

METHODS & DESIGNS

The SSR system: An open format event recording system with computerized transcription

GORDON R. STEPHENSON, DANIEL P. B. SMITH, and THOMAS W. ROBERTS

The University of Wisconsin, Madison, Wisconsin 53706

The SSR system is an event recording system that encodes the incidence, duration, coincidence, and sequence of entries in real time onto magnetic tape for subsequent high-speed transcription by computer. The keyboard is light weight and battery powered for field as well as for laboratory applications. A conservative encoding scheme employs phase encoding and multiplexing circuitry to sample the set of 48 alphanumeric and other characters 20 times per second. The sampling rate provides a time base that is independent of tape speed. The encoding signal is recorded on an audio tape recorder at 1-7/8 ips (inches per second). A supplementary voice record can be made on a parallel track. At transcription, a small computer decodes the data tape at 15 ips; 1 h of data is transcribed into a string of characters in 7½ min and stored in binary form. A second program organizes the character string into a timed manuscript of lines. The character set and all character functions, e.g., the subset of characters that start lines, are entirely software defined. A completely open format and the flexibility of user-defined software grammars facilitate the entry of subjects, actions (both momentary and continuous), objects, and other contextual information in whatever form the user requires. The SSR system is described both as a system of ideas about the problems of encoding observations for computerized transcription and as it is currently embodied in a specific set of software and field-tested hardware. The rationale for each major aspect of the system is presented in detail from a user's point of view.

Within the limitations of a catalogue of categorical behavior patterns, observers of behavior may wish to record the incidence, duration, coincidence, and sequence of the patterns observed in real time. Voice notes on magnetic tape or ink pen tracings on moving paper have been used to record behavior in these terms, but both of these methods entail tedious

and time consuming transcription before analysis. On the other hand, most manual recording methods are compromises; checklists or clocks and counters lose sequence information, while handwritten notes tend to lag behind the action. Further losses occur in the course of frequency shifts of the observer's attention from action to notebook and back. Lastly, if quantitative analysis or searches of the data are to be done by computer, all of the above methods require keypunching.

This work (Publication No. 15-021 of the Wisconsin Regional Primate Research Center) was supported by Grant GB-7551 from the National Science Foundation to Drs. John T. Emlen and Gordon R. Stephenson and by Grants RR-00249 and RR-00167 from the National Institutes of Health to the Laboratory Computer Facility, University of Wisconsin, and to the Wisconsin Regional Primate Research Center, respectively. The time and talents of many members of the Department of Zoology, University of Wisconsin, were directly involved in the creation of the SSR system. Special thanks are due to Kenneth G. Olesen for reorganization of the subassemblies of the keyboard into a more readily serviced arrangement and for fabricating most of the units now in use. Special thanks also to Richard J. Ganje and John E. Dallman for design and fabrication of the keyboard cases. The advice and assistance of Deborah L. Yoshihara, Walter R. Holthaus, Cheryl M. Hughes, and Peter E. Berryman are much appreciated. The field experiences of Ann L. Bleed and Timothy D. Johnston helped in revising earlier versions of both keyboard and software. The first author is Assistant Scientist, Wisconsin Regional Primate Research Center, 1223 Capitol Court, University of Wisconsin, Madison, Wisconsin 53706. The second author's address is The Eye Research Institute of Retina Foundation, 20 Staniford Street, Boston, Massachusetts 02114. The third author is Lecturer, Department of Physical Education—Women, Lathrop Hall, University of Wisconsin, Madison.

With these problems in mind, several investigators have devised keyboard event recording systems that encode entries and time onto magnetic tape for playback through interface circuitry to a Teletype or keypunch (Dawkins, 1971; White, 1971; Sykes, Note 1; Sackett, Stephenson, & Ruppenthal, Note 2). Touch control of keyboard entries allows the observer to follow the action without shifting his attention. While an elaborate hardware interface for decoding information on tape is required in such systems, the cost is more than recovered by the elimination of manual keypunching. Transcription of the taped information by means of a hardware interface, however, typically requires that all observations be entered in a standard format of preset length. This length is typically limited to six characters. Furthermore, the character set is typically limited to numbers, thereby precluding the use of mnemonic alphabetic codes for the labels of categorical behavior patterns. Many who have tried

these systems have found the preformatting requirement and the relatively small number of characters per entry too restrictive for their work.

Editing the transcripts from most keyboard event recording systems presents further problems. The Teletype produces a punched paper tape transcript of the record, but working with paper tape is unwieldy at best. Editing a transcript on punch cards is easier but can be time consuming if there is more than one entry per card. On the other hand, if there is only one entry per card, but many entries per record, storing the card decks can be expensive.

The problems outlined above severely limited application of these data recording methods to our studies of communication in primates (Stephenson, 1967, 1973, 1975). The complexity, density, and quantity of the data in our studies required a new approach to data collection and management. In the course of the last 6 years (see Note 3), we have built, programmed, and field tested a keyboard event recording system that overcomes the problems inherent in the aforementioned methods. Instead of committing the system to an elaborate hardware interface, we took our cue from the increasing availability of small but powerful computers and produced a system that is primarily software defined.

The primary design objectives of our system were that it be (1) able to transcribe rapidly data encoded on audio magnetic tape, (2) flexible enough in format to be applied to a wide range of user interests, (3) lightweight, battery operated, and reliable enough for field applications, and (4) simple to use. The sixth version of the SSR system (for senders, signals, and receivers) has been implemented on the BOZO (botany-zoology) Computer Facility in the Department of Zoology, University of Wisconsin, Madison, for more than one year. More than 500 h of data have been transcribed. In addition, the system is in the process of being implemented on small computers in the Department of Psychology and in the Waisman Center on Mental Retardation and Human Development at the University of Wisconsin, Madison.

OVERVIEW

The SSR system is an event recording system that encodes the incidence, duration, coincidence, and sequence of entries in real time onto magnetic tape for subsequent high-speed transcription by computer. The system includes a keyboard for encoding categorical observations of ongoing events, an audio tape recorder for recording keyboard output, a playback tape deck, signal conditioning circuit and analog to digital channel for input to a small computer, and software for transcription and timing of the encoded record (see Table 1). The system has been used to study the temporal structure

Table 1
Overview of the SSR System

Observe	Record
catalogue code	keyboard encoding signal audio tape recorder data tape
	Interface
	data tape playback deck signal conditioner "binary" wave form
Transcribe—First Stage (read data tape)	Transcribe—Second Stage (organize characters)
A to D channel	binary form from disk file 1
variable timing reference	computer
computer	program SSRA
program SSR	timed manuscript of lines
character string in core	screen or printer for display
binary form on disk file 1	character form on disk file 2
	computer tape

of crayfish behavior, feeding sequences of spiders, social interactions among monkeys, and the use of space by children.

The SSR system keyboard (Figure 2) has a complete set of alphanumeric characters (A-Z and 0-9) and 12 special grammar characters (+, -, /, *, !, \$, @, =, ?, &, %, and "segment"). The character set and all character functions are all software defined. For example, in our current applications of the system, +, -, /, *, !, and "segment" are defined as carriage control characters. In the first stage of transcription (see SOFTWARE section), the encoded keyboard record is decoded simply as a continuous string of characters. In the second stage of transcription, this character string is organized into a manuscript of lines. Whenever any one of the software-defined carriage control characters is found in the character string during the second stage of transcription, a new line is begun in the manuscript. Additional character functions further tailored to a user's particular needs can be defined in later stages of software processing. If, for example, in a user's application of the system, an entry coincides with a line, then one of these carriage control characters can be used to indicate the beginning of a new entry. In our current applications, lines beginning with a + indicate the start of an event. Lines beginning with a - indicate the termination of a continuous event (momentary events, by definition, do not require termination). These functions are software defined in the data analysis programs (see example in section titled AN APPLICATION) and are not inherent in the keyboard encoding scheme or specified in the basic transcription software. In essence, the format and functions of entries in a particular record are completely open, restricted only by a software limit of 54 characters per line and the structure of the user's problem.

To take account of the vagaries of audio tape recording under field conditions, the encoding scheme of the SSR system keyboard is very conservative (highly redundant and with a "discrete channel" for each character in the character set) and the decoding process at transcription includes extensive checking for electronic errors such as tape dropouts. Briefly, the scheme is based on phase encoding. A character is entered into the record by a single push of the character's key on the keyboard. The keys are alternate action (push-ON/push-OFF) switches. When pushed, the state of the key changes either from OFF to ON or from ON to OFF. The primary information is the change in state; with the exception of the "segment" key, which produces an open angle bracket when going from OFF to ON, and a closed bracket when going from ON to OFF, the direction of the change of a key's state does not add information to the record. The state of each of the 48 data keys is sampled in sequence by a multiplexing circuit that sweeps the keyboard 20 times per second. As each key is sampled, its current state (ON or OFF) is encoded as one of two square waves that differ in the direction of the zero crossing transient. (In the square waves used, the signal trace passes through a zero value level of voltage as it goes from positive to negative and vice versa. See Figure 5.) Multiplexing the phase encoded state yields a signal that can be recorded on audio tape. In contrast to digital encoding schemes used with incremental tape recorders, the SSR system encoding scheme permits simultaneous recording of keyboard output and observer voice notes, animal vocalizations, or other analog data on parallel tracks of the same audio tape.

In the course of recording and playback, the precise shape of the keyboard encoding signal is partially degraded (see Level D in Figure 5). At transcription, a signal conditioning circuit (see **HARDWARE** section) incorporating a Schmitt trigger reshapes the output of the playback deck into binary information that can be evaluated by the computer. The Schmitt trigger is a voltage level detector; as each zero crossing transient in the playback signal passes a preset voltage level above zero, the trigger shifts from one to the other of its two output values.

