputting of records is continued, and a record "2" is fed to the input terminal I_1. We now have the same condition as in the previous word time in that there are no unfilled memories, and a Z blank exists in the least memory A (Step 13 of FIG. 5). Therefore, again a B** blank is applied to the input terminal I_2. This displaced the last Z blank out of the least memory A, and the records are now arranged in the memories in the manner shown in Step 14 of FIG. 5.

For the 15th record time, a record "5" is input to the system by way of the input terminal I_1 (FIG. 4). There is now no Z blank in the least memory A, and there is no unfilled memory. Therefore, normal input phase conditions are restored, and a Z blank is fed to the input terminal I_2. The record "1" is altered by the logic circuit C_1, and it becomes 1*, the greatest record. The record 1* is placed in the E memory, as shown in Step 15 of FIG. 5. The other records are distributed in the manner also known in Step 15 of FIG. 5.

The normal input operation continues during the record times 16—18, with the records "1," "12" and "11" being successively placed in the system by way of the input terminal I_1, and with Z blanks being successively applied to the input terminal I_2. Another interruption occurs at word time 19. During the latter word time the memory E is unfilled, and in order to prevent a B** blank from being fed into that memory, Z blanks are applied to both the input terminals I_1 and I_2. This creates the condition shown in Step 19 of FIG. 5.

The latter interruption phase continues through the record times 20 and 21, and since the same conditions prevail as during the word time 19, the Z blanks are still applied to both the input terminals I_1 and I_2 during each of these word times. Therefore, at the 21st record time, there are Z blanks in both the A and B memories, and there is a Z blank in the C memory adjacent the record 6.

Now, when the input operation is resumed at the record time 22, a record "8" is applied to the input terminal I_1 (FIG. 4). Since there is no Z blank in the least memory A, and since there is an unfilled memory E, a Z blank is applied to the input terminal I_2. This causes the records and the Z blanks to assume the positions shown in Step 22 of FIG. 7.

In accordance with the present invention, the inputting operation is continued until the sorter system reaches capacity, and overflows, as described above. The subsequent steps for achieving this are shown, for example, in the examples of FIGS. 6 and 7, and in FIGS. 8 and 9. In the example of FIGS. 6 and 7, for example, data continues to be introduced for the following 23—26 record times, whereas in the example of FIGS. 8 and 9, no further data is introduced, but the data already in the machine is outputted from the system during the following 37—56 word times.

During the 23, 24 and 25th record times in the example of FIG. 6, records identified by the integers 9, 24 and 25 respectively are introduced to the sorter by way of the input terminal I_1, and the B** blanks are introduced to the sorter by way of the input terminal I_2. The sorter is filled during these three steps, and no information appears at its output terminals. Instead, the Z blanks appear at the output terminal 0_1 and the B** blanks appear at the output terminal 0_2.

In accordance with the invention, information continues to be fed into the sorting system despite the fact that it is already filled. Therefore, during the 26th record time, a record identified by the integer 15 is applied to the input terminal I_1 and the B** blank is applied to the input terminal I_2. As the record 15 enters the sorter, the lowest record 1 is displaced and appears at the output terminal 0_1, whereas the B** blank appears at the output terminal 0_2, and the information in the sorter assumes the positions shown, for example, in FIG. 7. During the 27th record time in the example under consideration, an interrupt occurs and a Z blank is applied to the input terminal I_1, with a B** blank applied to the input terminal I_2. This causes the Z blank to appear at the output terminal 0_1, and the B** blank to appear at the output terminal 0_2. Again, the information in the sorting system assumes the configuration shown in FIG. 8.

During the 28th record 2a record identified by the integer 19 is fed into the system by way of the input terminal I_1, and the B** blank is fed into the system way way of the input terminal I_2. This causes the information in the system to assume the positions shown in FIG. 7, with the record 2 being displaced out from the output terminal 0_1 and the B** blank appearing at the output terminal 0_2. The process continues in the manner shown in FIG. 6, with B** blanks being applied to the input terminal I_2 whenever a record is applied to the input terminal I_1, and with a Z blank being applied to the input terminal I_1 during an interrupt interval, with a corresponding B** blank being applied to the input terminal I_2.

As shown in FIG. 6, whenever a record is introduced into the sorting system, another record is displaced from the output terminal 0_1. Also, the records displaced from the output terminal 0_1 are in a generally sorted condition. In fact, in the example of FIGS. 6 and 7, and FIGS. 8, 9, the records appearing at the output terminal 0_1 appear in a fully sorted condition. However, for many files, the records, although generally sorted at the end of the first pass, are in a condition such that some of the records may be displaced, and this requires subsequent passes through the system, as will be described.

In the steps of FIGS. 8 and 9, and as mentioned above, although the input operation is terminated, the system continues to displace records through the output terminal 0_1, until all the records in the sorter have been displaced in a generally sorted condition out of the system. In order to achieve this, and as shown in FIG. 8, a B** blank is applied to the input terminal I_1 and a B** blank is applied to the input terminal I_2 whenever it is desired to displace a record out of the system, from the output terminal 0_1. The outputting of the records may be interrupted at any time, such as shown during the record times 41, 45, 48 and 49, by introducing a Z blank to the input terminal I_2, instead of a B** blank. The Z blank, being smaller than the lowermost record, is then displaced from the output terminal 0_1, instead of a record, as shown. During this latter operation, the records in the sorting system assume the positions shown in FIG. 9 for each successive step, until all the records have been displaced, and until only B** blanks remain.

Therefore, the basic steps involved by the sorting operation of the present invention includes the first step of filling the sorter with records. This is achieved, as described above, by feeding the unsorted records of the file successively to the system at the input terminal I_1, and by feeding Z blanks to the system concurrently at the input terminal I_2. In the interruption intervals during this initial filling operation, and as described above, B** or Z blanks are fed to the input terminal 1 1 and Z blanks or B** blanks are fed to the input terminal I_2, in accordance with the previously described formula.

After the sorting system has been filled, the remaining records of the file to be sorted are fed successively into the sorting system by way of the input terminal I_1, and as explained in conjunction with FIGS. 6 and 7. During this operation, the records are fed into the sorting system by way of the input terminal I_1, and the B** blanks are fed into the system by way of the input terminal I_2. This operation continues until the end of the file, with each record inputted into the system displacing generally the lowest record out from the system, as explained. During any interruption in the file during this latter operation, Z blanks are fed to the system by way of the input terminal I_1, and B** blanks continue to be fed to the input terminal I_2. During these interruptions, the Z blanks, rather than the records in the system appear at the output terminal 0_1, as previously explained.

When the end of the file is reached, and as shown in the example of FIGS. 8 and 9, for example, the system continues to displace the records at the output terminal 0_1. For this latter operation, the B** blanks are applied to both the input terminal $I_{10}$and I_2. Whenever these blanks are applied to both the input terminals, the generally lowermost record is displaced from the output terminal 0_1. An interruption may be achieved during this output operation by introducing a Z blank, rather than a B blank for each interruption record time to the input terminal I_2.

When the end of the file has been reached at the output terminal 0_1, some means must be provided to indicate whether or not the file has been completely sorted, so that a second pass may be carried out in the event the file is not sorted. This may be achieved in a manner to be described.

Returning now to FIGS. 2A and 2B, it will be appreciated that during the input filling operation, each of the logic circuits A_1—A_5 serves to supply B** blanks to the input of the sorter S so long as the corresponding memory A—E is not filled with records. This, as mentioned, saves the time which would otherwise be required to prepare the system to receive new data. Each of the logic circuits F_1—F_5 detects the first output record fed into the corresponding memory A—E and sets a flip-flip N_i which indicates when the corresponding memory contains at least one record. Each of the logic circuits A_1—A_5 then responds to the first record outputted by a corresponding memory A—E to set a flip-flop D_9 to indicate that the particular memory is full of records. The circuits A_1—A_5 each also serves to remove the distinction appended to the record output of the corresponding memory during the input filling operation, if a distinction has been affixed to a particular record.

During the aforesaid input operation, the logic circuits B_1—B_5 of FIG. 2A and C_2—C_5 of FIG. 2B are inactive. The logic circuit C_1 is active during the input operation, in that it appends a distinction to records fed into the memory A, as explained. As also mentioned, distinctions are not applied to the Z blanks when they are fed into the memory A during the input operation.

The logic circuits B_1—B_5 serve during the output operation to restore the distinction to a corresponding record under certain conditions. If an undistinguished record enters any of the logic circuits B_1—B_5 and the memory whose output is to be compared with that record contains a record in addition to that output, the record as it appears at the output of the corresponding one of the logic circuits B_1—B_5 is distinguished.

The sorting units of the sorter S are disabled at the end of the input operation. Then, due to the connections of the sorting units in the sorter S, as shown in FIG. 3, the records which enter the sorter when it is inhibited are not rearranged, and appear in the same order at its output terminals. The logic circuits C_1—C_5 serve to detect a distinguished data output from any of the memories. If such a distinguished data output enters one of the logic circuits C_1—C_5, the corresponding sorter S'_1 and S'_5 is inhibited from exchanging its outputs.

The logic circuits F_1—F_5 serve to detect an undistinguished record output. If an undistinguished record output enters any of the logic circuits F_1—F_5, and a record is in the memory A—

E whose record is to be compared therewith, the corresponding logic circuit F_1—F_5 appends a distinction to its output.

A typical sorter circuit is shown in FIG. **10**. This sorter circuit is appropriate, for example, for the sorters S_0—S_5, for the sorters S'_1—S'_5, and for the individual units of the sorter S in FIGS. **2** and **4**. The particular sorting circuit shown in FIG. **11** is similar to that described in the aforesaid U.S. Pat. No. 3,399,383. However, as explained in the patent, other types of sorting circuits, including one using a three-state memory, as also described in the patent, may be used.

