

lines the operating steps for the machine procedure. Many of the rules for writing the expressions are merely a review of the fundamentals taught to a beginning student of algebra. These rules must be strictly adhered to so that both the programmer and the machine interpret each situation in the same manner. The talk accompanied by the digest proves to be sufficient information for a beginner. Naturally, there are individual questions at a later time. These questions usually concern the inclusion of additional features to improve the system. All of these new ideas are welcome and whenever possible they are included immediately.

It is worth mentioning that most improvements and additions are readily included. The BACAIC system was originally planned with that desired flexibility in mind. It is fairly easy to include an additional mnemonic symbol but, if the symbol is to refer to a library subprogram for its operation, the library subprogram must also be available.

The specific information for each problem; i.e., the expressions and the input data, can be relayed by means of the telephone from another department to the computing facility. This service is possible because of the standard procedure of the BACAIC system after the expressions

are formulated. The availability of telephone service tends to decrease the elapsed time for solving a problem, and also tends to increase the correctness of the written expressions. Individual pride and reputation play an important part in reducing careless errors. Usually, the "debugging" of the expressions is possible in a few minutes prior to any machine operation. This fact is of tremendous advantage economically since without an automatic system considerable machine time is spent on check-out for each problem.

Conclusions

A natural computation language eliminates the machine coding details currently responsible for the expenditure of large amounts of man and machine time. This language can include complete machine operating directions as well as the mathematical problem statements. The time now spent in digital computer problem preparation can be reduced by as much as 90 per cent through the use of machine self-coding systems. A fundamental computation language makes evolutionary machine changes possible without extensive personnel retraining. The man-hours required to construct an autocoding system are no more than those formerly spent on a subprogram library. The basic interpretive principles for an algebraic computing system are applicable to most present-day stored program digital computers. These principles can be incorporated in the hardware of future machines.

Lincoln Laboratory Utility Program System

H. D. BENNINGTON C. H. GAUDETTE

THIS paper discusses a utility program system to assist the coding, check-out, maintenance, and documentation of large-

H. D. BENNINGTON and C. H. GAUDETTE are with the Lincoln Laboratory of the Massachusetts Institute of Technology, Lexington, Mass.

scale control programs. A typical program contains 50,000 instructions, 1,000,000 bits of data storage, and is prepared by a staff of 20 to 40 programmers, many relatively inexperienced. The utility system requires 25,000 registers.

An Automatic Supervisor for the IBM 702

BRUSE MONCREIFF

VERY little experience has been accumulated in the operation of a large commercial data-processing center. However, reflection on the subject has

BRUSE MONCREIFF is with The Rand Corporation, Santa Monica, Calif.

led to the conclusion that, in the large-scale operation of such a system, there will be a different emphasis from the one usually present in the operation of a large-scale computing installation. The general administrative problem in both cases is, of course, to keep both staff and equip-

ment operating efficiently. In the latter case, however, the emphasis is on new problem preparation, while in the case of the business application the emphasis must be on the efficient day-after-day operation of the same routines. The automatic supervisory routine described here is an attempt to solve those operating and programming problems peculiar to this "routine-dominated" situation.

The excuse for solving these problems with a machine program, rather than by instructions to the operator, is twofold:

1. The human operator cannot compete in speed with the machine in making routine decisions and in controlling the processing operations.

2. The human operator is more likely to make mistakes in carrying out routine instructions.

The purpose of a supervisory routine is, therefore, to keep the machine running efficiently in spite of the slowness and fallibility of the human operator.

The various aspects of the proposed supervisory routine will be approached by looking at these problems of both operator and programmer which the routine helps solve. A summary description of the way the routine works will follow.

Operating Problems

On a machine such as the 702, the major operating problem is the efficient handling of tape reels. Every few minutes, during a long run, either an input tape will have to be selected and mounted, or an output tape will have to be labeled and stored away. The malfunctioning of tape units will probably result in the assignment of units to the various input-output functions being changed from day to day. Trying to keep the machine running under these conditions, without a good system of operation including adequate checks, could be a difficult job, with a high probability of error. The component parts of this tape-handling problem will be examined one at a time.