In the first stage of transcription, an assembly language subprogram tracks the sequence of changes in the value of the trigger output via an analog to digital input channel to the computer. The subprogram decodes pairs of values (one value for each half of the square wave form representing the state of a switch) into the state of the key switch represented at each position in the multiplexing sweep. Our current computer (Datacraft 6024/5) has a 24-bit word length and our current keyboard has 48 character positions. As it is decoded by the subprogram, the key's state is stored as a one or a

zero (ON or OFF, respectively) in the bit position of a double word of memory that corresponds to the key's position in the multiplexing sweep. During the delay portion of each sweep (see Level C in Figure 5), the decoded data portion is compared to that found in the previous sweep. If the values of the two double words representing the current and the preceding sweeps are different, the bit that differs is found and converted to its corresponding character, the sweep count is incremented, and the character is stored together with its sweep count in the string of characters that makes up the decoded record at the end of the first stage of transcription. If the two double words are not different, the program simply increments the sweep count and waits for the data portion of the next sweep.

By counting sweeps, the time base of an SSR system record is independent of tape speed. During the first stage of transcription, sweeps are counted from the first character detected in the record. Since the sweep rate is 20 times per second, the sweep count represents the relative time of the character in the record. In the second stage of transcription, the character string is partitioned into lines as the program encounters the software-defined carriage control characters. In the resulting manuscript form of the record, only the sweep counts of the line starter characters are retained; they preserve the relative time of each line. Relative time is then translated into real time through the software interpretation of a special time statement (!Tnnnn, where n is a digit, e.g., !T0802) that is entered on the keyboard by the user during the observation session.

The SSR system is reasonably efficient. While the encoding signal is recorded at a tape speed of 1-7/8 ips, it is played back for transcription at 15 ips. Thus, 1 h of data on tape can be read through the first stage of transcription of 7½ min. At these tape speeds, the signal is well within the frequency response capabilities of most commercially available ¼-in. audio tape recorders. In contrast, a frequency (tone) encoding scheme with the same number of discrete channels of information would be technically more difficult (hence more expensive) than the SSR system scheme and without elaborate interfacing equipment could not offer as favorable a ratio of observation to transcription time. In an additional few minutes (depending on the number of characters entered into the record), the data are organized by the second stage of transcription into a timed manuscript, displayed on the CRT for previewing, and written out onto computer tape for storage and subsequent data processing.

AN APPLICATION

The basic steps in applying the SSR system to a research problem are listed here in chronological

order and discussed in detail in appropriate sections below (see Table 1). Before the system can be put to optimal use, a catalogue and code for entering categorical observations must be defined by the user. These typical steps then follow: (1) entering the observations onto the keyboard in terms of the catalogue code; (2) transcribing the keyboard encoded record at the computer; (3) printing the raw, timed manuscript; (4) writing a copy of this manuscript from core or disk file onto computer tape; (5) verifying the information in the manuscript against the information on the parallel voice track, making editorial notes directly on the printout from Step 3 as soon as possible after the observation recording session; (6) editing the raw manuscript into a corrected version by using the Teletype and software editor at the computer to manipulate the retrieved copy that had been written on computer tape in Step 4; and (7) replacing the raw version on computer tape with the edited one for later reference and data analysis. For safety, a second copy of the edited version is stored on a second computer tape. We then rerecord the voice track onto minimum-quality tape at 15/16 ips, should later review of the voice record be necessary. The original high-quality ¼-in. keyboard recording tape is erased and used again.

To demonstrate the SSR system in more detail before explicating the individual steps outlined above, we will describe our use of the system for recording drinking behavior in a 1-h priority of access to water test among members of a troop of 45 rhesus monkeys (*Macaca mulatta*) after 22 h of water deprivation. The basic assumptions in the use of this test are that the deprivation has made the monkeys thirsty and that the sequence in which individuals come to drink or in which they attain some minimum cumulative total of drinking time (here, 10 sec cumulative time; see also Alexander & Harlow, 1965; Weisbard, 1975) is related to their rank position in the social structure of the troop. Discussions of the merits (Boelkins, 1967; Clark & Dillon, 1973) and problems (Bernstein, 1970) of this kind of priority of access test as a measure of social rank are not germane to the theme of this report.

In our water test, two observers enter their observations of the animals' access to water on SSR system keyboards. For each, keyboard output is recorded on one track of a quarter track stereo tape recorder (Uher 4400 Report). To facilitate keeping up with the often-hectic pace of the action, observers are encouraged to verbalize their observations into a microphone as well as enter them on keyboards. The highly redundant and somewhat more interpretive voice record is recorded on the second of the stereo pair of recorder tracks. The primary observer's task is to record the drinking behavior of individuals as they drink at a water fountain. The fountain is the

Table 2
A Grammar for Entering Water Test Data on
an SSR System Keyboard

1. There are two kinds of drinking events:
 - (a) A sip is a momentary event, arbitrarily assigned a duration of .5 sec;
 - (b) A drink is a continuous event and requires termination.
2. + indicates the onset of drinking.
3. - indicates the offset of drinking.
4. D after a + indicates that, in the observer's judgment, drinking continued for more than .5 sec, and the event is a drink.
5. The names of individuals are two-digit numbers.
6. The name must appear after the first + in each of an individual's drinking bouts, but is not obligatory after that.
7. When a name is used, it is entered immediately after the +.

only source of water in the test environment. Its physical design allows only one animal to drink at a time. The primary observer is separated from the fountain by a pane of glass, but is able to monitor the drinking behavior from a position about 18 in. from the water source. As the animals, one by one, drink at the fountain, the primary observer records the onset of drinking, the name of the individual drinking, and the offset of drinking, according to the set of rules in Table 2. The secondary observer, using a much more complex set of rules and much less rigorous sampling scheme, records observations of social interaction and context in the test environment that will be used in interpreting the results of the water test, especially those interactions that occur in the immediate vicinity of the water fountain or that involve the major sources of dependent rank (see Kawii, 1958; Missakian, 1972; Varley & Symmes, 1966). This more elaborate set of rules is a grammar devised for recording the subject, action, object, modifiers, proxemic relations, spatial location and immediate corrections to any of the above in real time in field studies of primate social behavior and communication.

The simple set of rules for the primary observer in Table 2 was constructed from an analysis of manually prepared transcriptions of four voice-recorded water tests. The analysis gave us a feel for the temporal structure of drinking bouts, the rapidity and frequency with which animals supplanted one another at the fountain, and the kinds of observer errors that occurred and would have to be corrected before the data in the record could be used. (We believe that event recording systems like the SSR system cannot be applied advantageously until the user has this kind of information.)

The rules in Table 2 work together to produce the record of drinking behavior in Column 4 of Figure 1. In Line 23 (see Column 2), for example, 06 began drinking at Frame 11306 (Column 3). The female monkey's drink was brief, less than the time it took the primary observer to push the +, assess the situation, be sure of the identification and enter her

RAW RH4 H2O 140375 TEST AND OBSERVER

REAL TIME	RH4 H2O 140375 DLY KB6/2	RH4 H2O 140375 GRS KB6/11
	LINE FRAME ENTRY	LINE FRAME ENTRY
15:00:25.6	3 9313 C	
15:00:27.9	4 9358 !D140375	
15:00:32.4	5 9468 !DBLY	
15:00:36.0	6 9521 !SRH4 H2O	3 9527 C
15:00:36.6		4 9533 !OGRS
15:00:38.9		5 9578 !D140375
15:00:44.5		6 9691 !SRH4 H2O
15:00:50.1	7 9802 /EXPT	
*15:01:00.0	8 10000 !T1501	7 10000 !T1501
15:01:08.9	9 10178 !W	
15:01:09.7	10 10195 +400044	
15:01:19.6	11 10293 -	
15:01:20.0	12 10400 +06D	
15:01:25.5	12 10510 -	
15:01:27.2		8 10546 +79eLDT
15:01:34.4	14 10689 +78D	
15:01:37.6	15 10753 -	
15:01:38.2		9 10764 +36eCWL
15:01:40.2	16 10804 +0	10 10813 +85eLDM
15:01:41.2	17 10825 /44	
15:01:43.6		11 10873 +89eLDF
15:01:46.0		12 10921 +06eT
15:01:49.1		13 10982 10982 42eT
15:01:51.3	18 11026 -	
15:01:52.9	19 11059 +0	
15:01:53.9		14 11079 +82eLDeCWR
15:01:55.4	20 11108 -	
15:01:57.6	21 11153 +0	
15:02:01.2	22 11224 -	
15:02:02.2		15 11244 +79L
15:02:05.2	23 11304 +06	16 11308 +06eA
15:02:07.5	24 11350 +40D	17 11342 +06L
15:02:09.0	25 11380 -	18 11375 +80AeLDT

Figure 1. Raw manuscripts of the primary (DLY) and secondary (GRS) observers' SSR system keyboard records of a priority of access to water test. Titles spanning Columns 2-4 and 5-7 include the troop name (RH4), kind of data (H2O), day, month, year, observer's initials, and the keyboard used. The title is entered into the record with the Teletype at the time of transcription. Column headings are inserted into the record by the basic transcription and initial data processing programs. The frame is the sweep count in which the line started. Real time is rounded down to the nearest .1 sec. The two records were recorded on separate keyboards and tape recorders. Just before the water was turned on, the secondary observer began a time statement in each record by simultaneously pushing the ! key on each keyboard. The records are printed here in temporal registration by third-stage software that aligned the two records in register on the initial, simultaneous time statements. Where a line in one manuscript begins within a half second of the beginning of a line in the other manuscript, the lines are printed on the same line of the printout and the frame of the primary observer's line is used to compute the real-time equivalent. To facilitate editing after this printout is verified against the parallel voice track, separate line counts are kept for each manuscript.

name. This kind of very short drink is called a sip. A sip is arbitrarily defined as a momentary event of .5-sec duration. A drink is defined as a continuous event and, as such, requires termination by entry of a -. For example, just after 06 sipped, her son, 40, came to the fountain and, beginning at Frame 11350, drank for 1.5 sec ([11380-11350]/20 sweeps per sec). Using a + to indicate the onset of drinking and only then distinguishing a drink from a sip enhances the accuracy of recording the temporal location of drinking behavior (a + is a line starter; its relative temporal position in the record is preserved throughout data processing). After beginning the entry with the +, the observer has time to consider the content of the entry in greater detail. If drinking behavior continues, a D is entered after the +, and the entry becomes a drink rather than a sip. A drink requires termination; it must be

followed by a -. Using a D in the record to distinguish a drink from a sip provides a grammatical (or syntactic) basis for error checking in our subsequent data processing programs. Thus a +D (with or without a name) followed by a + line is detected as an error, as is a - followed by a -. A + (with or without a name) followed by a - is also an error because a sip is defined as a momentary event and cannot be terminated, but a D can be forgotten.