The sorting circuit of FIG. **10** includes a first input terminal A which receives, for example, a first record A; and it includes a second input terminal B which receives, for example, a second record B. The input terminal A is connected to an "AND" gate **112** and to an inverter network **119**. The input terminal B is connected to an "AND" gate **113**, and to an inverter network **120**. The inverter network **119** responds in known manner to the input record A to produce its complement $\overline{A}$ on a bit-by-bit basis; and the inverter network **120** responds to the record B to produce its complement $\overline{B}$ on a bit-by-bit basis.

The inverter network **119** is connected to an "AND" gate **121**, and the inverter network **120** is connected to an "AND" gate **122**. The "AND" gates **112** and **113** are connected to an "OR" gate **114**. The "OR" gate **114** is connected to the output terminal designated "Hi." The "AND" gates **121** and **122** are connected to an "OR" gate **123**. The "OR" gate **123** is connected to an inverter network **125** which, in turn, is connected to the "Lo" output terminal.

The input terminal A and the output terminal of the inverter **120** are connected to an "AND" gate **135**. This means that the binary coded records A and $\overline{B}$ are introduced to the "AND" gate **135**. The input terminal B and the output of the inverter **119** are connected to an "AND" gate **137**. Therefore, the binary coded records B and $\overline{A}$ are applied to the "AND" gate **137**. Appropriate bit timing clock pulses derived from the storage medium **12**, in the manner described above, are also applied to the "AND" gates **135** and **137** for bit timing purposes.

The "AND" gate **135** is connected to the set input terminal of an inhibit flip-flop Q_1. The "AND" gate **137** is connected to the set input terminal of an exchange flip-flop Q_3. Prior to each input operation, reset pulses are applied to the input terminals of the flip-flops Q_1—Q_3 to reset the flip-flops. The reset output terminal $\overline{Q}_1$ of the inhibit flip-flop Q_1 is connected to the "AND" gates **113** and **121**, and also to the "AND" gate **137**. This output terminal $\overline{Q}_1$ of the flip-flop applies the term $\overline{Q}_1$ to the "AND" gates **113**, **121** and **137**. The reset output terminal $\overline{Q}_3$ of the exchange flip-flop Q_3 is connected to the "AND" gates **112** and **122**, and to the "AND" gate **125**. This output terminal of the exchange flip-flop applies the term $\overline{Q}_3$ to the "AND" gates **112**, **122** and **135**.

As noted above, in order for the system of FIG. **3** to perform its sorting function, it is necessary for the records A and B to be applied to the sorting system in a bit-by-bit serial manner, with the most significant bit of the identification field of each in the lead. At the commencement of the sorting operation, both the inhibit flip-flop Q_1 and the exchange flip-flop Q_3 are reset, so that both the terms Q_1 and Q_3 are false. Therefore, at the beginning of the operation, the "AND" gates **112**, **113**, **121** and **122** are all enabled. The records A and B pass through the "AND" gates on a serial bit-by-bit basis, so long as the corresponding bits of the two records are both either 1 or 0. These bits then appear unchanged, at the output terminals "Hi" and "Lo."

Assume that a bit in the identifying field of the record A is 0 at the time when the corresponding bit in the identifying field of the record B is a 1, and that this is the first instance that a condition of one record having a bit different than the corresponding bit of the other record has occurred, this means that the identifying signal of the record B represents a higher integer than the identifying signal of the record A. Under this condition, the term $\overline{A}.B$ is true, so that the "AND" gate **137** is enabled. The next clock pulse CL sets the exchange flip-flop Q_3, so that the "AND" gates **112** and **122** are disabled. Also, the "AND" gate **135** is disabled, so that no further triggering of either of the flip-flops Q_1 or Q_3 is possible during the particular comparison operation.

Following the detection of a 0 bit in the identifying field of the A record, and the simultaneous detection of a 1 bit in the identifying field of the B record, indicating that the B record is greater than the A record, the remaining bits of the control and data fields of the data record pass through the enabled "AND" gate **122** in complemented ($\overline{A}$) form, and through the "OR" gate **123** and the inverter **125** to the output terminal "Lo."

The corresponding bits of the remaining portions of the identifying signals and the bits in the data field of the B record pass through the enabled "AND" gates **112** and **114** to the output terminal "Hi." Therefore, the detection of inequality between the identifying signals in the identifying fields of the A records causes the lower A record to appear at the output terminal "Lo" and the higher B record to appear at the output terminal "Hi."

Conversely, should the comparison described above reach a point at which the identifying signal of the Z record has a 1 bit at the time when the identifying signal of the B record has a 0 bit indicating that the A record is greater than the B record, the term A.B is applied to the "AND" gate **135** to enable the "AND" gate, so that the next clock pulse sets the inhibit flip-flop Q_1.

This setting of the inhibit flip-flop Q_1 causes the "AND" gates **113** and **121** to become disabled, and also causes the "AND" gate to become disabled. The disabling of the "AND" gate **137** prevents any setting of the exchange flip-flop **123** during the remaining portion of the comparison process for the particular records under consideration.

Under the latter set of conditions, the greater record A is passed through the "AND" gate **112** and through the "OR" gate **114** to the "Hi" output terminal; whereas, the lower record B is passed through the inverter **120**, through the "AND" gate **122**, and through the "OR" gate **123** to the inverter **125**. This latter record B is reinverted to the inverter **125**, and it appears at the "Lo" output terminal in its original form.

As noted above, the sorting network of FIG. **10** can be used for the sort units S_0—S_5 and S_1'—S_5' and for the units making up the sorter S.

For convenience, the bit timing of the control field bits of the different records and blanks used in the system are shown in FIG. **11**.

The flip-flops involved in the system are as follows:

Input Mode (I.M.) (FIG. **12**)—Indicates the input operation;

Data Input (D) (not illustrated—Controlled by the system introducing records to the sorter;

Part Full (P.F.) (FIG. **13**)—Indicates when any of the memories A—E in the system has a data input with a B** output;

End of File (E.O.F.) (FIG. **12**)—Indicates that the last record in the file has been fed into the system;

Exchange Indicator (R) (FIG. **13**)—Detects exchanges in sorters during each pass of the information through the system;

Sorter Full (S.F.) (FIG. **13**)—Indicates that the sorter is full of records, so that the output operation may begin;

Z Blank Detector (Z1) (FIG. **13**)—Detects the presence of Z blanks in the memory A;

B Blank Detector (B1) (FIG. **13**)—Detects B blanks in the memory A to indicate the completion of a pass through the sorter;

Output Mode (O.M.) (FIG. **12**)—Indicates that an output operation is taking place;

Data Indicator (N_f) (N_3—N_5) (one of each memory except A and B) (FIG. **14**)—Indicates that a record has been introduced to the corresponding memory;

Memory Full Indicator (D_i) (D_1—D_5) (one for each memory) (FIG. 13)—Indicates that the corresponding memory is full of records by detecting a record at the output of the memory and, during the output operation the flip-flop indicates the presence of an undistinguished record in the corresponding memory.

As shown in FIG. 12, the input mode flip-flop I.M. has an "AND" gate **300** connected to its set input terminal, and an "AND" gate **302** connected to its reset input terminal. The terms D and t_1 are introduced to the "AND" gate **300**, and the terms E.O.F. and t_1 are introduced to the "AND" gate **302**. This means that the input operation of the sorter is commenced, by the setting of the input mode flip-flop, under an appropriate command from the system applying the records to the sorter, which sets the aforementioned flip-flop D (not shown). The input operation is terminated, and the flip-flop I.M. is thereby reset, when the end of file indication ($\overline{D}$) from the system applying the records causes the end of file flip-flop E.O.F. to be set.

The end of file flip-flop E.O.F. has an "AND" gate **403** connected to its set input terminal, and an "AND" gate **306** connected to its reset input terminal. The terms I.M., $\overline{D}$ and t_1 are introduced to the "AND" gate **304**; and the terms B_1 and t_1 are introduced to the "AND" gate **306**. This means that the end of file flip-flop E.O.F. is set, so as to terminate the input operation, upon the receipt of the end of file ($\overline{D}$) command from the system feeding the records to the sorter, and assuming that the sorter is in its input operation (I.M.). The end of file flip-flop E.O.F. remains set until the B blank indicator flip-flop B_1 detects blanks in the memory A to indicate the completion of the output operation.

The output mode flip-flop (O.M.) includes an "AND" gate **308** connected to its set input terminal, and an "AND" gate **310** connected to its reset input terminal. The terms E.O.F. and t_2 are applied to the "AND" gate **308**, and the terms B_1 and t_1 are applied to the "AND" gate **310**. The output mode is initiated when the output mode flip-flop O.M. is set, this occurring when the sorter becomes full of records, as indicated by the sorter full flip-flop S.F. being set. The output mode then continues until a blank is detected in the least memory A, as indicated by the setting of the blank indicator flip-flop B_1.

The exchange flip-flop R in FIG. 13 includes an "AND" gate **330** connected to its set input terminal, and an "OR" gate **332** connected to its reset input terminal. The exchange flip-flop is set by any blank or distinguished record out of any of the memories A—E, due to the terms L_i and t_2 applied to the "AND" gate **330**. Then, the terms L'_i, $\overline{t_2}$ and E_i applied to an "AND" gate **334**, one of which is provided for each sorter, and which are connected to the "OR" gate **332**, cause the exchange flip-flop R to be reset should a record exchange take place in any one of the sorters. Should no exchange take place, the exchange flip-flop R remains set during the particular pass.