1. Going from one job to another, aside from the internal problem of replacing the old program with a new one, involves a new input-output arrangement of the tape units. If this arrangement were the same every time a particular job is run, a table of assignments for each job could be preprinted and used day after day. Because of the malfunctioning problem mentioned, this is not feasible. Also, whenever there are sufficient tape units operable, the procedure of alternating tape units should be used. While one tape of a certain type is running, the next tape of that same type can be mounted on another tape unit. The problem of getting the machine to deal first with one tape unit and then the other is a programming problem to be dealt with in the following. However, the day-to-day changes in the input-output setup for each job, to take advantage of all the operable tape units, is a problem for the operating staff. The solution proposed here is to get the supervisory routine to calculate the optimum input-output arrangement before each job, having been given a list of the presently operable tape units. The routine should assign alternate tape units as far as possible, starting with those tape types which have the greatest number of tapes per run. Also, assign-

ments of alternate pairs should be made to physically adjacent tape units, as far as this is possible. The machine will print out a table of assignments which the operator will use as a guide in mounting tapes.

2. Once the tapes have been mounted, the supervisory routine will check to see that this has been done properly. Each input tape will be checked for proper type, proper cycle number (e.g., that it was the output of yesterday's run), and proper sequence number within type and cycle. The first record of each tape in the system will be a label containing the information required for this check. The supervisory routine will consult a table of cycle increments, one for each input type by job, and will keep track of the sequence of tapes mounted during each job. The routine will also check each tape mounted for the purpose of receiving output to guarantee that the information on this tape is no longer needed. It will do this by comparing the cycle number of the tape with the current-run cycle number minus an increment (file protection period) which may be different for each tape type.

To summarize the foregoing paragraph: Each tape type will have associated with it both an input increment and an output increment; the former will guarantee the accuracy of input-tape mounting, and the latter will protect information against accidental erasure until it is no longer needed.

3. An identification label will be recorded by the supervisory routine on all output tapes as well as printed out on the typewriter for mounting on the reel can. This will provide positive visual identification for each can of tape in the system.

There is another operating problem, not connected with tape handling, which a supervisory routine can help solve. This is the problem of setting alteration switches before the running of a job. Two points are worth noting in this connection. First, the number of physical alteration switches provided on the 702 will occasionally be inadequate. So provision must be made for logical or programmed alteration switches. Secondly, there are what might be called the active and the passive types of switches. In the active type, the operator has a present need to manipulate a switch in order to alter the machine activity. The passive type of switch is one which he has been instructed to set in a certain way at a certain point in the processing cycle. While the operator is not likely to make mistakes with active switch settings, the probability of error increases in the case of

the passive switches. It is recommended that the physical switches be reserved for the active use, while a system of logical, or programmed switches, to be set by a punched card, be used for the passive, job-setup type. The supervisory routine will establish whether or not an alteration switch card is needed for each particular program, and if so, will wait until a card of the proper type is provided. This card will be stored in a standard drum section. This system relieves the operator of the responsibility of remembering to make the switch settings before each job. A file of prepunched cards can be maintained which will cover the majority of setups. Most, if not all, occasions of the use of program parameters can be handled by this control card.

As a further aid to the operator, especially for the purpose of retracing the steps of a process that went wrong, the supervisory routine will automatically keep a log. Entries will include job numbers, tape setups, switch settings, information entered manually or by card, and error stops. One solution to the problem of the multiple use of the typewriter is the creation of a carbon copy. The tape labels can be cut out of the original, with the carbon left intact as a log.

Programming Problems

The problems which a supervisory routine can help solve, in the programming area, are those of co-ordination among programmers and those of co-operation between programmer and operator. In the first place, the employment of a supervisory routine can prevent the duplication of effort which arises where each program must contain the same general control and housekeeping routines. Each programmer must, in each of his routines, provide for tape-unit assignment, tape-unit alternation, tape-mounting instructions and checking, and a labeling system. He must also provide for loading his routines and for alteration switches of the program variety. All of these matters can be taken care of once and for all by means of incorporation in a supervisory routine.