Other of our grammatical conventions are also apparent in the water test data in Column 4 of Figure 1. In Line 8 (see Column 2), !T1501 was entered as the first time statement in the record. By putting an asterisk in front of the real-time notation in Line 8, the transcription program has labeled this entry as the base for computing the real time of each line. The !W in Line 9 indicates the moment at which the water was turned on. An editorial mark

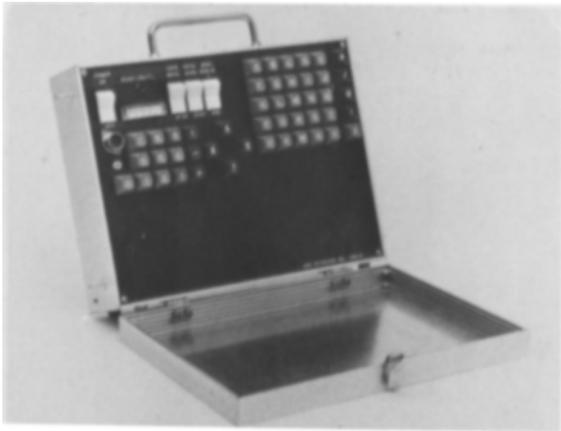


Figure 2. Front view of the SSR system keyboard. Half hinges allow the cover to be detached. Side brackets are for attachment to the harness that suspends the keyboard in front of the user in the field. See text.

around the !W notes that a + should be edited into this line after the W. A + is a line starter; the only way that a + could appear within a line is by having been placed there with the edit program. We place it there as the final step of editing. If our water test data processing programs do not find a + after the W in the !W line, the data processing run is terminated by the program rather than risk using computer time to process unedited data. There is also an editorial mark around the incomplete correction in Line 10. The ? is a within-line grammar character. During the test, it was used by the observer to correct a misidentification. Our data processing programs scan each line and process only the data characters that occur after the last ? in a line. This convention preserves the line starter, hence the correction entered after the last ? will be associated with the appropriate relative time. Our rule for errors sensed at the time of recording is, when in doubt, enter the data again. In Line 10, the name was corrected immediately after the error was realized in the course of recording the data, but the D was forgotten and must now be added during editing. If the observer had not forgotten the D after the 44, no editing would have been necessary in this line, for the correction would be properly interpreted by the data processing software. The / starting Line 17 introduces a comment into the observer's keyboard record that, in this instance, refers to a comment on the voice track concerning the name correction above. Lines beginning with a / are treated as comments and ignored in the data processing programs.

While these conventions are used in the water test record of the primary observer, they are all drawn from the more complex grammar used by the secondary observer. A more complete description of their use will be deferred until they can be presented

in the context of that larger and more comprehensive grammar. They are included here to suggest the power of employing a software-defined grammatical method for entering "on-line" observations of complex behavior and social interactions and to emphasize that the open format of the SSR system readily accepts the highly variable entry length that such a method requires.

HARDWARE

From our point of view, the following description of hardware embodies a system of ideas rather than the ultimate technological solution to the problem of encoding observations for computerized transcription. Our intent is to describe the circuits in enough detail so that they can be followed directly or serve as the basis for new designs by others. Requests for technical information beyond that which is presented here or for copies of our updated circuits and for assistance in getting copies of our printed circuit boards should be sent to the senior author of this report. Critical comments and suggestions would be much appreciated.

Keyboard

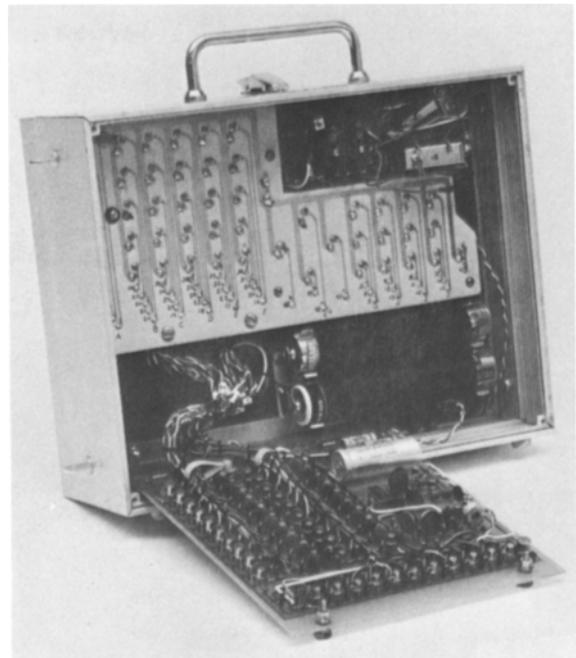
Physical description. Outside dimensions of the keyboard are $12 \times 9 \times 3$ in. The detachable cover measures $12 \times 9 \times 1$ in. Weight without cover is 6 lbs; the cover weighs 1 lb. Case and cover sides are made of extruded aluminum channel (Vector Company) cut and fashioned with mitered corners. Bottom and top are made with .062 (Brown & Sharp 14 ga) aluminum panels. (See Figure 2.)

Keys, control switches, meter, and output jacks are set into a punched aluminum panel (Nu-Way Manufacturing Company, Skokie, Illinois). A battery holder and the printed circuit boards for the key and encoding circuits are attached to the underside of the punched panel (see Figure 3). To facilitate service access to the circuit boards, removal of four screws allows the bottom panel of the case to slide out of its track in the extruded aluminum sides. The encoding circuit board is attached with a hinge so that it can be swung out for easy access to both of its sides.

Power is supplied by a set of six C-cell batteries. Current draw is less than 20 mA; at a usage rate of 4 h per day, battery life exceeds 100 h at 70°F.

The two output jacks are at the upper left of the punched panel. The large 5-pin jack (Switchcraft/Preh) accepts a plug like that on the Uher microphone cable. A locking collar prevents accidental disconnection. A similar plug connects the other end of the cable with one of the microphone input jacks on the tape recorder. Either end of the cable may be used at the keyboard jack. The single-pin jack below the 5-pin jack provides a connection to the keyboard output

Figure 3. Rear view showing circuit subassemblies of the SSR system keyboard. Bottom panel slides out for access to circuit subassemblies. The encoding circuit board is hinged to open for easy access to components. In this view, power supply and multivibrator (clock) are on the right portion of the encoding circuit board, ring counter is on the left. The upper circuit board is for the data keys. A key can easily be replaced by unsoldering its pair of contacts in the upper board while pulling it away from the front panel of the keyboard. Cut-out portion of the upper board allows access to the meter and rocker switches.



for externally synchronizing an oscilloscope when examining the wave form of the encoding signal. The other components on the punched panel are discussed after description of the encoding process.

Encoding circuit and logic. The functions of the SSR system keyboard are to produce a discrete representation of the state of each of 48 data keys, to multiplex these 48 discrete representations into a single encoding signal, and to provide a time base for the record that is independent of the tape speed at which it is recorded or played back.

The discrete component circuit that currently performs these functions is diagrammed in Figure 4.