The sorter full flip-flop S.F. responds to the terms D_1—D_5 applied to an "AND" gate **336** to be set and indicate that all the memories in the sorter contain records, and this flip-flop serves to set the output mode flip-flop O.M. to put the system in its output operation after it is filled with records. The setting of the output mode flip-flop serves to reset the sorter full flip-flop S.F. The S.F. flip-flop indicates that the system cannot accept any more records, although under some circumstances there still may be Z blanks in the system.

The Z_1 flip-flop is used to detect Z blanks in the memory A, so that these blanks will not be distinguished during the input operation. The Z_1 flip-flop includes an "AND" gate **340** connected to its set input terminal. The set output terminal of a flip-flop **342** is connected to the "AND" gate **340**; and the term t_3 and $\overline{L_1'}$ are applied to the "AND" gate. An "AND" gate **344** is connected to the set input terminal of the flip-flop **342**, and the terms $\overline{L_1'}$ and t_2 are applied to the "AND" gate **344**. The term t_4 resets the flip-flop **342**, and the term t_1 resets the term Z_1.

The presence of a Z blank in the A memory sets the flip-flop **342** to enable the "AND" gate **340**, and thereby permit the Z_1 flip-flop to be set.

The flip-flop B_1 detects B blanks in the least memory A. This flip-flop is set each t_1 time by the term t_1 applied to its set input terminal. An "OR" gate **346** is connected to its reset input terminal, and "AND" gates **343**, **345** and **347** are connected to its reset input terminal. The terms L'_1 and t_1 are applied to the "AND" gate **343**; the terms $\overline{L'_1}$ and t_2 are applied to the "AND" gate **345**; and the terms L'_1 and t_1 are applied to the "AND" gate **347**. The flip-flop B_1 remains set only if a B blank is fed into the memory A.

As mentioned above, each of the memories A—E includes a D_i flip-flop, and the memories C—E each include a N_i flip-flop. When both the N_i and D_i flip-flops are reset, the corresponding memory is empty. When the flip-flop N_i is set, it means that the corresponding memory is partially full of records. When both the flip-flops N_i and D_i are set, it means that the corresponding memory is full of records. As also mentioned above, the flip-flops N_i and D_i are used to indicate the presence of undistinguished records in the output mode.

As shown in FIG. 14, the flip-flop B_1 is associated with the memory A, and the flip-flop B_2 is associated with the memory B. Since the memories A and B are "one word" memories, there is no need for the part full flip-flop N_i to be used in conjunction therewith. The flip-flops D_3 and N_3 are associated with the memory C; the flip-flops D_4 and N_4 are associated with the memory D_4; and the flip-flops D_5 and N_5 are associated with the memory E.

An "AND" gate **350** is connected through an "OR" gate **351** to the set input terminal of the flip-flop D_i; and an "AND" gate **352** is connected to the reset input terminal of that flip-flop. An "AND" gate **353** is connected to the "OR" gate **351**. The terms O.M., $\overline{L_1'}$ and t_2 **2** applied to the "AND" gate **353**. The terms I.M., $\overline{L_1}$ and t_1 are applied to the "AND" gate **350**; and the terms O.M., $\overline{L_1}$ and t_2 are applied to the "AND" gate **352**. This means that whenever a record is outputted from the memory A during the input operation, the terms $\overline{L_1}$, t_1 set the flip-flop D_1. Conversely, whenever an undistinguished record is outputted from the memory A during the output mode, the terms $\overline{L_1}$, t_2 reset the flip-flop D_1.

An "AND" gate **354** is connected through an "OR" gate **355** to the set input terminal of the flip-flop D_2, and an "AND" gate **356** is connected to the reset input terminal. An "AND" gate **357** is connected to the "OR" gate **355**. The terms O.M., L_1' and t_2 are applied to the "AND" gage **357**. The terms I.M., L_2, D_1 and t_1 are applied to the "AND" gate **354**, and the terms O.M., L_2 and t_2 are applied to the "AND" gate **356**. This means that whenever a record B is outputted from the memory B during the input operation, the term L_2-t_1 sets the flip-flop D, whereas whenever an undistinguished record is outputted from the memory B during the output mode, the term L_2, t_2 resets the flip-flop D_2.

With respect to the flip-flop D_3, an "OR" gate **358** is connected to its set input terminal, and a pair of "AND" gates **360** and **362** are connected to the "OR" gate **358**. An "AND" gate **364** is connected to the reset input terminal of the flip-flop D_3. The terms O.M., L_3' and t_2 are supplied to the "AND" gate **360**; whereas the terms I.M., L_3, t_1, N_3 and D_2 are applied to the "AND" gate **362**. The terms O.M., L_3 and t_2 are applied to the "AND" gate **364**.

The flip-flop D_3 is set, by the output from the "AND" gate **362** during the input operation whenever a B record is derived from the memory C, indicating that the memory C is full of records. Also, the "AND" gate **360** is enabled during the output mode when an undistinguished record is fed into the memory C, so as to set the flip-flop D_3.

The flip-flop N_3 associated 13)—the memory C, includes an "OR" gate **366** connected to its set input terminals, and a pair of "AND" gates **368** and **370** connected to the "OR" gate. An "AND" gate **372** is connected to the reset input terminal of that flip-flop. The terms O.M., $\overline{L_3'}$ and t_2 are applied to the "AND" gate **368**; the terms I.M., D_2 and $\overline{L_3'}$ and t_1 are applied

to the "AND" gate **370**; and the terms O.M., t_3, $\overline{L_3}$ and $\overline{D_3}$ are applied to the "AND" gate **372**.

Therefore, the flip-flop N_3 is set during the input operation, assuming that the flip-flop D_2 of the memory B has been set to indicate that the memory B is full of records, this being a condition which must be fulfilled before the memory C can be partially full, and the inputting of a D record into the memory C causes the "AND" gate **370** to become enabled, so as to set the flip-flop N_3 to indicate that the memory C does have data in it. It will be remembered that when the flip-flop N_3 is set, and the flip-flop D_3 is reset, the indication is that the memory C is partially filled with records. The flip-flop D_3 is also set when the memory C is full of records. Whenever an undistinguished record is introduced to the memory C during the output operation, the gate **368** is enabled, so as to set the flip-flop N_3.

It will be appreciated that the flip-flops D_4 and N_4 associated with the memory D have similar logic associated with them as that described in conjunction with the flip-flops D_3 and N_3. It will also be appreciated that the flip-flops D_5 and N_5 associated with the memory E, have similar logic associated with them as the logic associated with the flip-flops D_4, N_4 and D_3, N_3.

The logic diagrams of FIGS. **15** and **16** show the logic associated with the sorters S_1, S_2, $S_3$and S_0, S_1', S_2' and S_3', specifically the logic circuits A_1, A_2, A_3, B_1, $B_2$and B_3, C_1, C_2 and C_3 and F_1, F_2 and F_3. It will be appreciated that similar logic circuitry is associated with the sorters S_4 and S_5, and with the sorters S_3', S_4' and S_5'. The later logic circuits are largely repetitive, and have been omitted for purposes of clarity.

The sorters S_1, S_2 and S_3 (FIG. **15**) have a first set of "OR" gates **400**, **402** and **404**, respectively connected to their respectively lower input terminals. These "OR" gates are respectively connected in the logic circuits A_1, A_2 and A_3. The sorters S_1 and S_2 have respective "OR" gates **406** and **410** (FIG. **15**) connected to their respective upper input terminals, the latter "OR" gates being respectively included in the logic circuits B_2 and B_3. Likewise, the sorter S_0 (FIG. **16**) has an "OR" gate **412** connected to its upper input terminal, the latter "OR" gate being included in the logic circuit B_1.

Also in FIG. **16**, the sorter S_1' has an "OR" gate **414** connected to its lower input terminal, the latter "OR" gate being included in the logic circuit C_1. In FIG. **15**, a plurality of "AND" gates **420**, **422**, **428** and **432** are connected to the "OR" gate **404**. The terms I.M., $\overline{t_2}$, L_3 and D_3 are introduced to the "AND" gate **420**, the term L_3being the output from the memory C in FIG. **2A**. The terms $\overline{N_3}$, I.M. and B** are applied to the "AND" gate **422**. The terms L_3, O.M. and $\overline{t_2}$ are applied to the "AND" gate **428**. The terms O.M., L_3, N_3, $\overline{D_3}$and t_2 are applied to the "AND" gate **432**.

A plurality of "AND" gates **434**, **436**, **442** and **446** are connected to the "OR" gate **402**. The terms I.M., $\overline{t_2}$, D_2 and L_2 are applied to the "AND" gate **434**, the term L_2 being the output from the memory B of FIG. **2A**. The terms $\overline{D_2}$, I.M. and B** are applied to the "AND" gate **436**. The terms O.M., L_2 and $\overline{t_2}$ are applied to the "AND" gate **442**. The terms O.M., $\overline{D_2}$, $\overline{t_2}$ and L_2 are applied to the "AND" gate **446**.

A pair of "AND" gates **452** and **458** are connected to the "OR" gate **410** in the logic circuit B_3. The output from the lower terminal of the sorter S_3, and the term I.M. are applied to the "AND" gate **452**. These connections form a circulating path through the B_3 logic during the input mode. The terms O.M., E_3, N_3 and t_2 are applied to the "AND" gate **458**.

The logic circuit B_2 includes a pair of "AND" gates **462** and **468** connected to the "OR" gate **406**. The output from the lower terminal of the sorter S_2 is applied to the "AND" gate **462**. The term I.M. is also applied to the "AND" gate **462** so as to provide a path through the B_2 logic during the input mode. The terms O.M., E_2, N_2 and t_2 are applied to the "AND" gate **468**.