In addition to eliminating most of the duplication of effort among programmers, the supervisory routine can be an unobjectionable means of enforcing the necessary standard practices. This is especially important in the matter of tape identification and tape-mounting checks. Unnecessary confusion would arise if every programmer were left to design his own system of output-tape identification. Then there would have to be designed a

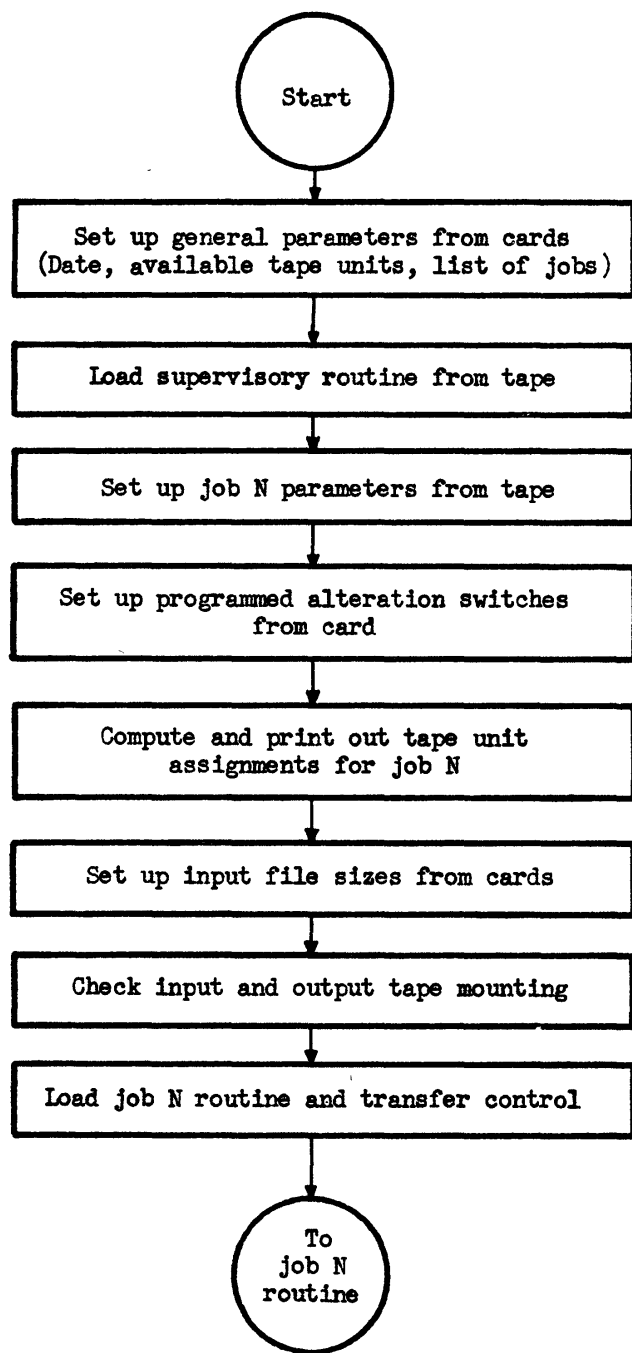


Fig. 1. Supervisory routine: initial operations

Operation of the Supervisory Routine

The supervisory routine will now be approached from the viewpoint of the machine operator (Fig. 1). The processing will be started each day by entering, via punched card, the day's date, cycle number, list of available tape units, a (standard) list of jobs with a starting point, and select and read orders for bringing in the first section of the supervisory routine from tape. This information, all but the select and read orders and the list of jobs, will be typed out as the day's first log entry. The list of jobs will control the processing sequence, and may be altered manually at any time. The programs for the various jobs will be automatically called up from tape by reference to this table. When the initial setup has been completed, the supervisory routine will be loaded from tape, and will search for the first job. The first section of each specific program contains the necessary program parameters, principally tape-unit requirements, to be used by the supervisory routine in accomplishing the setup. The supervisor will check first to see if there is need for a new setting of the programmed alteration switches. If so, the card will be read and the settings made. The complete tape-unit setup for this particular run will be computed and printed out. The machine will stop, waiting for the operator to signify, by pushing the start button, that the tapes have been mounted.