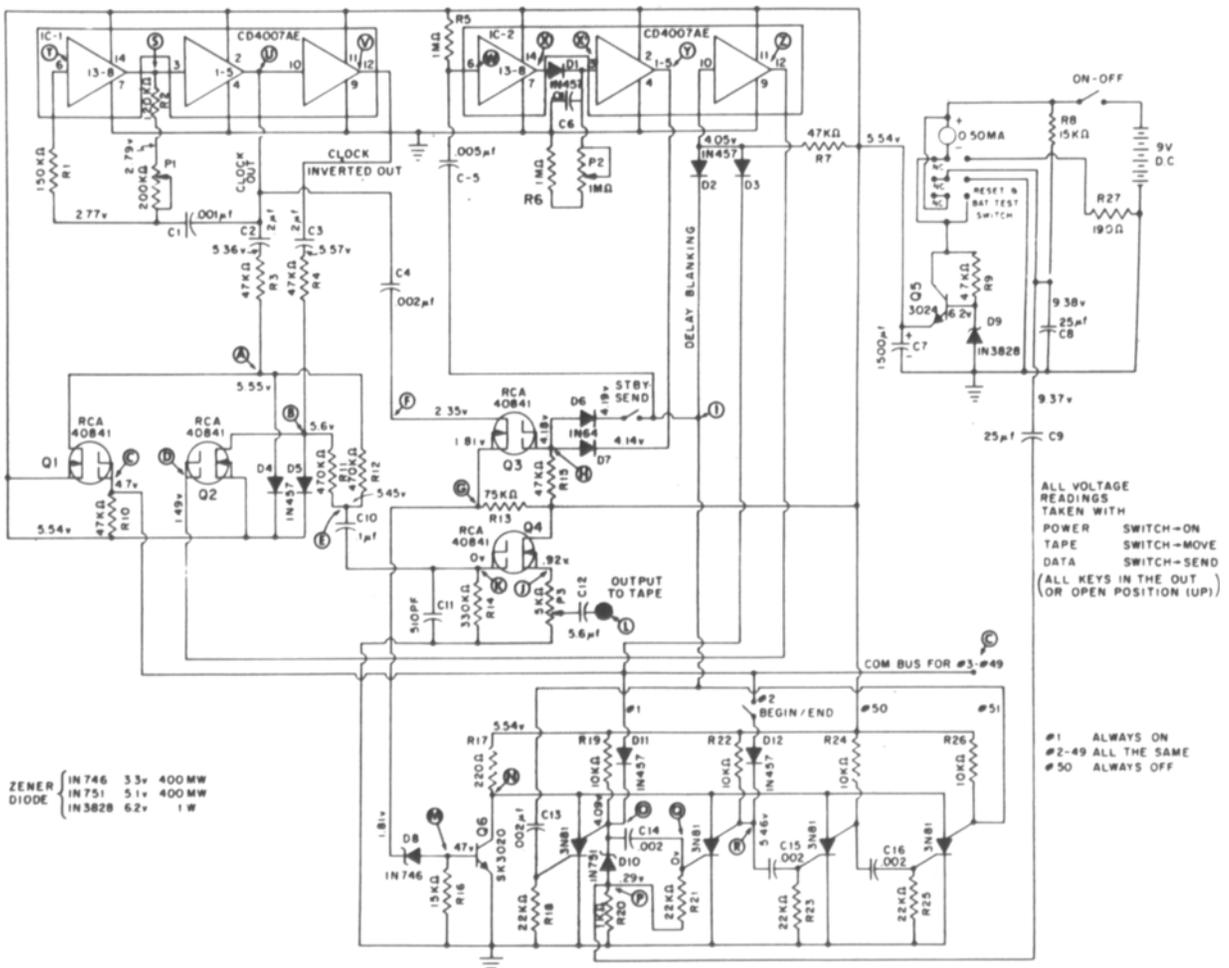


Figure 4. Discrete components version of the SSR system keyboard encoding circuit. Major sections are the power supply, multivibrator (clock), ring counter, and data buss to the bi-stable switch.

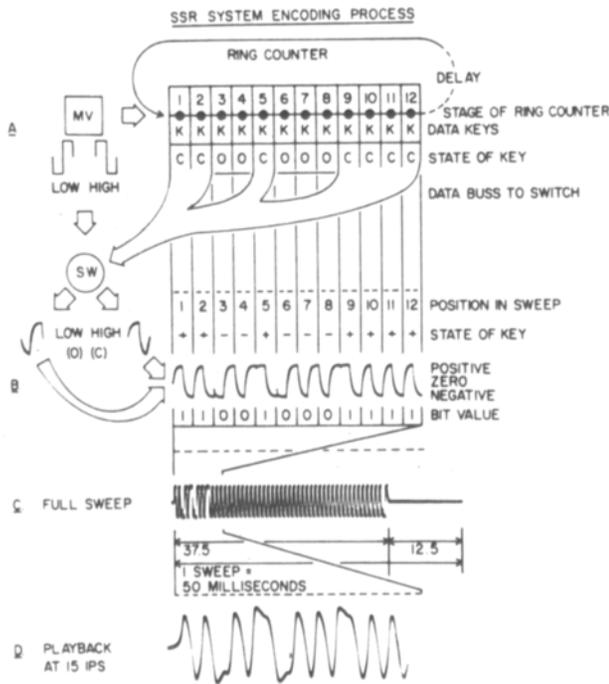


Figure 5. Representations of data at serial levels of the SSR system encoding process. A multivibrator (MV) generates two complementary square waves at a cycle time of .75 msec. The wave form with the descending zero crossing transient drives a ring counter. The ring counter supplies a bias voltage to the keys in sequence one at a time. If, when supplied, the key is closed (c), the bias voltage causes the bi-stable selector switch (SW) to go into its high state and the square wave form with the descending zero crossing transient is routed to output. If a key is open, the bias is not supplied, the bi-stable switch goes low, and the square wave with the ascending transient is output. Thus, the state of each data key on the keyboard is phase encoded in sequence and thereby multiplexed into a single signal for output to an audio tape recorder. The oscilloscope trace of phase encoding wave forms at Level B represents the states of the first 12 key positions depicted at Level A. The trace of the full sweep, with both data and delay portions, is shown at Level C. The typical degradation of form from recording the encoding signal at 1-7/8 in./second (ips) and playing it back for transcription at 15 ips can be seen in the trace at Level D.

More recently available devices like C-MOS integrated circuits could be substituted for portions of the present circuit with substantial savings in cost and assembly time. The version presented here is a proven circuit. Sixteen units of this design are now in use in nine different research groups on the University of Wisconsin—Madison campus.

The logic of the encoding circuit is redrawn in Figure 5. The primary functional components of the circuit are a multivibrator, a 51-stage ring counter (or continuous shift register), a set of 48 alternate action data keys, and a two-state (electronically bi-stable) selector switch. Each data key is connected to a different stage of the ring counter. Driven by the multivibrator, the ring counter sequentially samples the data keys. As a key is sampled, its state is phase

encoded for output to the tape recorder. The oscilloscope trace at Level B in Figure 5 shows the output from stages represented at Level A. The relationship of these stages to a complete sweep of all stages is shown at Level C. The sequence of zero crossing transients in the trace corresponds to the sequence in which the ring counter samples the data keys. From the phase encoding process, the direction of the transient corresponds to the state of the key sampled there. At transcription, detection of a change in state results in storage of the character that corresponds to the data key sampled at that position in the sweep.

In the present circuit, the multivibrator serves as both a clock and as the source of the encoding signal. The clock function is based on the multivibrator's .75-msec cycle time. (This function could be performed with a crystal oscillator and solid state dividing circuits). Each time it cycles, the two sides of the multivibrator put out complementary square waves that differ in the direction of their zero crossing transients. These two forms, with an ascending or descending transient (labeled LOW and HIGH in Figure 5, respectively), are used to phase encode the states of data keys on the keyboard.

The function of the ring counter is to sample the state of each data key in sequence and thereby multiplex the 48 data key channels into one channel of information for recording on magnetic tape. A discrete sweep position for each key rather than binary coding of the character set in fewer sweep positions enhances error checking at transcription and facilitates reconstruction of a record should one of the positions fail. The ring counter's circuit is so designed that only one stage is ON at a time. The ON position is driven around the ring by output from the multivibrator. Each time the multivibrator cycles, the positive wave front of the descending square wave form turns the present ON stage OFF. In going from ON to OFF, a transient voltage is produced and passed along to the next stage in the ring. The transient voltage turns the next stage ON. The new ON stage stays ON until the next cycle of the multivibrator. As the ON position moves from stage to stage around the ring, it passes a bias voltage to the corresponding data key for approximately .75 msec.

The function of the two-state selector switch is to phase encode the state of a data key by routing one of the two complementary square wave forms to the keyboard output circuit. Data keys are push-ON/push-OFF alternate action switches (Type M61-0800; Cherry Electrical Products Corporations, Waukegan, Illinois); a key is either CLOSED (in its physically down and electrically on position) or OPEN (physically up and electrically off) during the period when its corresponding stage is the ON position of the ring counter. When a data key is in the CLOSED

position, the bias voltage from its corresponding stage is passed to the selector switch, the switch goes into its HIGH state, and the square wave with the descending zero crossing transient is put out to the tape recorder. When the data key is in the OPEN position, the bias voltage is not passed, the selector switch goes into its LOW state, and the square wave with an ascending transient is put out to the tape recorder.

Beginning at the second stage, each of the 48 data keys is wired to a different stage of the 51-stage ring counter. The portions of the encoding signal that correspond to these stages can have either an ascending or a descending zero crossing transient, depending on the state of the corresponding key. To provide fixed cues for the transcription program, the 1st and 50th stages are hardwired as if the key corresponding to the 1st stage were CLOSED and that corresponding to the 50th were OPEN. In the sweep at Level C in Figure 5, the first zero crossing transient represents the first stage hardwired in the CLOSED position, the 2nd through 11th crossings represent the keys as indicated at Level A in Figure 5, the 12th through 49th crossings all represent CLOSED keys (put in the position on the keyboard for the example), and the 50th crossing represents the stage hardwired in the OPEN position.

The 51st stage of the ring counter does not appear as a zero crossing transient in the encoding signal. When the ON position comes to the 51st stage, it initiates a delay circuit that uncouples the multivibrator from the ring counter for 12.5 msec. During transcription (at eight times real time), the delay provides about 1.5 msec for error checking and character processing. In real time, the delay plus the data portion of the sweep ($50 \text{ stages} \times .75 \text{ msec/stage} = 37.5 \text{ msec}$) requires 50 msec total. At a constant rate of 20/second, the sweep gives a time base to the record that is independent of the tape speed at which the encoding signal was recorded.