A plurality of "AND" gates **470**, **472**, **478** and **482** are connected to the "OR" gate **400**. The output from the memory A (L_1) is applied to the "AND" gates **470**, **478** and **482**. The terms $\overline{t_2}$, D_1 and I.M. are also applied to the "AND" gate **470**. The terms $\overline{D_1}$, I.M. and B** are applied to the "AND" **472**. The terms L_1, O.M. and $\overline{t_2}$ are applied to the "AND" gate **478**. The terms O.M., $\overline{D_1}$ and $\overline{t_2}$ are also applied to the "AND" gate **482**.

The logic circuit B_1 in FIG. **16** includes a plurality of "AND" gates **492**, **496** and **498** connected to the "OR" gate **412**. The output from the lower terminal of the sorter S_1 of FIG. **15** is applied to the "AND" gates **492** and **496**. The term I.M. is applied to the "AND" gates **492** to provide a path through the B_1 logic circuit during the input mode. The term O.M. is also applied to the "AND" gate **496**. The terms O.M., E_1, N_1 and t_2 are applied to the "AND" gate **498**.

The logic circuit C_1 includes a plurality of "AND" gates **500**, **501** and **502** connected to the "OR" gate **414**. The terms I.M. and Z_1 are applied to the "AND" gate **501**. The terms I.M., $\overline{Z_1}$ and t_2 are applied to the "AND" gate **500**. The term I.M. at the output from the upper terminal of the sorter S_0 are applied to the "AND" gate **502** to maintain a path through the C_1 logic circuit during the input mode.

The output from the upper terminal of the sorter S_1' is applied directly to the lower terminal of the sorter S_2'. The output from the upper terminal of the sorter S_2' is directly connected to the inner terminal of the sorter S_3'. The lower output terminal of the sorter S_1' is connected to an "AND" gate **516** in the logic circuit F_1; the lower output terminal of the sorter S_2' is connected to an "AND" gate **518** in the logic circuit F_2; and the lower output terminal of the sorter S_3 is connected to an "AND" gate **520** in the logic circuit F_3. These "AND" gates **516**, **578** and **520** provide paths through the respective logic circuits F_1, F_2 and F_3 during the input mode.

The "AND" gate **516**, together with a pair of "AND" gates **522** and **524** in the logic circuit F_1 are connected to an "OR" gate **526**, the output of which provides the L_1' input to the memory A of FIG. **2A**. The "AND" gate **518**, together with a pair of "AND" gates **528** and **530** are connected to an "OR" gate **532**, the output of which provides the input L_2' to the memory B of FIG. **2A**. The "AND" gate **520**, together with a further pair of "AND" gates **534** and **536** in the logic circuit F_3 are connected to an "OR" gate **538**, the output of which provides the input L_3' to the memory C of FIG. **2A**.

The terms O.M., D_1 and t_2 are applied to the "AND" gate **522**; the terms t_2, O.M., $\overline{D_1}$ and E_1' are applied to the "AND" gate **524**; the terms O.M., D_2 and $t_2$are applied to the "AND" gate **528**; the terms t_2, O.M., $\overline{D_2}$ and E_2' are applied to the "AND" gate **530**; the terms O.M., D_3 and t_2 are applied to the "AND" gate **534**; and the terms t_2, N_3, E_3' and $\overline{D_3}$ and O.M. are applied to the "AND" gate **536**.

As shown by the logic circuits of FIGS. **13** and **14**, a corresponding one of the "partially full" flip-flops N_3—N_5 is set, whenever a corresponding one of the F_3—F_5 logic circuits inputs a record to a corresponding one of the memories C—E. Also, the F_1 logic circuit resets the flip-flop Z_1 when a Z blank is introduced to the memory A, so that the logic circuit C_1 will not distinguish a Z blank. The A_1—A_5 logic circuits will set corresponding "memory full" flip-flops D_1—D_5 whenever their corresponding memories input records. It will be remembered that the memories A and B are one word memories, and so do not have "partially full" flip-flops associated with them. These latter memories either have an empty condition ($\overline{D_1}$ or $\overline{D_2}$) or a full condition (D_1 or D_2). It will also be remembered that the outputs from the memories A, B, C, D and E are identified by the respective terms L_1, L_2, L_3, L_4 and L_5; whereas the inputs to the various memories are designated by the respective terms L_1', L_2', L_3', L_4' and L_5', as shown in FIGS. **2A** and **2B**.

During the input mode, the flip-flop I.M. (FIG. **12**) is set. As mentioned above, the logic circuits A_1—A_5 (FIG. **2A**) perform a variety of functions during the input mode. For example, these logic circuits apply B** blanks to the system whenever the corresponding ones of the memories A—E are empty. This latter function is achieved, for example, by the logic circuits A_1, A_2 and A_3 in FIG. **17** by means of the respective

"AND" gates **422**, **436** and **472**. In the case of the "AND" gates **436** and **472** which are associated with the one word memories A and B, respectively; whenever the D_1 flip-flop is reset during the input mode, the "AND" gate **472** passes B** blanks to the system; and whenever the D_2 flip-flop is reset during the input mode, the "AND" gate **436** passes the B** blanks to the system. In the case of the "AND" gate **422** in the logic circuit A_3, however, which is associated with a multiword memory C, it passes the B** blanks to the system during the input mode (I.M.) when the corresponding "partially full" flip-flop is reset ($\overline{N_3}$).

It will also be remembered that the logic circuits A_1—A_5 also function during the input mode to remove the distinctions (*) from the distinguished records. This is accomplished, for example, in the case of the logic circuits A_1—A_3 in FIG. **15** by the "AND" gates **420**, **434** and **470**. If the corresponding memory A, B or C produces a record at its output, the corresponding flip-flop D_1, D_2 or D_3 will be set, indicating a "full" memory condition. Also, if the particular record produced at the output of the particular memory is distinguished, the t_2 bit will be a **1** (FIG. **11**). The gates **420**, **434** and **470** are disabled at the t_2 bit time, so that the t_2 bit of each record passed thereby will be returned to 0 if the record has been distinguished.

The circuits B_1—B_5 (FIG. **2A**) serve no function during the input mode. Also, the logic circuits C_2—C_5 (FIG. **2B**) serve no function during the input mode. However, the logic circuit C_1 distinguishes the data L_1' applied to the memory A, but does not change any Z blanks which may be fed into that memory. This function is carried out by the "AND" gate **500** in the C_1 logic circuit of FIG. **16**. So long as the word applied to the "AND" gate **500** is not a Z blank, the "AND gate will cause the t_2 bit to be a 1 during the input mode, since the Z_1 flip-flop is then reset. If the word is a Z blank, the Z_1 flip-flop is set, and normal circulation is provided through the "AND" gate **501**. Therefore, any Z blank fed through the logic circuit C_1 during the input mode is passed to the sorter S_1' unchanged, whereas any D record passed to the C_1 logic circuit during the input mode is fed to the sorter S_1' in a distinguished D* condition.

When the "sorter full" flip-flop S.F. of FIG. **13** is set, the input mode continues and the input mode flip-flop I.M. remains set. However, as shown in the previous examples, the setting of the S.F. flip-flop causes B** blanks to be applied to the input terminal I_2, and records begin to appear in a generally sorted condition at the output terminal O_1. This operation continues until the end of the file is reached, at which time the E.O.F. flip-flop of FIG. **13** is set. The setting of the E.O.F. flip-flop causes the I.M. flip-flop to be reset, thereby ending the input mode; and also causes the O.M. flip-flop of FIG. **12** to be set, thereby starting the output mode. The logic of the system could be designed to maintain the circuit in the input mode throughout the entire pass. However, a more complete sort may be achieved in each pass, if the record is switched to an output mode when the end of file is reached. The following operations controlled during the output mode cause a more complete sort of the contents of the sorter to be achieved, as would be the case if it were left in its input mode.

When the output mode flip-flop O.M. of FIG. **12** is set to initiate the output mode, B** blanks are applied to both the I_1 and I_2 input terminals, as shown in the previous examples, except during record times at which no record is to be removed from the sorting system. For the latter condition, the B** blanks continue to be applied to the input terminal I_1, but Z blanks are applied to the input terminal I_2, as shown in the previous examples of FIGS. **8** and **9**.

During the output mode, whenever the last record in a memory A—E is fed out of that memory, its distinction (*) is removed and it is returned to its undistinguished (D) form. This latter operation is achieved by the gates **428**, **442** and **478**, for example, in the logic circuits A_1—A_3 of FIG. **15**, and by similar gates in the logic circuits A_4 and A_5. The flip-flops D_1—D_5 of FIG. **14** are first reset during the output mode when the least record (now undistinguished as mentioned above) is fed out of the different memories. Whenever one of the D_1—D_5 flip-flops is reset during the output mode, the (*) is removed from the next record fed out of the corresponding memory A—E. This latter operation is achieved by the "AND" gates **432**, **446** and **482** in FIG. **15**.

Subsequently when a B blank is fed out of any particular memory A—E during the output mode (O.M.), indicating that the corresponding memory is empty, the corresponding "partially full" flip-flop N_f is reset. This resetting of the N_f flip-flops is achieved by the "AND" gates **372**, **374** and **376** in FIG. **14**. The logic circuits B_1—B_5, on the other hand, replace the (*) to the record if an exchange was made in the corresponding sorter when the corresponding memory was not empty (N_f set). This is achieved, for example, by the "AND" gates **458** and **468** in FIG. **15**, and by the "AND" gate **498** in FIG. **16**.