As soon as the tapes are mounted, each input tape and then each output tape will be checked as described. Any incorrect tape mounting will result in a printout describing the trouble, followed by an error stop. When the error is corrected, a recheck by the supervisor will be made of the whole setup, starting back with the first input tape. As the output tapes are checked, the identification record is written on the tape. This record is also printed out as a label when the tape is filled.

Upon satisfactory completion of the tape-mounting check, the supervisor will load the job program. After the specific program is loaded and arranged properly in memory and drum, a transfer of control to its starting point will take place. A transfer of control back to the supervisory routine, a small fragment of which is left stored in memory, will take place under any of the following conditions (Fig. 2):

1. An output tape has been filled. The output label will be printed, including any "closing" information left by the specific program in a standard location. If the

different input check for each type of input for every job.

The elimination of confusion is also the primary problem in the relations between the programmer and the operator. In systems where the operator has to run many different programs each day, he has many points of contact, and must understand and deal with many ways of doing things. In a sense, the supervisory routine provides a single point of contact between the operator and the programs. Once the characteristics of this supervisory routine are learned by the opera-

tor, his job is made easier. Changes in the specific job routines, once "debugged," are of little concern to him. Only changes in the supervisory routine directly affect the operator, and it is expected that most of these changes will grow directly out of operational needs. The previously discussed method of program alteration, the uniform method of tape identification, and the universal checking of tape mounting, will all but guarantee that the plans of the programming staff will be correctly carried out by the operating staff.

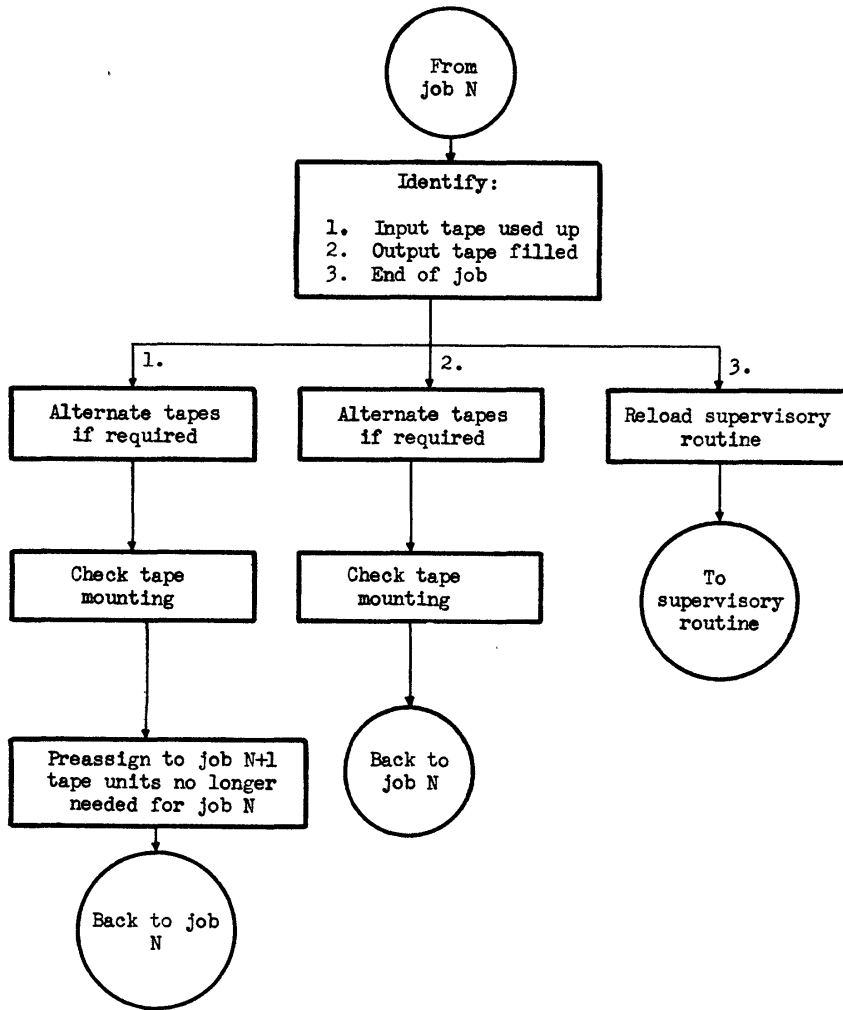


Fig. 2. Supervisory routine: between-tape operations

tape is one of those being alternated, controls will be set for the alternate tape unit. The routine will proceed immediately to the routine check of the tape mounted on the alternate unit. If this particular tape is not being alternated, the machine will pause until the new tape is mounted, before proceeding with the check, and the writing of the identification record. Control will then be transferred back to the specific routine.