Durations of the data and delay portions of the sweep can be independently set to their correct values of 37.5 and 12.5 msec, respectively, by adjusting the appropriate potentiometers in the circuit while watching the keyboard output on an oscilloscope. A third potentiometer is used to set the amplitude of the encoding signal to 70 mV. Manual adjustment of the timing potentiometers can yield a time base that is accurate to within 10 sec in an hour. Greater accuracy would be routine, were a crystal oscillator used in place of the multivibrator. Where the time base in our records is discovered to be in error, third stage software (see sections AN APPLICATION and SOFTWARE) compares the number of sweeps found between the first and last time statements in a record to the number there should have been (as calculated from the difference between the stated

times), computes a ratio, and corrects the frame counts of all of the lines in the record.

As can be seen in the oscilloscope trace at Level B in Figure 5, the encoding signal from the keyboard is not simply a train of square wave forms. To avoid "ringing" in the audio circuits of the tape recorder, the sharp wave forms put out by the multivibrator are modified by rounding their leading edges just prior to output. The notches that sometimes appear in the signal between the modified square waves, especially where the phase form following is the complement of the form preceding, are the recordable aspects of spurious zero crossing transients. The zero crossing portion of these transients does not appear in the data record on tape, however, because their rise times are in the microsecond range, beyond the frequency response capability of the Uher 4400 Report tape recorder recording at 1-7/8 in./second; the spurious transients are simply filtered out of the encoding signal in the process of recording it.

The form of the encoding signal permits the user to verify in the field all aspects of proper keyboard operation other than timing without extra equipment other than the tape recorder and the keyboard themselves. With the recorder in RECORD mode and PAUSE, the user can listen to the input to the recorder by adjusting the playback volume. When the keyboard is operating properly, the delay portion at the end of each sweep interrupts the approximately 1.34-kHz tone set up by the data portion of the sweep, and the sound has a "chatter" quality. In contrast, a steady tone means that one of the stages is malfunctioning and the ON position of the ring counter is not able to propagate all of the way to the 51st stage, where it would normally initiate the delay. No tone, of course, means a malfunctioning recorder, a bad connection between recorder and keyboard, or a malfunctioning keyboard.

At a more detailed level, the user can test whether each data key is functioning properly by setting all of the keys into their physically down and electrically on position and then, one by one, pushing each key several times while listening to the sound. With all of the keys down, the wave form in the data portion of each sweep is nearly a sine wave and has a nearly pure tone quality. Each time the key being tested is pushed into its physically up and electrically off position, the wave form in its corresponding portion of the sweep is switched to its opposite phase and the tone quality changes. No change in tone quality indicates a mechanically malfunctioning key. It is still possible to lose data if a key becomes malfunctioning between tests. Having a "discrete channel" for each key (per the multiplexing process), a properly constructed grammar and code, and a redundant voice record on the parallel recorder track, however, markedly facilitates reconstruction of

a record in which data characters are missing because of a malfunctioning key. It is best, of course, to monitor regularly the sound of the keyboard encoding signal, to examine occasionally its wave form on an oscilloscope, and, if in the field, to send sample data tapes periodically from field to computer lab for transcription.

Test points for troubleshooting the circuit assembly and maintenance are labeled as encircled letters on the circuit diagram in Figure 4. Measured values are also noted, but these may vary some from keyboard to keyboard. Instructions for using these test points, and wave forms expected at each of them, are detailed in a user's manual that is available upon request. Values, sources, and approximate costs of all components are also detailed in the manual, together with complete instructions for operating all aspects of the system.

Rationale of keyboard organization. Our particular layout of the components on the punched panel is designed to facilitate rapid entry of the subject, action, object, etc., in communication events among primates. The layout can be customized to another user's design by submitting a blueprint for programming the punch press (at Nu-Way Manufacturing Company) and making the appropriate changes in the negative for printing the circuit board for the keys.

In our layout (Figure 2), keyboard switches, data keys, meter and jacks are all located in one half of the front panel so that the keyboard may be worn like a cigarette tray, supported by a harness at about waist level in front of the user. This arrangement facilitates following moving subjects while taking data in the field. To provide room for the user's hands, the keyboard is worn with the component section away from the body.

Keyboard control switches are arranged according to function at the upper left of the panel. Rocker switches are used because they indicate the status of the function they control without drawing power (as indicator lights would). The rocker switch to the left of the meter is the POWER ON-OFF switch. Three other rocker switches are grouped to the right of the meter. The leftmost of these, labeled TAPE, MOVE, STOP, utilizes a circuit in the keyboard track line input jack to operate the PAUSE mechanism on the Uher 4400 Report tape recorder; it can be used for remote starting and stopping of the tape transport. For example, in the field, the user can preset the tape recorder in record mode and then carry it in a backpack frame while wearing the keyboard in front. The middle rocker switch is the DATA, SEND, STBY (standby) switch; it switches the multivibrator into or out of the ring counter circuit at a point that insures that the first sweep of a record begins at Stage 1 and that the last sweep ends at Stage 51. The keyboard sends the encoding signal to the tape

recorder only when the power switch is ON and the data switch is in the SEND position.

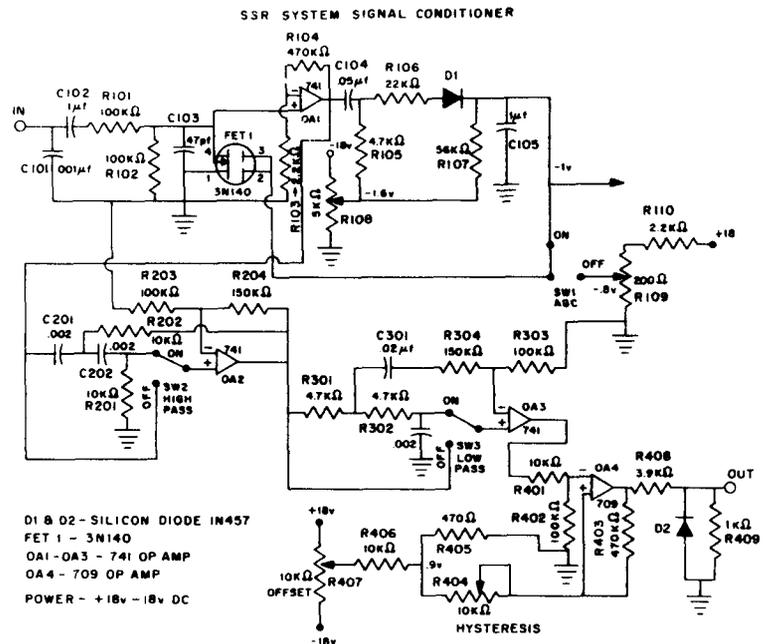
The rightmost rocker switch is the SEGMENT, BEGIN, END switch. This is the exceptional data key that produces two different characters in the record, depending upon whether the key switches from OFF to ON or from ON to OFF when it is pushed (from END to BEGIN and from BEGIN to END, respectively). Pushing the rocker into the BEGIN position produces an open angle bracket in the record; pushing it into the END position produces a closed angle bracket. This convention is employed to mark the beginning and the end of a data segment within a record, e.g., each sample from a series of focus subjects in an hour-long record. In subsequent data processing, scanning for the open bracket in the line starter position of the data format can speed sorting for a particular segment.

The last control switch is the round momentary push button above the meter. Its primary function is to RESET the ring counter and turn on its first stage. The circuitry of ring counters is such that they can propagate more than one ON position around the ring at the same time. The encoding signal produced by more than one ON position cycling around the ring in an SSR system keyboard would be uninterpretable by the transcription software. When there is more than one ON position being propagated, current draw is greater and the reading on the milliammeter is higher than normal. To insure that only one ON position is being propagated around the ring, the circuit is routinely initialized each time that the keyboard is powered up, pushing the RESET button once after the power has been switched ON. While the RESET button is depressed, it also engages circuitry that changes the meter from a milliammeter to a voltmeter, thereby enabling the user to test voltage level of the battery power supply.

The left to right order of the keyboard control switches is the sequence in which they are used to initiate a keyboard record on tape after the keyboard has been turned ON and RESET, i.e., TAPE MOVE, DATA SEND, SEGMENT BEGIN. A record is terminated in the reverse order, SEGMENT END, DATA STBY, TAPE STOP, and (optionally) POWER OFF. This procedure results in nesting the record of keyboard output between a start and a stop of the tape transport, and within that, nesting a data sample between an open angle bracket and a closed angle bracket.

Data keys are arranged in blocks to facilitate the entry of + or -, digits for the subject's name, etc. (see AN APPLICATION). Within the numeric block, the 0 through 9 keys are arranged in the standard adding machine format, except that 0 is at the far lower left rather than at the bottom of the block. The physical arrangement of the A through Z keys in the alphabetic block is completely arbitrary. The

Figure 6. The SSR system signal conditioner circuit. Major sections are the power supply, AGC, high- and low-pass filters, and the Schmitt trigger.



arbitrariness does not slow alphabetic coding of behavior once the keyboard layout has been learned, but it does tend to discourage using the keyboard as a typewriter. The grammar keys are located according to frequency of use and a left to right organization of grammatical usage in the complex set of rules designed for encoding monkey social behavior noted above (AN APPLICATION section).

Tape Recorder

While any audio tape recorder with an adequate frequency response could be used to record keyboard output, nearly all of our field testing and experience has been with Uher 4400 Report quarter track stereo tape recorders. This model of Uher has a frequency response of 40-10,000 Hz at 1-7/8 in./second tape speed (-3 dB for 600-6,600 Hz). The stereo feature provides one track for keyboard output and another for observer voice notes. The quarter track feature increases efficiency in that both sides of a tape can be used. This means that there is less to carry in the field and less time rewinding tape during data collection and transcription.

The Uher 4400 Report with carrying case, shoulder straps, microphone, and NiCad batteries weighs about 12 lbs. Cassette tape recorders are much lighter, but we have not yet found a means for high-speed playback of cassette tapes during transcription.