The logic circuits F_1—F_5 function during the output mode to restore the distinction (*) to any record or blank fed out of a memory for which the D_f "Memory full" flip-flop is set; indicating that an undistinguished record has been fed into the corresponding memory. This is achieved by the "AND" gates **522**, **528** and **534** in FIG. **16**. Also, the distinction (*) is restored to any record during the output mode when the corresponding D_f flip-flop is reset and the corresponding N_f flip-flop is set, and an exchange (E'_1) was made in the corresponding sorter S_1'—S_5'. This latter operation is carried out by the "AND" gates **524**, **430** and **536** in FIG. **16**. This appending of a distinction to any of the records causes the output mode to continue, since that record does not get displaced out from the sorter.

A new pass is ordered th, the sorter th all the records, th an appropriate control of the flip-flop D (not shown). If a new pass is ordered, the flip-flop D is set, so as to set Table input mode flip-flop of FIG. **12**, and th information is again passed through the sorter. A new pass is 1 if any exchanges were made in the sorter S_0 (E_0) after the first record was outputted from the sorter system that is, after the sorter full condition was reached, by setting the sorter full flip-flop S.F. of FIG. **13**. The absence of an exchange in the sorter S_0 after the first record has been outputted from the sorter shows that none of the subsequent records fed into the sorter was less than a previous record fed out of the sorter, so that a complete sort of the records had been achieved by the particular pass.

During any pass of the sorter system of the invention, the highest memory A—E which outputs a record that is exchanged with the output L_1 of the memory A is noted. Then, during the next succeeding pass, the sorter capacity is limited, so that no more time or capacity is used for each successive pass than is absolutely necessary. One manner of achieving this is to use only the memories up to and including the one which exchanged its output with the output of the A memory during the preceding pass, and to fill the other memories with B** blanks.

Then, for each succeeding pass, the capacity of the sorter is effectively decreased until only two records are contained in the sorter at any one time, at which time the sort is complete. That is, the sort of all the records will be completed at the end of the pass in which only two records were contained in the sorter, or where no exchange flip-flop E_f greater than E_1 has been set during a pass. The effective capacity of the sorter may be reduced for each succeeding pass by means of the logic circuitry shown, for example, in FIGS. **17** and **18**. In FIG. **17**, for example, the flip-flop P_5 is set by the term E_5 indicating that no exchanges were made in the sorter S_5. When that occurs during any particular pass, for example, the flip-flop P_5 remains set during the next pass. Then, the sorter full flip-flop S.F. is set, by the logic of FIG. **17**, not when the memory D_5 becomes full of records, but when the preceding memory D_4 becomes full, so that the sorter is set to start outputting records at a reduced capacity. It will be remembered that whenever the flip-flop S.F. is set, the B** blanks are applied to the input terminal I_2, so that the sorter system starts to output its records.

In order to achieve the aforesaid control, the set logic associated with the flip-flop S.F., as shown in FIG. **18**, is slightly different from the logic shown in FIG. **13**. That is, the terms D_2, D_3, D_4 and D_5, instead of being applied directly to the "AND" gate **336**, are "OR"ed with the respective terms P_2, P_3, P_4 and P_5 in respective "OR" gates **600**, **602**, **604**, **606**.

If exchanges occurred in the other sorters S_2—S_4, for example, the flip-flops P_2—P_4 are reset. Then, the sorter full flip-flop S.F. of FIG. **18** is not set until the corresponding memory is actually filled. However, whenever the condition arises that there is no exchange in a corresponding sorter, the flip-flops P_4, P_3 and P_2 successively become set, so that in each successive pass, the sort full condition is reached by the use of fewer and fewer of the memories A—E, until only the two memories A and B contain records, with the sort being completed during the next pass.

The capacity of the sorting system can also be controlled by controlling the effective length of the individual memories A—E (L_1—L_5), for example, as will now be described. As shown in FIG. **19**, the memories A (L_1), B (L_2) and C (L_3) may be the same as in the previous embodiment, with the memories A and B having a capacity to store one record each, and with the memory C having the capacity to store two records. The memories D (L_4) and E (L_5) in the embodiment shown in FIG. **19** are made up of a plurality of two-record segments intercoupled respectively by logic circuits G_i shown in FIG. **20**.

The individual logic circuits G_i, in turn, are controlled by respective flip-flops designated X_{ab}, in FIG. **20**, these corresponding, for example, to a flip-flop X_{41} in the memory D, and the flip-flops X_{51}, X_{52} and X_{53} in the memory E. When the flip-flop X_{41} is reset, for example, only one two-record segment of the memory D is used, and when it is set, for example, both the two segment sections of the memory are used. Likewise, when all the flip-flops X_{51}, X_{52} and X_{53} are reset, only one segment of the memory E is used, with the other segments being used as the corresponding flip-flops are set.

The interconnection of the various segments of the individual memories in response to the setting of the flip-flops X_{ab} may be carried out by the logic circuit shown in FIG. **20**. A pair of segments designated **01** and **02** are shown in FIG. **20**, each, for example, having a capacity to hold two records, and which may be incorporated into any of the memories. It might be pointed out at this time that the selection of the segments to have a length corresponding to two records is merely arbitrary, and any desired number may be used. A factor to be considered is that as the number of segments is increased in any particular system, the sorting time is correspondingly decreased, but the circuit and system complexities are correspondingly increased.

The input records are fed by way of an input terminal L'_i to the input of the segment **02**. The output of that segment is applied to a pair of "AND" gates **700** and **702**. The "AND" gate **700**, together with a further "AND" gate **704**, are connected to a "NOR" gate **706**. The output of the "OR" gate is introduced to the output terminal of the circuit, designated L_i.

The "AND" gate **702**, together with a further "AND" gate **708**, are connected to an "OR" gate **710**. The output of the "OR" gate **710** is applied to the input of the segment **01**, whereas the output of the segment **01** is applied to the "AND" gate **704**. The source of blanks B** is connected to the "AND" gate **708**, as is the reset output terminal of the flip-flop X_{ab}. The reset output terminal of the flip-flop X_{ab} is also connected to the "AND" gate **700**, and the set output terminal of that flip-flop is connected to the "AND" gates **702** and **704**.

It will be appreciated that so long as the flip-flop X_{ab} is reset, the records inputted at the input terminal L'_i are passed to the output terminal L_i through the "AND" gate **700** and the "OR" gate **706**. At this time, the "AND" gate **708** is enabled, so that the B** blanks may be fed into the unused segment **01**. In this respect it should be pointed out that the "X's" in FIG. **23**, as in FIGS. **5**, **7** and **11**, represent the B** blanks.

Then, when the flip-flop X_{ab} is set, the output from the segment **02** passes through the "AND" gate **702** and through the "OR" gate **710** to the input of the segment **01**; and the output from the segment **01** is applied through the "AND" gate **704** and through the "NOR" gate **706** to the output terminal L_i. Therefore, whereas in the case of the flip-flop X_{ab} being reset, only the segment **02** is used, the setting of the flip-flop X_{ab} causes both segments to be used in a series relationship. During this set condition of the flip-flop X_{ab}, the "AND" gate **708** is disabled, so that the B** blanks are no longer fed to the segment **01**.

The diagram of FIG. **21** shows how the sorting system accepts records fed into it, in accordance with the example of FIG. **6**. As shown in FIG. **21**, the first record **13** fed into the system at the first record time is provided with a (*) and stored in the memory A. During the second step, since no record is fed into the system, the record **13*** remains in the memory A. During the third step, during which the record **16** is fed into the system, the record **16** is stored in the memory A, and the record **13*** is stored in the memory B. During the fourth step, during which the record **10** is fed into the system, the records **13** and **16** appear in the memories A and B, whereas the record **10*** appears in the memory C. These preliminary steps are similar to the steps shown in FIG. **5**. During the fourth step, since the memory C is part full, the term $N_3\overline{D_3}$ is true.

During the fifth and sixth steps an interruption occurs in the information fed into the system, and the records in the system rearrange themselves as shown in FIG. **21**. During the fifth step the memory C becomes full, so that the term $N_3 \cdot D_3$ is true. During the fifth and sixth step, the flip-flop Z_1 is set, as described above, to indicate a Z blank in the memory A, and in order that the Z blank will not be distinguished.

During the seventh step, a four is fed into the sorter system, and the records establish themselves in the manner shown in FIG. **21**. Then, during the eighth step, a three is introduced into the system and is distinguished and introduced into the first segment of the memory D. When that occurs the term $N_1 \cdot \overline{D_1} \cdot \overline{X_{11}}$ is true. Then, during the next step, a seven is introduced so that the records 4 and 3 are in the first segment of the D memory, causing that segment to be filled. When that occurs, the term $N_4 \cdot D_4 \cdot \overline{X_{41}}$ is set true.

Then, during the 10th 11th and 12th steps, an interruption occurs in the input data, so that the records recirculate in the sorting system to assume the positions shown in the Table of FIG. **21**. During the 13th step the 1 record is fed into the system and it is placed in the first segment of the E memory, since it is distinguished as 1*. For this latter condition, the term $N_5 \cdot \overline{D_5} \cdot \overline{X_{51}}$ is true. For the next step, the record 2 is fed into the system, and it is placed in the first segment of the E memory, along with the 1 record, since both are distinguished (2* and 1*). The first segment of the E memory now becomes filled, so that the term $N_5 \cdot D_5 \cdot \overline{X_{51}}$ is true.

The number 5 record is fed into the system during the next step, and for this condition the second segment of the D memory is used (X_{41}) so that the information assumes the positions shown at step 15. For this latter condition the memory D is not full, so that the term $N_4 \cdot \overline{D_4} X_{41}$ is true.

For the next step, the 12 record is fed into the system, and now the D memory is filled, and a second segment of the E memory is activated. We now have the condition in which the terms $N_4 \cdot D_4 X_{41}$, and $N_5 \cdot \overline{D_5} \cdot \overline{X_{52}} \cdot X_{51}$ are true.