2. An input tape has been used up. The same procedure for handling alternating tape units is followed, where necessary, before the new input tape is checked. The cycle number and serial number are checked and the serial number is stepped. Control is then returned to the specific program.

3. End of job. The supervisory routine is called up, and the setup for the next job is started. This consists of picking out the next job number and going through a setup procedure (Fig. 3) similar to that described.

Following the last legitimate job number of a cycle, there will be a pseudo job number which will call up a cycle closing

routine. This routine will step the cycle number and then test an alteration switch to see if the operator desires a shutdown or a new processing cycle.

Preassignment of Tape Units

Certain items appearing in the block diagrams have not been explained. These have to do with the preassignment of tape units. This subject will be dealt with separately here, since some background discussion is required.

The delays caused by tape mounting have been reduced as much as possible by having the supervisory routine alternate tape units whenever they are available. But this "hides" only the tape mounting time which is internal to a job. If the operator must wait until a job is completed before starting to mount tapes for the next job, this whole setup time will constitute a delay. To ameliorate this situation, the supervisory routine has

been designed to make preassignments whenever possible.

For output functions there seems to be no simple way of determining when tape units are no longer needed. On the other hand, the number of tapes in each input file is known in advance. This information, if available to the supervisory routine, can be used to trigger a preassignment of the no longer needed tape unit to the next job. In order to accomplish this preassignment, the routine needs to have available the job setup information for the next job, as well as the current one. The block diagram of the initial operations (Fig. 1) should be amended to include the setting up of job $N + 1$ parameters. The block diagram of the between-job operations (Fig. 3) should also be amended at the beginning of this routine to show that job $N + 1$ becomes the current job, job N , at this point. Then the first block again reads correctly, since it is the next job, job $N + 1$, parameters that are actually being set up at this time. These complications of the diagrams are not self-explanatory, hence they were not included.

The normal end-of-file system is not adequate, since in the case of alternating tape units, the first preassignment takes place when the next to the last tape has been used up. The method adopted for satisfying this need was to provide a card for each output file, punched automatically by the supervisory routine at the completion of a job. The card, in addition to identifying information, carries the number of tapes produced during this run. During a later cycle when the output file becomes input, the card is used to set up the controls needed for the preassignment function. Similar cards can be created manually for input files created outside the system.

When a job is completed, the remainder of the input-output setup for the next job is computed with the preassignments being conserved. The routine makes the best choice of physical location for alternate tape units, just as it did for the first job setup. But where preassignment is involved, the choice of the first of a pair of units has already been made. These second choices are made before any other assignments are made.

Programming the Supervisory Routine

Two general points of programming philosophy were adopted early in the detailed job of realizing the supervisory routine in a code for the 702. The first was that each requirement for informa-

tion storage would be satisfied by the lowest possible level in the 702 hierarchy of storage devices. The second was that time would be used wherever possible as a trade-off for storage. The result of this approach was: (a) a complication of the logic and (b) a requirement for only 700 characters of permanent high-speed storage.