Recording Tape

Keyboard output is packed onto a quarter track of 1/4-in. magnetic tape at an effective rate of 700 bits/in. After testing a variety of products, the 3M Scotch brand Type 212 (Catalogue Code No. 212-1/4-R45) is the best readily available type;

electronic errors due to dropouts are very rare, averaging less than one detected per hour of data transcribed.

Playback Tape Deck

A Sony tape deck (Model TC-850) is used for high-speed (15 ips) playback during transcription of keyboard encoded tapes. A less sophisticated 1/4-in. tape deck could probably be used, but we have found this particular model to be mechanically stable and easy on the data tapes while other manufacturers' models were not.

Signal Conditioner

When the SSR system encoding signal is played back for transcription at eight times recorded speed, it is passed through a signal conditioner circuit to resolve the partial degradation that the signal incurs in the course of recording and playback (see Level D in Figure 5). The conditioner consists of four basic sections, the first three of which are equipped with bypass switches. The first section is an automatic gain control (AGC). This has proved to be a great help in securing reliable operation of the transcription process by reducing the variation in amplitude of the signal. The second and third sections are high- and low-pass filters, which aid in removing noise and reducing certain kinds of wave form distortion that result from variations in the frequency response of the tape recording system. The fourth and final section is a Schmitt trigger, which converts the signal to a square wave that is TTL-compatible and suitable for computer input.

The signal conditioning circuit is diagrammed in Figure 6. Unlike the keyboard, the conditioner is a one-of-a-kind prototype. Persons wishing to

duplicate the conditioner should be cautioned that some parts of the circuit may be sensitive to component variation, and that some experimentation with components and component values may be necessary to secure good results. This is particularly true of the AGC section, since proper operation may depend on the characteristics of FET1.

AGC. Capacitor C102 blocks dc. R101 and FET1 form a voltage divider in which the ratio is controlled by the FET gate voltage. In normal operation, the FET source-to-drain voltage is about 20 mV peak to peak so that operation is essentially ohmic.

The voltage divider is followed by an amplifier consisting of operational amplifier OA1 and feedback network R104 and R103, which gives a gain of 200. The voltage divider and amplifier together form a variable-gain amplifier with an effective overall gain varying from 100 to .1 as the gate voltage on FET1 swings from -1.2 V to -.2 V. R102 serves to limit the maximum gain. C101 and C103 help to suppress oscillations that can otherwise occur when the input signal drops to zero and the AGC gain increases to its maximum.

When the AGC bypass switch SW1 is in the OFF position, the gain voltage is determined by R109 and gain is fixed at a value of about 1.5. When the AGC bypass is ON, the AGC is engaged and the gate voltage is derived from the output of OA1 by a rectifier and filter network consisting of XR1, C105, and associated components. A dc voltage of about one-third the peak-to-peak swing at OA1 is developed at C105.

Capacitor C104 blocks dc and permits the voltage developed at C105 to be referenced, not to ground, but to a dc offset level set by R108. Thus, the output at C105 is the sum of one-third the peak-to-peak output of OA1 plus the voltage at the wiper of R108. This voltage is about -1.6 V. It would probably be advantageous to insert a 22K resistor between the -18-V supply and the upper terminal of R108, to widen the adjustment range. The effect of R108 is to set the output level to be maintained by the AGC; R108 should be adjusted for an output of about 4 V peak-to-peak.

In operation, a rise in the output of OA1 causes a rise in the gate voltage of FET1 and thus a reduction in gain.

The AGC time constant is determined by C105. Since the SSR encoding signal drops to zero during the delay portion of the sweep, it is not feasible to have a time constant that is short compared with the sweep time (about 6 msec during playback for transcription at eight times recorded speed); if it were comparatively short, the AGC would increase gain during the delay portion of the sweep. A value of 1 μ F gives a time constant of about 50 msec and yields good results.

Although the AGC does not act fast enough to remove dropouts, it has an important stabilizing

effect. By fixing the level from which the dropouts drop, it permits the threshold adjustments of the Schmitt trigger to be permanently set at an optimum level relative to the signal. The combination of AGC and Schmitt trigger eliminates the effects of many mild dropouts.

The AGC also has an important effect in eliminating drifts in signal level. It has been found that recordings made on portable tape recorders operating on battery power often contain significant shifts in level as the recording progresses. This is probably due to changes in battery condition and tape tension. The shifts, amounting to perhaps 6 dB in some cases, might not be noticeable in audio recording, but they can have a serious effect on the proper transcription of an SSR system encoded record unless AGC is provided. Before AGC was added to the signal conditioner, some of the lower quality recordings could not be transcribed without unacceptably high error rates unless the gain was constantly monitored by the operator.

Filters. OA2 and associated components form a high-pass filter with a 12-dB/octave cutoff at a frequency of $\frac{1}{2}\pi RC$. For the values given, cutoff is at about 8 kHz. When the switch SW2 is in the OFF position, the RC network is bypassed and the section becomes a simple amplifier with a gain of about 1.7.

Similarly, OA3 and associated components form a low-pass filter, cutting off at about 16 kHz.

Theoretically, the optimum ratio for the feedback networks R203:R204 and R303:R304 would be 1:1.72 for optimum results when each filter is used by itself. When they are used together as a bandpass filter, however, reducing the ratio to 1:1.5 has a beneficial effect in leveling the response within the passband. As a practical matter, the difference is not of great importance.

The high-pass filter (OA2) has proved to be quite valuable in removing several kinds of noise, including ac hum and low-frequency components arising from partial rectification of the signal. The low-pass filter (OA3) has been less important.

The effect of the filters in improving the signal may be due, not entirely to the frequency response characteristics, but to their effect on modifying phase relationships within the signal. It is characteristic of the signal, as played back, to contain a high-frequency ringing at three times the base frequency. This probably results from a harmonic response to some component of the base signal and is not important in itself. However, the phase relationship between this component and the base frequency is quite important in determining the time of zero crossings, and the most favorable relationship appears to be one in which the slope of the zero crossings is highest.

Although the described filters work extremely well with our particular combination of equipment, it is possible that experimental adjustment of the R and

C values in the filters may be necessary with other equipment. If the critical effect of the filters is a phase rather than a frequency response adjustment, it is possible that some other filter design may be more appropriate.

Schmitt trigger. OA4 and associated components form a Schmitt trigger circuit. All other sections of the circuit use a Type 741 operational amplifier. In the trigger circuit, a Type 709 is necessary to attain a high switching speed. Because it is used in a switching mode, networks usually necessary to stabilize the amplifier are not needed.

In essence, the Schmitt trigger is used to detect zero crossings. In our particular application to the SSR system encoding signal, however, it is desirable to offset the trigger levels from zero and to build in some degree of hysteresis. A controlled amount of hysteresis is beneficial because the signal contains a certain amount of high-frequency noise that could otherwise produce multiple crossings as the signal crosses the threshold level. It also prevents small amounts of ringing from causing multiple triggering.

Control R407 determines the average offset of the two trigger levels from zero, and R405 determines the distance between them. The optimum setting with our unit proved to be at the extreme adjustment ranges of the potentiometers, resulting in switching levels of +.5 V and +1.5 V (i.e., an offset of 1 V and a hysteresis of 1 V). It would probably be beneficial to expand the adjustment range by increasing R405 to 1K and reducing R403 to 220K.

An offset is necessary to insure that the delay portion of the sweep produces a constant ON output from the trigger. The delay portion of the sweep will not always be exactly at 0 V, because for various reasons the data portion often contains some dc component, even though none is intended. In fact, the delay portion, when examined carefully, may show an exponential decay to 0: If the triggering level is set close to 0, there may be a crossing during the delay portion. A positive offset insures that the delay portion will yield a solid ON, and also insures that the Schmitt trigger always finishes in the ON state when the signal is removed. These considerations are important for the decoding logic in the first stage of transcription (see SOFTWARE).

Variable Timing Reference

To synchronize the logic in the first stage of transcription with the stream of values coming from the signal conditioner during playback, the interval between expected zero crossings is calculated during the delay portion of each sweep. The interval can be adjusted both before and during playback by means of a potentiometer. The potentiometer (knurled knob with a dial scale) adjusts the voltage to an analog input channel of the computer over a range of -4.096 V to +4.096 V. A FORTRAN call to a system

subroutine returns an integer value in the range -2048 to +2047. In one case, in which a user's battery powered tape recorder had gradually slowed down in the course of recording a record, the record was recovered by monitoring the electronic errors (in this case, mainly due to the resulting gradual shift in the time base) through the BOZO system loudspeaker during the first stage of transcription (see SOFTWARE section) and adjusting the value of the potentiometer to keep them to a minimum.

Computer

A Datacraft 6024/5 computer with 16K words of core memory was used for all of the transcription and data processing described in this report. The 6024/5 has a 24-bit word and a cycle time of 1 μ sec. Peripherals include a 5-megabyte random-access disk memory, Teletype and CRT display, and a drive for seven-track computer tape.

Oscilloscope

Occasional access to an oscilloscope is necessary in order to check keyboard switches and output wave forms, and to troubleshoot malfunctions of both the keyboard and the tape recorder. For field studies in which the SSR system keyboard is a primary tool, access to a portable field model oscilloscope is highly recommended. If an oscilloscope is not available, sample records should be sent from the field to the laboratory for transcription at regular intervals throughout the field period.