For the 18th step, the record of 11 is fed into the system and the second segment of the E memory becomes full. Now we have the condition in which $N_5 \cdot D_5 \cdot \overline{X_{52}} \cdot X_{51}$ is true.

The information is recirculated for the next three steps, and it assumes the positions shown in the table of FIG. **21**. For the 22nd step, the record 8 is fed into the system, and the third segment of the E memory is activated Now we have the condition in which the term $N_5 \cdot D_5 X_{51} \cdot X_{52} \cdot \overline{X_{53}}$ is true.

The control of the flip-flops X_{ab} during the first pass is represented by the logic diagram of FIG. **22**. The logic diagram of FIG. **22** shows the control of the flip-flops X_{41} and X_{51},

it will be understood that a similar control is exerted on the flip-flops X_{52} and X_{53}. As shown in FIG. 22, the flip-flop X_{41} is set when the term $N_4 \cdot D_4 \overline{X_{41}}$ is true to indicate that the first segment of the memory D is full; and also when the term $N_5 \cdot D_5 \overline{X_{51}}$ is true to indicate that the first segment of the memory E is also full. It will be remembered from the examples shown in FIG. 21, it is under those conditions that the flip-flop X_{41} is set to bring the second segment of the D memory into the system.

Likewise, the flip-flop X_{51} is set to bring the second segment of the E memory into the system, when the term $N_4 \cdot D_4 X_{41}$ is set to indicate that the D memory is full, and also when the term $N_6 \cdot D_6 \overline{X_{61}}$ is true to indicate that the first segment of the E memory is full. It will be remembered in the example of FIG. 21, that it is under those conditions that the second segment of the memory E is brought into the system. It will be understood that the flip-flop X_{52} will be set when the terms $N_4 \cdot D_4 X_{41}$ and $N_6 \cdot D_{6a} \cdot X_{61} \cdot \overline{X_{62}}$ are true; and that the flip-flop X_{63} will be set when the terms $N_4 \cdot D_4 X_{41}$ and $N_6 \cdot D_6 X_{62} \cdot X_{61} \cdot \overline{X_{63}}$ are true.

For each succeeding pass, the capacity of the sorter should be reduced by a factor of $1/2j$ where $j=n-i-1$, where n is the number of memories, i is the largest memory from which an exchange to the least memory occurred during the preceding pass.

At the end of each pass, for example, all the flip-flops X_{ab} are reset. Then, the sorting system is set up with the $1/2j$ reduced capacity for the succeeding pass, by setting each of the X_{ab} flip-flops where $a-b$ is equal or greater than $n-i+3$, assuming that one of the P_i flip-flops of FIG. 17 is reset. For example, if the flip-flop P_5 is reset in a system having seven memories, for example, the flip-flop X_{72} will be set since 7−2 equal 7−5+3. However, the flip-flop X_{62} will not be set since 6−2 is less than 7−5+3. The control of the flip-flops X_{ab} for the succeeding passes in the particular example, under consideration is represented by the logic circuit of FIG. 23.

For example, if at the end of the first pass there was an exchange between a record from the memory E(L_5) and the least memory A(L_1), the flip-flop P_5 of FIG. 17 is reset, assuming that the memory E is the largest in the particular example under consideration. The term $\overline{P_5}$ will set the flip-flop X_{41} in FIG. 23, and it will also set the flip-flop X_{52}. Moreover, the term $\overline{P_5}$ will set the flip-flop X_{51} by virtue of an "OR" gate 800. This is in accordance with the X_{ab} formula of $a-b$ greater or equal to $n-i+3$ given above. Therefore, in the second pass, the flip-flop X_{41} is set to provide a full line memory D, but only the flip-flops X_{51} and X_{52} in memory E(L_5) are set so that only part of the memory E_s is used for the succeeding pass.

Now, should the maximum line at which an exchange is made with the least line be the memory D(L_4), then, as shown in the logic diagram of FIG. 23, only the flip-flop X_{51} is set. This means that in the succeeding pass, only one segment of the memory D(L_4) is used, and two segments of the memory E(L_5) are used.

Therefore, by the appropriate control of the flip-flops X_{ab}, the capacity of the memory is reduced for each succeeding pass to adjust the capacity required to handle the records yet remaining unsorted, so that the time required for each succeeding pass is proportionally reduced. The X_{ab} flip-flops are reset when the memories become empty, by the respective terms N_i.

As mentioned previously, in order to adapt the system for use in conjunction with a variety of sizes and types of memories, the shorter delay line memories may be replaced by shift registers, and such an embodiment is shown in the fragmentary block diagram of FIG. 24. In FIG. 24, the memories A, B and C have been replaced by shift registers designated A', B' and C', whereas the longer length memories D and E of the embodiment of FIGS. 2A and 2B, for example, are provided by a standard magnetic drum or disc type of memory designated 10'. It will be appreciated that other types of memories may be used. As mentioned above, the shift registers A', B' and C' may be incorporated into a unit which also incorporates the sorter logic, such as the logic circuits A_1-A_7, and the other circuitry shown, for example, in FIGS. 2A and 2B. The shift registers need by shifted only when data is fed into the system, and Z blanks need not be used.

The resulting unit may have a plurality of input terminals, such as designated L_1, L_3, L_6 and L_7. These input terminals may be used in conjunction with different memories 10', depending upon the size of the particular installation for which the standard sorter unit is to be used. When used in conjunction with the illustrated memory 10', for example, it is assumed that the input terminals L_6 and L_7 are not used, and B** blanks are applied to both those input terminals.

As mentioned above, if the shorter memory lengths, as represented by the shift registers A', B' and C', are so selected that the longer memory lengths within the memory 10' may consist of an integral number of tracks on the drum or disc, then a standard magnetic memory equipment 10' may be used in conjunction with the sorter network.

The unit of FIG. 24 may be constructed to accommodate a large number of memory outputs, and there is no need for providing a sorter network S big enough to handle all the inputs. As shown in FIG. 25, for example, some of the lines may bypass the sorter S, and be connected directly to the S_i sorter networks and from there to the S'_i sorter networks, as shown in the fragmentary representation of FIG. 25. When a smaller sorter network S is used, certain economies may be realized. The sorting process is not affected, except that more passes are required with the smaller sorter networks S, than with the larger sorter networks.

By means of the embodiments shown in FIGS. 24 and 25, a drum memory of any convenient size, for example, may be used, so long as its pulse rate and number of bits per track are reasonably compatible with the selected shift rate of the shift registers A', B', C', and record length. Also, standard units may be provided with smaller or larger sorter networks S, as dictated by the economies of any particular range of situations.

As mentioned above, an appropriate logic unit designated a "change box" (CB) may be included in the sorter system. The change box is capable of changing the identification fields of selected records in a predetermined manner. This permits, for example, any desired group of records in the file to be shifted to the front or rear of the file, for examination of that particular group. As shown in the fragmentary representation of FIG. 26, the change box (CB) may be interposed between the sorter S_0 and the logic circuit C_1.

As mentioned, the change box (CB) permits the alteration of the identifying field of selected records to permit a prescribed placement of the selected records in the file. That is, instead of each identification field being considered as a single multibit binary number, it may be made up of a number of multibit characters. These characters may designate decimal numbers, in which 4-bit characters would be used; or they may designate letters of the alphabet in which 6-bit characters would be used.

The change box (CB) may also be used to transform a code, character by character, in some or all of the records as they are transmitted to the sorting system. When so used, the code changes may occur in a part of the record which is not used in the sorter in performing the sorting operation, and such code changes will not affect the manner in which the records will be handled by the sorter.

When the change box (CB) is used to control the identifying fields of the records, however, the actual sort of the records by the sorter is affected. For example, the records involved in a particular sorting operation may include multicharacter identifying fields which identify a group of subscribers by name. The sorting system may then be set up normally to arrange the records in alphabetical order. However, by an appropriate control of the change box (CB), any particular subscriber, or group, may be placed in the front or rear of the file.

When the change box (CB) is used, the sorting system must be constructed to assure that all the records in the file will pass through the change box, and that each record passes through just once. When the change box is used in the manner

described in the preceding paragraph, this may be accomplished by altering the input mode of the sorting system so that the distinctions appended to the records by the C_1 logic circuit, as described above, are not removed in the A_i logic circuits. The change box (CB) may then be controlled selectively to alter the desired records, only if they are not distinguished.

When the change box (CB) is used to control the identifying fields of the selected records so as to determine the manner in which such records will be handled by the sorter system, it may be connected into the sorter system of FIGS. **2A** and **2B** in the manner shown in the fragmentary diagram of FIG. **26**. That is it may be interposed between the upper output terminal of the sorter S_0 and the input terminal of the logic circuit C_1. Appropriate means (not shown) may be provided in the system for reconverting the data back to its original form for output.

The logic circuitry of FIG. **27**, namely the "AND" gates **800** and **802**, and the "OR" gates **801** and **804**, assure that only undistinguished records (t_2=0) will be handled by the change box (CB) and that all distinguished records (T_2=1) are bypassed around the box CB to the logic circuit C_1. The "AND" gate **800** also has the terms I.M. and $\overline{D}$ applied to it so that, for convenience, the box CB is made effective only during the input mode and only when no new data is being transmitted to the sorter. The "OR" gate **801** and the "AND" gate **802** assure that under all other conditions the S_0 output will be passed directly to the logic circuit C_1 through the "OR" gate **804**.

For purposes of explanation, and in order to simplify the description, it will be assumed the identifying field of each record may include a series of 2-bit characters, such as a character A (0, 0); B (0, 1); C (1, 0); or D (1, 1). The change box (CB) provides that any one of these characters may be selected to be changed into any one of the other characters. A separate circuit is provided in the change box (CB) for each character, and one such circuit is shown in FIGS. **28** and **29**, the circuits for the other characters may be the same. It will also be understood that the circuits may be modified to handle more or less characters of any appropriate numbers of bits. It should also be pointed out at this time that the control on the change box (CB) may be provided by any suitable manual keyboard, for example, and need not necessarily be a computer control.