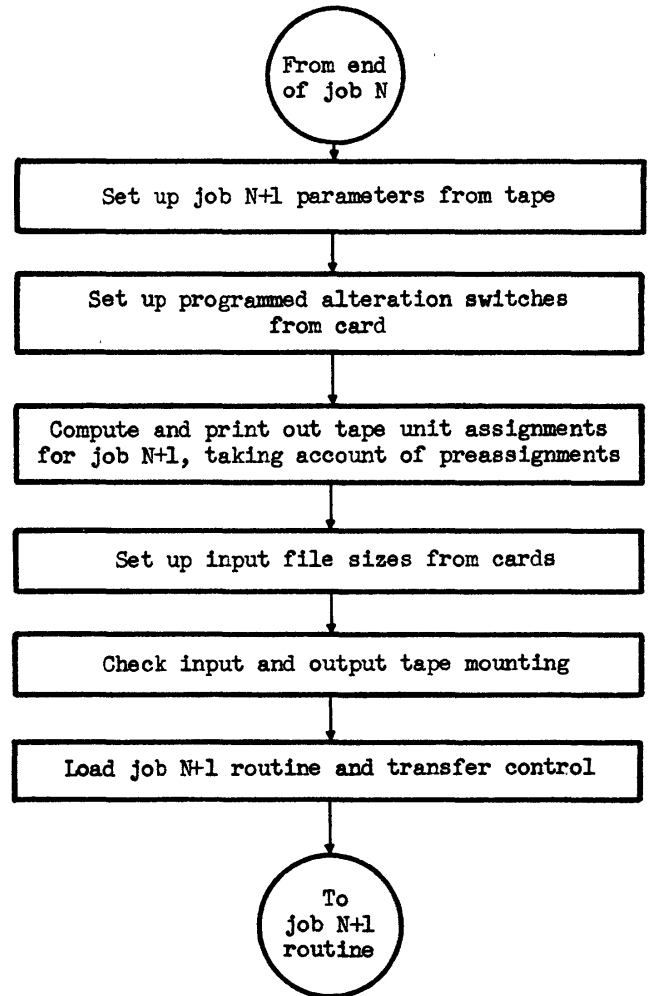
The supervisory routine is designed for storage on tape. The part which operates between jobs can fill the memory if need be (it does not, of course). This part is wiped out by the loading of the specific job program. The part of the routine which operates between tapes is stored on 15 200-character drum sections. For execution, these sections are loaded one at a time as needed into a common area in high-speed storage. Five additional drum sections are required for storing the tables of information used in the problem. When referred to, these are brought up into a second common area.

The part of the supervisory routine executed between jobs is reloaded from tape when needed. To save tape searching time, this program can be recorded again and again, between specific job programs. If this is done, and if the specific jobs are done in the order in which they appear on the program tape, there will be no tape searching required. However, full searching facilities have been built into the routine, for purposes of greater flexibility.

A single sheet (22-inch by 30-inch) flow chart was constructed initially, and then subjected to many changes. In a problem of this type, where logical complications are mostly those of relationships, a single sheet chart can be a great help. If carefully constructed, changes can become only a minor annoyance.

The routine was coded in IBM 702 symbolic, taking approximately 1,200 lines. This includes all entries required for the complete assembly of the problem for the 702.

Fig. 3. Supervisory routine: between-job operations



Conclusion

The activities of a computer operator can be classified as (1) those things which he does when the machine is working well, and (2) those things which he does when the machine is working poorly. The supervisory routine described is intended to be an aid to and substitute for the activities of only the former class. If considered as a research problem, this approach may be realistic, but it does not constitute a complete practical solution

which could be adopted as it stands. In fact, the aspect dealt with is most certainly the easiest part of the total problem of keeping the equipment occupied with useful work.

The purpose of the research was to gain a feeling for the complexity of a problem which as far as is known, has not heretofore been extensively investigated. The conclusion is that the construction of such a routine is not as complex as it originally seemed to be. The operating practicality is as yet not established.

Magnetic Recording Head Design

A. S. HOAGLAND
ASSOCIATE MEMBER AIEE

AN analysis of the process of magnetic recording of digital data is presented from which qualitative magnetic head design concepts are developed and their usefulness demonstrated, both in the evaluation of magnetic head structures and in design for high-density storage.

This work was begun in connection with the development of the International Business Machines Corporation (IBM) 305 magnetic disk storage file, where rather stringent requirements necessitated a program for improving the performance of the existing magnetic recording unit. The external constraints imposed on this work essentially limited the design parameters to those concerned with the magnetic head, excluding the gap dimension. The investigation of this subject led to the development of an analytical formulation of the process of magnetic recording which is directly applicable to data recording. The insight derived from the theory and its relative simplicity in application will be demonstrated through its successful employment in guiding head design development in connection with the IBM 305 magnetic disk project.