SOFTWARE

The functional aspects of the basic SSR system software are described below. Program SSR performs the first stage of transcription; it decodes the keyboard encoded data tape and stores the characters and their sweep counts in binary form in a logical file on the disk. Program SSRA performs the second stage of transcription; it converts the string of characters from binary form to character form and organizes them into a timed manuscript of lines. An SSR system utility program called SPLICE joins sequential portions of an interrupted record, each of which has its own relative time base counting from the first character in the portion, into one record with a time base that is continuous and consistent across all of the portions joined. Another utility program, called REFRAM, corrects the timing of a record in which the sweep rate is not equal to exactly 20/second. COMBO, the last program discussed here, is included as an example of third stage software, the stage at which programs are generally tailored to a specific user's problems. Program COMBO produces the printout in Figure 1. All of these programs except the first are written in Datacraft's FORTRAN IV. The first, SSR, has a

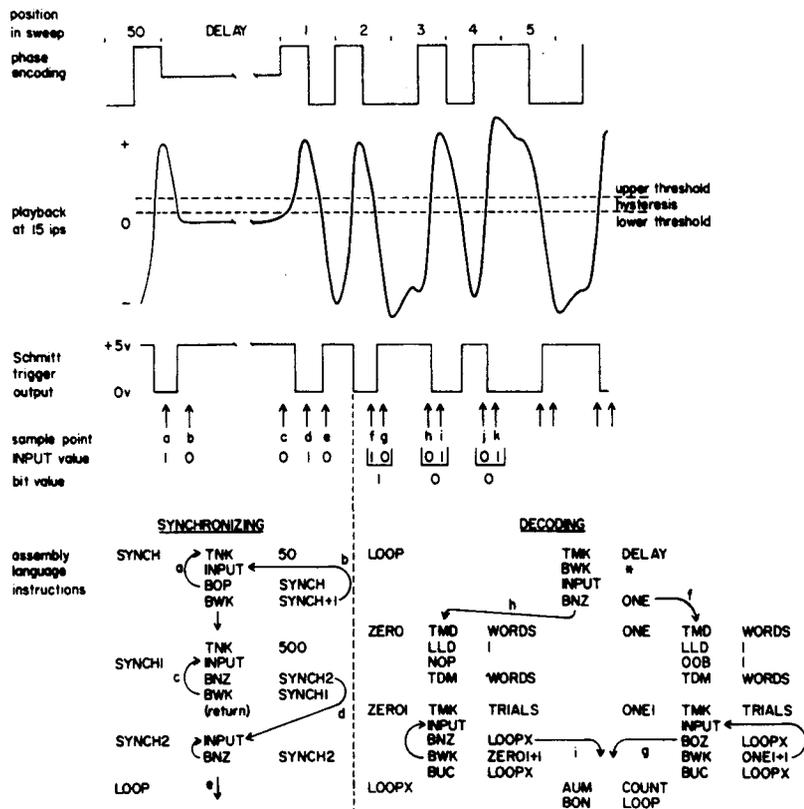


Figure 7. Relationship between the output of the Schmitt trigger and the logic of the assembly language subprogram. In the top half, output of the trigger is related to output of the playback deck. The assembly language instructions are grouped into functional series at the bottom half. Sample points relate specific aspects of the trigger output to specific series of instructions. The left half of the figure illustrates the method of synchronizing logic with output. The right half of the figure illustrates the decoding process. See text.

basic framework in FORTRAN IV, but its main functions are programmed in assembly language. Listings of these programs are available without cost upon request.

Program SSR—First Stage of Transcription

Upon call from the disk, Program SSR advises the operator to set the variable timing reference to the proper value for calculating the delay between expected zero crossings in the encoding signal of an SSR system record played back at 15 ips. It also requests that a title for the record be entered via the Teletype. With entry of the title, the program is ready to decode the output of the signal conditioning circuit.

Decoding is performed by an assembly language subprogram. The first task is to synchronize the logic of the subprogram with the stream of values coming from the Schmitt trigger portion of the signal conditioner. Once synchronized, the succeeding values in the data portion of the sweep are sampled, one after another, and decoded, pair by succeeding pair in sequence, into bit values that represent the states of the keys. The bit values are stored, bit by bit, in a double word of memory in the same order in which the states of the keys had been phase encoded and multiplexed into the keyboard encoding signal (see Level B in Figure 5). These two system-specific parts of the transcription process, i.e., synchronizing subprogram logic with trigger output and decoding

the states of the keys encoded in the data portion of the sweep, are diagrammed in Figure 7.

Before considering the methods of synching and decoding in detail, recall that the keyboard encoding signal is partially degraded in the course of recording and playback (see Level D in Figure 5). The primary function of the Schmitt trigger in the signal conditioner circuit is to reshape the output of the playback deck into binary information that can be evaluated by the subprogram. The trigger is so devised that when the playback signal drops below the trigger's lower threshold, the output of the trigger is ON at +5 V. When the playback signal rises above the upper threshold, output of the trigger is OFF at 0 V. When the playback signal is in between the upper and lower thresholds, output of the trigger is ON or OFF, whichever was last. This result is a function of the hysteresis that has been designed into the trigger circuit (see HARDWARE section, signal conditioner). The relationship between the output of the playback deck and output of the trigger is illustrated in the upper part of Figure 7.

The relationship between the output of the trigger and the assembly language instructions that synchronize subprogram logic with trigger output is illustrated on the left half of Figure 7. Beginning at the program address SYNCH, the integer -50 is transferred to the K register of the computer (TNK: Transfer Negative operand to K; here the operand is

50). INPUT is a subprogram macroinstruction that utilizes BOZO (Botany-Zoology Computer Facility) system functions resident in core to yield a zero when the output of the trigger is ON and a positive number (represented as a 1 in Figure 7) when the output is OFF. When INPUT yields a positive number (as it would at Point a in Figure 7), the subprogram branches back to address SYNCH (BOP stands for Branch On Positive). When INPUT yields a zero, the subprogram advances to the BWK instruction. BWK is a mnemonic for Branch When K is not equal to zero. This instruction increments the contents of the K register by one and then tests whether it is equal to zero. When it is not zero, the subprogram branches to the address in the operand field, here to SYNCH+1, which puts it back at the INPUT macroinstruction. If the subprogram were to encounter a series of zeros from successive executions of INPUT, it would cycle through the instructions INPUT, BOP, and BWK, each time incrementing and testing the value in the K register. When K finally does equal zero (here after 50 cycles through the series of instructions), the subprogram moves to the next instruction in sequence, here TNK 500. Should INPUT yield a positive value in the cycling before K has been incremented up to zero, the subprogram would Branch On Positive to SYNCH, where a -50 would be transferred to register K and, in our usage here, the search would begin again. INPUT requires 17 μ sec and the other instructions in this series require 1 μ sec each. Thus, the first step of synchronizing subprogram logic with trigger output proceeds by sampling the output of the trigger every 19 μ sec until the output is found ON (i.e., INPUT yields a value of zero) for 50 consecutive samples. This would occur only if the subprogram were sampling in the delay portion of a sweep or if the record had ended. The trigger is ON during the delay portion of a sweep because the second half of the wave form representing the last position in the data portion of the sweep drops from positive to zero (recall that the 50th stage of the ring counter is hardwired as an OFF). For the same reason, the trigger is also ON at the end of a record. A record is ended by switching the DATA rocker from SEND to STBY, which uncouples the multivibrator from the ring counter at the 51st stage, thereby insuring that the last sweep is completed through the 50th stage before the record ends.

The second step in synching logic and output distinguishes between the delay portion of a sweep and the end of a record by the number of consecutive zeros that a series of successive executions of INPUT yields after the 50 in the first step. The K register is incremented from -500 toward zero each time the subprogram, beginning at address SYNCH1, cycles through the series of instructions, INPUT, BNZ (Branch when Not Zero), BWK. As above, the series requires 19 μ sec. In playback at eight times recorded

speed, the delay portion of each sweep lasts about 1.563 msec ($12.5 \div 8$). The 50 consecutive zeros in the first step of synching required about .950 msec; hence, if the sample count in the second step exceeds 32, the record has probably ended. To pass over extended delays due to tape splices or long periods of signal loss due to large dropouts, another 468 consecutive zeros are required before the record is taken as terminated. (The total of 500 required is arbitrary and can be made closer to 32). INPUT will continue to yield zeros until the sampling encounters the Schmitt trigger's response to the first half of the wave form representing the first stage of the ring counter, which is hardwired in the ON stage. At that point (Point d in Figure 7), the signal rises from zero to positive, trigger output drops from +5 V to 0, INPUT yields a positive value, and the subprogram branches to address SYNCH2.

The time requirements of the sets of assembly language instructions from SYNCH2 on are specifically tailored to decode the output of the Schmitt trigger as it follows the output of the playback deck operating at 15 ips. The two instructions at SYNCH2 require only 18 μ sec, less than a quarter of the 93- μ sec interval (.75 msec \div 8) required for playback of the wave form representing the state of a key. By sampling within a quarter of an interval after the Branch when Not Zero instruction at Point d in Figure 7, the point at which the logic of the decoding loop is precisely synchronized with output of the trigger is located in the third quarter of the interval that corresponds to the first position in the data portion of the sweep. From this point on (Point e in Figure 7), synchrony is maintained by the time requirements of the instructions in the LOOP section of the subprogram.

The relationship between the output of the trigger and the assembly language instructions that decode it is illustrated on the right half of Figure 7. The time required for execution of the series of four instructions beginning at LOOP is controlled by the value of DELAY, which is computed from the value of the variable timing reference at the beginning of each sweep. Optimally, the INPUT macroinstruction samples trigger output in the second quarter of the interval corresponding to a sweep position, or, in terms of wave form, just before the zero crossing transient. For example, the wave form at the second position in the sweep would be sampled for the first time at about Point f in Figure 7. The optimal setting for the DELAY would position the Sample Point f very close to 3/4 of an interval after Sample Point e. In this example, execution of INPUT at Point f would yield a positive value and the subprogram would Branch (when Not Zero) to Program Address ONE.