The change operation in the change box (CB) is under the control of four flip-flops (A', B', C' and D') in correspondence with each different character. Thus, when a character A is detected, that character is directed through a first circuit, which causes it to be changed to B', C' or D' or to remain unchanged, depending upon which of the four flip-flops A, B, C or D in that circuit is set. Likewise, when a character B is detected, that character is directed through a second circuit, which causes it to be changed to A', C' or D', or to remain unchanged, depending upon which of the four flip-flops A', B', C' or D' associated with that circuit is set; and so on.

The flip-flops A', B', C' and D' in each of the different circuits referred to in the preceding paragraph may be set under the control of different setting circuits **806**, such as the setting circuit of FIG. **28**, and each under the control of a common shift register S_1, S_2, S_3, S_4 and a separate key or switch A, B, C and D in a keyboard **808** for the different circuits.

In the case of the setting circuit of FIG. **28**, the switch A is converted to a series of "AND" gates **810**, **812**, **814**, **816**; as are the different output terminals of the shift register. These gates are connected to respective flip-flops A', B', C', D' to set the flip-flops as desired. It will be understood that the switches B, C and D will be connected, through like gates to their flip-flops A', B', C' and D'.

The shift register is common to all the aforesaid circuits. When it is set to its first position, by any appropriate control, the keys A, B, C or D are closed, depending upon which of the original characters is to be changed to an "A," for example. The shift register is then set to its second position, and the keys are closed for all changes in the original characters to a "B," and so on. It will be appreciated that similar circuits may be used to return the altered records to their original format, at the end of the operation.

In the manner described, for example, all records identified by the characters, D, A, B, B; may be selected from the file, merely by setting the circuits, such as the circuit of FIG. **28**, to change all the characters D and B to A, and the leave the A characters unaltered. Then, the "DABB" records will become "AAAA" records and will appear together in front of the file at the sorter output.

As mentioned above, the detection circuit for the change box (CB) is shown in FIG. **29**. The detection circuit includes a pair of flip-flops X and Y. The input to the change box is applied to the flip-flops X, directly to its set input terminal and through an inverter I to its reset input terminal; and through "AND" gates **850** and **852** to the flip-flop Y. The flip-flops are connected as a shift register and they store the first and second bits of each character fed through the change box (CB), long enough to detect whether the character is A, B, C or D. The pulses "CL" are clock pulses. The flip-flops enable the gates **852**, **856**, **858**, **860** in correspondence with the particular character detected by the flip-flops X and Y.

Each of the gates **854**, **856**, **858** and **860** has logic associated with it, such as the logic circuit shown connected to the output of the gate **854** (A). All the logic is subsequently "OR"ed to the output terminal **880** of the change box. For example, the A output from the gate **854** is applied to a series of gates **862**, **864**, **866** and **868**. These latter "And" gates are selectively enabled by the setting circuit of FIG. **28**, and they determine which transformation shall be made to each "A" character.

Outputs from the "AND" gates **762**, **764**, **766** and **768** are applied to further "AND" gates **770**, **772**, **774** and **776**, and bit timing pulses CL_1 and CL_2 are also applied to the "AND" gates, so that the transformed bits in the characters may be properly set as A(0, 0); B(0, 1); C(1, 0); or D(1, 1). The transformed characters are then passed through an "OR" gate **778** to the output terminal **780**.

In a manner similar to that described above, one or both of the one-record memories may be composed of separate addressable segments so that the format of the records may be changed. That is, the control field of each record may be changed for another portion of the record by switching such segments, thereby permitting different file orders to be achieved. A one-record memory may be controlled, for example, so that the control field of a record enters one segment and the remainder of the record enters the other segment, and so that the remainder of the record leaves the other segment followed by the control field.

The system of the invention may be controlled by the circuitry of FIGS. **30A** and **30B** to handle records which are predetermined multiples of the records (M=1) considered previously herein. For example, by appropriate control of the circuitry shown in FIG. **30**, double length records (M=2) may be sorted, as well as records which are four (M=4) or eight (M=8 times the length of the original records. The diagram of FIG. **31** illustrates the paths taken by the various multiple length records in the presence of the various controls in the modified system.

In the modified system of FIGS. **30A** and **30B**, the illustrated logic circuitry is interposed between the output terminals L_1—L_5 of the memories described above in conjunction with FIGS. **2A** and **2B**, and the various logic circuits A_1—A_5. The circuit of FIG. **30B**, on the other hand, is intended to be inserted between the logic circuits F_1—F_5 of FIGS. **2A** and **2B**, and the input terminals L'_1—L'_5 of the aforesaid memories. The sorting circuitry of FIGS. **2A** and **2B** is represented by a block **900** in FIG. **30A**.

The logic diagram of FIG. **30A** includes a series of "AND" gates and the "AND" gates are **902**, **904**, **906**, **908**, **910**, **912**, **914**, **916**, **918**, **920**, **922**, **924**, **926**, **928**, **930**, **932**; and associated "OR" gates **940**, **942**, **944** and **946** interconnected in the illustrated manner between the output terminals L_1—L_5 of

the aforesaid memories and the logic circuits A_2-A_5. The terms $M=1$, $M=2$, $M=4$, and $M=8$ are commands designating the length of the records to be handled by the system. For example, when the records have the length considered in the previous embodiments, the command $M=1$ enables the "AND" gates **902**, as well as the "AND" gates **910**, **916** and **922**; on the other hand, when the record length is double the length considered in the previous embodiment, the command $M=2$ enables the "AND" gates **904**, as well as the "AND" gates **914**, **918**, and **924**. Likewise, when the records to be handled by the system are four times the length of the previous records, the command $M=4$ enables the "AND" gates **906**, as well as the "AND" gates **912**, **918** and **926**. Finally, when the records to be handled are eight times the length of the previous records, the term $M=8$ enables the "AND" gate **908**, as well as the "AND" gates **912**, **918**, **928** and **930**. The "AND" gate **920** is enabled for all multiple length records, as determined by the term $\overline{M=1}$.

The logic circuitry of FIG. **30A**, as well as the logic circuits F_1-F_5 are connected to a further series of "AND" gates **950**, **952**, **954**, **956**, **958**, **960**, **962**, **964**, **966**, **968**, **970**, **972**, **974** and **976** in FIG. **30B**. The latter "AND" gates are connected to a series of "OR" gates designated G1, G2, G3, G4 and G5. The outputs of the latter "OR" gates introduced respectively to the input terminals $L'_1-L'_5$ of the aforesaid memories.

The circuitry of FIGS. **30A** and **30B** is controlled so that the records follow the paths shown schematically in FIG. **31**. For example, when the normal length records are being handled by the system, the command $M=1$ causes the records to follow the paths shown schematically in FIG. **31A**, which is the path shown in FIGS. **2A** and **2B** and described previously herein. However, when a double length record is to be sorted by the system, for example, the command $M32$ 2 causes the two one-word memories L_1 and L_2 effectively to be connected in series, so that the memory sequence becomes 2, 2, 4 and 8. The paths followed by the outputs of the various memories is then shown in FIG. **31B**. Likewise, when the record length is four times the normal record length, the three lower memories are effectively connected in series, as shown in FIG. **31C**, so that the memory sequence becomes 4, 4, 8, and the record paths are as shown in FIG. **31C**.

Finally, when the records have eight times the normal length, all the memories except the longest are effectively in series, so that the memory sequence becomes 8, 8. Then, the records follow the illustrated paths in FIG. **30D**. It should be noted that at least two memories must be used in the system, or no sorting will be accomplished. It will also be understood, of course, that the system of the invention is not limited to any particular memory length, and that other similar sequences may be used and controlled in the manner shown in FIGS. **30A**, **30B** and in FIG. **31**.

The circuitry of FIGS. **30A** and **30B** provides that if data does not pass to any of the sorting circuits **A1—A3**, the **B**** blanks are passed to the circuit.

The invention provides, therefore, an improved sorter by which a file of records may be sorted in subsequent passes through the system. The sorter also has the capability, as described, of handling different length records, and of changing the identifying fields of selected records so that such records may be shifted into a desired position in the file.

The complexity of the system of the invention may be reduced, and the number of passes required for a complete sort may also be significantly reduced, by replacing the shorter memory elements, such as the two-record memory element L3, and the two one-record memory elements L2 and L1 in FIG. **2A**, by one-record register and sorter combinations formed from the basic module shown in FIG. **32**.

In FIG. **32**, the block R_f represents a one-record shift register, the circuit represented by the block S_f is a sorter, such as the sorters described above in conjunction with FIGS. **2A** and **2B**, whereas the circuits A_f and F_f correspond to the A and F circuits of FIGS. **2A** and **2B**. The module shown in FIG. **32** is a circulating type of circuit, in which the output of the network F_f is fed back to the input of the one-record register R_f.

For example, three such registers R_f may be included in the network to replace the three memory elements L3, L2 and L1 of FIG. **2A**. Such one-record registers are represented, for example, in FIG. **33** as R_0, R_1 and R_2. The registers and their associated networks A_0, A_1, A_2 and F_0, F_1 and F_2, as well as their sorters S_0, S_1, S_2 and S_3 are interconnected in the manner shown, for example, in FIG. **33**. The system shown in FIG. **33** has the property that although the data in it may not be ordered, the records may be withdrawn from it at any record time without delay.