The paper will first briefly present a general discussion of magnetic data recording, stressing the form of the readback wave form and its relation to bit density. Then the basic theoretical development will be presented and the significance of these expressions to magnetic head design will be stressed. The remainder of the paper will present experimental results and their correlation with the theory, followed by a description of the selected magnetic head for the magnetic disk file and its performance.

Data Recording

Magnetic recording of data involves the storage and processing of binary information, utilizing two states which are capable of being differentiated. For convenience consider NRZ (nonreturn-to-zero) recording in which the storage surface is continuously saturated during writing, in one direction for a "1" and in the opposite sense for a "0." A particular input is then composed of a

succession of alternating step changes in writing current. Reading involves a derivative-type action as indicated schematically in the diagram below, illustrating the overall transfer process from input to output. The surface coordinate is x , and \bar{x} represents the variable indicating the position of the specified surface magnetization relative to the magnetic head.

$$i(t) \rightarrow M(x) \rightarrow \phi_h(\bar{x}) \rightarrow vNd\phi_h/d\bar{x} = e(\bar{x}) = e(vt)$$

where:

$M(x)$ = distribution of magnetization set up in the storage surface
 $\phi_h(\bar{x})$ = reading coil flux as a function of surface position
 e = open circuit readback voltage
 N = number of turns on the reading coil
 v = surface velocity; t = time

Hence $e(vt)$ is a pulse-like signal for a step change in $i(t)$ ¹ and the surface velocity appears as a scaling factor in both time and amplitude for the output signal. An output voltage signal is then associated with each change in the direction of saturation or reversal in writing current.

Consider the output signals for arbitrary input patterns consisting of sequences of alternating step changes in writing current. On readback the magnetic field existing, that due to the magnetization of the surface, is extremely weak and hence the magnetic head will behave very nearly as a linear element. Further, writing definition or the width of a saturation transition region is much less than the corresponding reading resolution, due to the non-linear surface saturation characteristic. Thus, generally, reading will limit density before the point is reached at which adjacent changes in surface state modify one another. These considerations imply that the principle of superposition may be applied to the overall input-output transfer process, using the characteristic step function output response. The width and wave form of this characteristic response are thus extremely important and such responses will be shown as functions of x , or distance, as only in this manner are they meaningful. As indicated, the velocity enters only as a scaling factor between distance and time and only spatial relations are significant in this recording process. It is evident

that bit density considerations will follow directly.

Consider a symmetrical voltage pulse response with the base width of λ . The maximum bit density permissible where no mutual interference is to be allowed, i.e., each saturation change is to be resolved independently, is then equal to $1/\lambda$. On the other hand if the requirement is only that no pattern modulation occurs, implying no signal peak variation, then the bit density has a limit of $2/\lambda$.

Now a typical response signal is shown in Fig. 1. The base-line character (or wave form) is quite important, as well as pulse widths at various levels above the base-line, in density considerations.

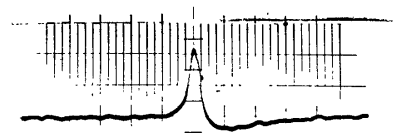


Fig. 1. Typical response signal

Scale: 3.5 mils per division

The "overshoots" may be broader than the main pulse. Thus no single criterion can be used in comparing responses.

A general objective for high-density design is to attempt to obtain a spike-like characteristic output signal, i.e., to eliminate base-line signal and maximize the rate of approach of the signal to the base-line. Considering undesired signal as noise, the overshoots tend to limit the signal-to-noise ratio at lower densities and first cause overlapping of signals as the bit density is increased. However, even with no base-line signal, as the bit density is increased and correspondingly a spacing reduction between adjacent sampling strobes errors will arise with an isolated pulse signal due to the gating of strobes on either side of the strobe which is actually associated with this output indication. This limitation is accentuated by pulse responses with gradual base-line approaches and can severely limit performance.

Theory

In addition to the principle of superposition the principle of reciprocity will be used, these forming the basic tools in this analysis. The applicability of the principle of reciprocity to readback follows as an immediate extension from the

A. S. HOAGLAND is with the University of California, Berkeley, Calif.