For example, when all the records to be sorted have been placed in the system of FIG. **33**, they may be removed serially by record, so that at each record time, the record removed is the first of the sorted file composed of the records within the store. The output operation may then be delayed or interrupted for any integral number of record times and then resumed. The type of sorting system shown in FIG. **33** is useful for very rapid sorting of relatively small numbers of records, or for use in conjunction with other types of sorting systems, such as shown in FIGS. **2A** and **2B** for sorting larger numbers of records. The sorter system of FIG. **33** is also useful for certain data retrieval computations.

The system of FIG. **33** may be controlled by the input made flip-flop I.M. and by the output mode flip-flop O.M. of FIG. **12**. The flip-flop I.M. is set when the records are fed into the system, and the flip-flop O.M. is set during periods when the records are to be fed out of the system. When records are to be fed into the system of FIG. **33**, the flip-flop I.M. is set under an appropriate external control, and the flip-flop O.M. is reset. When the flip-flop I.M. is set and the flip-flop O.M. is reset, the circuits A_f transmit Z blanks, thus effectively erasing the contents of the corresponding registers R_f, until data is fed into R_f. When a record is to be placed in the system, it is transmitted to the sorter S_0 by way of the input line I_1, where it will normally be exchanged for the Z blanks, and from the sorter S_0, via the sorters S_1, S_2 . . . to the block F_f. When a data record enters F_f, the flip-flop D of FIG. **14** is set, indicating the entry of a data record into a register R_f. When a flip-flop D_f is set, the circuit A_f transmits its input to its output. When each register R_f is filled, data may be transmitted via O_1, with consequent skimming of the file. If removal of the contents of the registers R_f is desired, such removal may be accomplished by transmitting **B**** blanks on the input line I_2, with consequent displacement of the data records along the line O_1, in the manner described above.

Without the provision of the flip-flop O.M., no data could be removed from the system of FIG. **33** until each data register has been filled, because the circuits A_f would transmit Z blanks if they contained no data. Because of this, the flip-flop O.M. is provided to cause the circuits A_f to transmit B blanks during the output mode.

The system shown in FIG. **33** operates in much the same named as the sorting system disclosed and claimed in U.S. Pat. No. 3,399,383, which issued Aug. 27, 1968, to the present inventor. The system of FIG. **33**, as used in conjunction with the system described in the patent, or as shown in FIGS. **2A** and **2B** hereof, is shown in FIG. **34**. As mentioned above, in the assembly of FIG. **34**, the one-record registers R_1, R_2, R_3 and associated networks replace the two-record memory element and the two one-record memory elements of the previous systems. It will be appreciated that the system shown in FIG. **34** is less complex than the previous system, in that the main sorter S is required to handle a smaller number of records, and may, in the particular embodiment, be a simple two-input two-output sorter, such as the other sorters shown in the system. Also, many of the components of FIG. **2B** are eliminated in the latter embodiment.

When the number of one-record registers R_f is equal to the length of the shortest memory element L_f, such as shown in FIG. **34**, no change in the input or output modes of operation from that of the aforesaid patent need be made except that in the output mode, distinguished data records are not stored in R_f, and the circuits C_f ($i<N$) are removed except for C1 which appends a * to data records in the sort mode.

In the sort mode, the setting of the Sort Mode flip-flop S.M. causes the circuits A_i to transmit their input data to their respective outputs. Thus the detection of the End of Sort Mode may be accomplished by noting the presence of a distinguished data record in register R_1. It should be noted, however, that the Sort Mode flip-flop S.M. can only be set if at least one data record was transmitted to the upper terminal of the sorter S'_2, for if data is stored only in the one-record registers R_i, no sorting mode is required.

The circuits A_i may be equipped to alter the formats of the data record outputs of the one-record registers R_i, and then a search may be made for those records with a specified property. This is precisely the function normally performed by an "associative memory". Each record includes a control number, and the required alteration of the control numbers may be accomplished by an arithmetic computation. That is, by proper selection of the inquiry record and also proper selection of the code for representing the data within the system.

The inquiry record may be treated as a representation of a number and simply added to the inputs of the A_i circuits and similarly subtracted from the outputs of the F_i circuits. After such an addition, the record with the smallest control number would enter the upper input to the sorter S_0, and would be available for removal from the system, in altered form.

If nondestructive readout is desired, the addition of a single gate permits reading of the data without removal thereof from the system. For example, and as shown in FIG. **35**, the "AND" gates **981** and **982**, together with an "OR" gate **980**, permit a nondestructive readout. The gate **980** is connected to the lower input terminal of the sorter S_0, one of the input terminals of the gate **981** is connected to the upper input terminal of the sorter S_0, and one of the input terminals of the gate **982** is connected to the blanks B** line. This gate circuit permits a nondestructive readout when the term N.D.R. is used to enable the gate **981**, for the upper input to the sorter S_0 is used to displace the sorted signal to the output line O_1 without loss since the displaced signal appears at the upper output terminal of the sorter S_0. Then, for the destructive output mode, the term D.R. is used to enable the "AND" gate **982**, so that blanks B** used to displace the output, as in the previous embodiment.

If more than one record should be present with a desired control number, the circuits C_i may be used effectively to circulate the data within the system until all the desired contents have been read.

It is evident that the records flowing in the system of FIGS. **32** and **33** must pass through numbers of gates determined by the ordering of the records within the store. Thus, the amounts of delay in the paths of the records must be considered in constructing a system embodying the module of FIG. **32**. However, a feature of the module, as combined into the system shown in FIG. **33**, is that the delays encountered in the system may be made part of the shift registers for the most part, and only the delays encountered between gating elements require special delay networks, so that the system is essentially modular.

The end of the sorting process by the system in FIG. **34**, for example, may be detected by noting the exchanges made in the sorting circuits S_i. If no exchange is made for a complete pass, the data will be available in sorted order. Such a means of checking when the sort is accomplished is normally more efficient than, for example, counting the number of required passes. However, it might be appropriate to provide means for counting the number of passes, so that the system may be stopped after a predetermined number of passes, known to be sufficient to achieve a complete sort, has been made, in order to prevent unlimited operation of the machine due to a malfunction in the detecting circuit.

It will be appreciated, therefore, that with respect to the sorting system of FIG. **33**, and the similar portion incorporated into the system of FIG. **34**, the essential characteristic is the module shown in FIG. **32**. Also, the sorting system, as described above, may be used in various ways, depending on the computation to be performed. Although, for purposes of the description, the registers R_i have been stated to be one-record registers, it may well be that for some systems, the registers may contain many records.

What I claim is:

1. A system for sorting multibit binary records into a desired sequence in a plurality of passes, each of the aforesaid records containing an identifying field and a data field, said sorting system including: input terminal means to which the aforesaid binary records are successively applied in an unsorted sequence; output terminal means at which the aforesaid binary records are successively produced in a generally sorted sequence; a plurality of storage means; sorting circuitry and logic circuitry intercoupling said input terminal means, said storage means and said output terminal means; said sorting circuitry including a sorting network having a plurality of input terminals connected to the outputs of said storage means and having output terminals, said sorting network responsive to said identifying fields of the multibit binary records applied to its input terminals to cause the records to appear in a particular sequence across its output terminals, said sorting network including circuit means indicating whenever a record produced at the output of one said storage means is exchanged with the record produced at the output of another of said storage means; and control circuitry included in said logic circuitry and coupled to said circuit means in said sorting network and responsive to indications therefrom for controlling the total capacity of said storage means for each succeeding pass of the records through the system.

2. The sorting system defined in claim **1**, in which said storage means comprise a plurality of separate memories having predetermined lengths in a particular progression, and which includes further logic circuitry for controlling the switching of said memories in and out of the system; and in which said control circuitry is coupled to said further logic circuitry to control the aforesaid switching of said memories into and out of the system.

3. The sorting system defined in claim **1**, in which certain of said storage means are composed of a plurality of individual segments, and which include further logic circuitry for controlling the switching of the said segments into and out of the system; and in which said control circuitry is coupled to said further logic circuitry to control the aforesaid switching of said segments into and out of the system.

4. The system defined in claim **1**, in which two of said storage means each has a capacity to store one of said records only, and the others of said storage means have respective capacities to store different numbers of said records in a predetermined progression of 2^1, 2^2, 2^3, $2^4 \ldots 2^n$.

5. The system defined in claim **1**, in which said storage means comprises a plurality of memory elements, certain of said memory elements being shift registers.

6. The system defined in claim **5**, in which other of said memory elements comprises a separate memory unit.

7. The system defined in claim **6**, in which said last named separate memory unit comprises a movable magnetic memory member having individual tracks thereon constituting said other memory elements.

8. The system of claim **1** in which said storage means comprises a plurality of memory elements, a first group of said memory elements being included in the aforesaid logic circuitry and the other of said memory elements being incorporated into a separate memory unit.

9. The combination defined in claim **8**, in which said first group of memory elements are in the form of shift registers.

10. The combination defined in claim **8**, in which said separate memory unit includes a movable memory member having individual tracks thereon constituting said other memory elements.

11. The system defined in claim **1** and which includes further logic circuitry included in the aforesaid logic circuitry and which is responsive to selected ones of such identifying fields for selectively changing the bit composition thereof.

12. The system defined in claim 1, and which includes control circuit means in said logic circuitry and responsive to selected ones of such identifying fields for selectively changing the bit composition thereof.

13. The system defined in claim 1, in which said multibit binary records are of a selected length, or selected multiples thereof, and in which said system includes logic circuitry responsive to applied command signals for controlling said storage means to respond to such binary records.

14. The system defined in claim 1 in which said plurality of storage means comprises separate memories through which the records pass and which have different capacities for storing said records; and which includes control circuitry in said logic circuitry for effectively interconnecting selected ones of said separate memories in series in response to applied command signals designating the introduction of multiple length records into the system.