The series of instructions beginning at ONE stores the value from INPUT. TMD Transfers the contents of a double word at Memory location WORDS to the

D register. LLD performs a Logical shift of the contents of the D register one position to the Left. OOB performs an exclusive OR on the Operand (here 1) and the rightmost Byte of the D register. TDM Transfers the contents of the D register back to Memory locations WORDS. TMK Transfers a value (-10) stored at Memory location TRIALS to the K register. Then another INPUT macroinstruction is executed. While INPUT requires a total of 17 μ sec, the sample is actually taken in the 11th μ sec. Thus, the time required to execute the series of instructions from TMD at Address ONE to the actual sample by INPUT immediately after Address ONE1 is 25 μ sec, or about one quarter of an interval after the previous sample. In terms of the wave form, this point (Point g in Figure 7) is just after the zero crossing transient, hence INPUT yields a value opposite the value it yielded last. In this example, after the positive value from the sample at Point f, INPUT yields a zero at g, and the subprogram Branches On Zero to LOOPX. If Point g had been early, before the zero crossing transient, INPUT would have yielded a positive value and the subprogram would have advanced to BWK, incremented Register k and branched to Program Address ONE1+1, which would have put it back at the INPUT macroinstruction. The second sample would have been 19 μ sec after the first, and unless there had been an electronic error, this would have occurred after the zero crossing, which would have then produced a zero value at INPUT and a branch to LOOPX. If this series of instructions does not yield a zero on the second sample after Address ONE1, the subprogram will continue to cycle through the series until INPUT does yield a zero or the contents of Register k has been incremented from -10 up to 0, at which point the subprogram will Branch Unconditionally (BUC) to LOOPX. If more than two samples are required, however, subprogram logic will probably be out of phase with trigger output through the rest of the sweep and the results will be detected as erroneous in subsequent parts of the first stage of transcription.

At LOOPX, the AUM instruction Aids Unity to Memory location COUNT. The contents of COUNT had been set to -48 at the beginning of the subprogram and the subsequent Branch On Negative instruction passes control to LOOP until 48 wave forms with zero crossing transients have been decoded. At LOOP, the value of DELAY again positions the sample point 3/4 of an interval after the previous sample. Here, at Point h in Figure 7, INPUT would yield a zero and the subprogram would advance to Program Address ZERO. The series of instructions from Program Address ZERO through BWK ZERO1+1 is the complement of the series from Program Address ONE reviewed above.

In summary, the sampling schedule of 3/4 interval plus 1/4 interval is offset from the boundaries of the

actual period corresponding to the wave forms that encode the states of the keys such that the first sample of a pair of samples occurs just before the zero crossing transient in the wave form and the second sample occurs just after. Pair by pair in sequence, the samples of trigger output are thereby decoded into the states of the keys that were originally phase encoded and multiplexed in that order in the sweep.

The other aspects of the first stage of transcription by program SSR are less system specific and are only summarized here. After a sweep has been decoded, the contents of the two double words representing the current and the immediately preceding sweeps are compared. If they are the same, there is simply a return to the main program, where the sweep count is incremented and the subprogram is called again to process the next sweep. If the contents of the two double words are not the same, the position of the bit that differs is found and transcribed into the character assigned to the key that was sampled at that position in the sweep. Return to the main program then results in the character being added to the string of characters that comprises the decoded record up to this point in the playback. Again, the sweep count is incremented and the subprogram called to process the next sweep.

When a character is added to the string, it is stored with its sweep count so that the relative time of each character is preserved. One byte of memory is required to store a character and three bytes are required to store its sweep count. Our present computer has three 8-bit bytes per word of memory. BOZO system functions (in Datacraft FORTRAN IV) are used to extract, rotate, and replace bytes so that the four bytes of information/character can be stored together in a large buffer of three-byte words. The current buffer is limited to 8,000 words of memory, but as 24,000 bytes, it can store 6,000 characters before transcription of a record will be automatically terminated by the SSR program. Even with this much storage, we occasionally exceed capacity and must later use program SPLICE (see below) to concatenate the separate parts into one continuous record.

In the course of decoding and character processing, several kinds of errors are detected by program SSR. Two criteria determine the detection of errors. The first is that a key must remain in one state (ON or OFF) for at least two sweeps. This is a reasonable criterion because humans cannot push the keys faster than about 10 times/second; changes in state for less than .1 sec (or two sweeps) are probably due to electronic errors on the tape. The second criterion is that there can be only one character encoded per sweep. This is a necessary criterion because the order of the character determines its interpretation by grammatical rules at subsequent stages of data processing. If more than one bit position is different when the two words are compared, return to the main

program results in the insertion of an error marker (a back arrow, which is also defined as a line starter) into the record at the current end of the character string. All other kinds of errors at the first stage of transcription, i.e., an incorrect number of zero crossing transients or loss of synchrony between subprogram logic and trigger output due to electronic errors on the tape are also detected as the case of more than one character in the sweep and are processed accordingly.

After the error marker is inserted into the record, a "tock" sound is generated over the BOZO Facility loudspeaker to inform the operator that any error has been detected. The operator can use the "tock" rate to evaluate the effects of manipulating the Automatic Gain Control, high- and low-pass filters, and/or variable timing reference when attempting to recover a nonstandard record, such as in the case where the user's tape recorder had gradually slowed down in the course of an observation recording session. In contrast to the "tock," a "tick" is generated whenever a character is successfully transcribed. This informs the operator that the first stage of transcription is proceeding normally.

The first stage of transcription is terminated when program SSR detects a loss of signal (a series of at least 500 consecutive executions in which INPUT yields a zero) or the operator indicates termination manually by pushing a button on a reference panel at the computer. Program SSR then reads the string of characters with sweep counts from the buffer in core memory and writes them in binary form onto a logical file of the disk. Lastly, program SSR chains to program SSRA, which will perform the second stage of transcription.

Program SSRA—Second Stage of Transcription

Program SSRA converts the string of characters produced by the first stage of transcription into a timed manuscript of lines. The program begins by asking the operator to designate via the Teletype the physical device on which the manuscript is to be displayed, e.g., screen (CRT), printer or none. It then reads the string of characters with their sweep counts from the logical file on the disk where they had been written by program SSR into an 8,000-word buffer in core.

It may seem unnecessary to store the character string on the disk between the first and second stages of transcription, but doing so provides two options that are quite useful. First, the timing in the first stage of transcription is rigidly determined by a condition external to the program, i.e., the necessity of keeping subprogram logic in synchrony with trigger output. The timing in the second stage of transcription is simply an internal matter of reading from the logical file when ready. In the first stage of transcription, the record accumulates in core until it ends or until the buffer is full, after which it is written

out onto the disk. Reading and writing in binary form is very rapid and only a few seconds are required to transfer a record from core to disk. Thus, if 30 sec of recording time are left blank (in DATA STBY) between records in the field, a data tape with a series of records can be processed nonstop through the first stage of transcription, with a substantial savings in operator and computer time. Secondly, program SSRA writes the timed manuscript of lines into a second logical file on the disk. Thus, the operator has access to both the binary character string and the timed manuscript forms of the record and can save either or both on computer tape for subsequent data processing.

After the character string has been read from the disk into the buffer in core, it is scanned for the first time statement (!Tnnnn, where n is a numeric character 0 through 9). Since the sweep count represents the number of 20ths of a second between the first character and any subsequent character in the record, the difference between the sweep count of the ! that starts the time statement and the sweep count of any other line starter character can be used to compute the real time of any line in the manuscript by reference to the hour and minute markers that had been entered after the !T. Once the time statement is found, and the numeric characters are converted to integers, the program returns to the beginning of the buffer. Using a BOZO system function to extract bytes from the buffer, the characters are retrieved, one by one, together with their sweep counts. Each character is compared to the list of characters that have been defined as line starters in a data statement of program SSRA. One by one, the characters are organized into a line until the current character matches one of the defined line starters. At that point, the sweep count of the previous line starter, which started the current line, is used to calculate the real time equivalent of the line. The time and the line are then written out onto the chosen physical device. After the whole manuscript has been displayed, a second pass through the buffer extracts and organizes the characters, calculates the real time, and writes the timed lines of the manuscript out onto a second logical file of the disk.

If a completely correct time statement is not found when the character string is scanned, the SSRA program asks whether the operator wants to enter a time statement after viewing the manuscript on the screen. If the operator enters NO, the manuscript is simply organized and written onto the chosen device and then into the logical file as described above, but without the real-time equivalents of the line frame counts. If the operator enters YES, the program will organize the manuscript and write it without the real-time equivalents, but only on the screen. Upon reaching the end of the character string in the buffer, the program asks the operator to enter the frame number and the four digits representing the time. The

program then proceeds as described above, i.e., as if it had found a complete time statement on the first scan through the character string. At the completion of this second pass through the character string, the manuscript of timed lines is ready for storage on computer tape or for processing by third-stage software that is specifically tailored to the user's particular interests.

Program SPLICE

Program SPLICE is an SSR system utility program that is used to join parts of an interrupted record together. The stream of data in a record can be interrupted by large dropouts due to momentary electronic malfunctions of the tape recorder or playback deck or to uneven density of the magnetic recording material on the tape, by accidental switching of the keyboard or recorder controls, or by exceeding the capacity of the buffer with too many characters. To splice the parts separated by such interruptions, the user need only determine the number of missing sweeps. When the interruption is due to a dropout or to user error, the time between the end of the preceding part and the beginning of the succeeding part can be estimated or measured with a stopwatch during playback at recording speed (1-7/8 ips). When due to too many characters, the tape can be rewound over the point at which storage capacity was exceeded and a second record can be transcribed so that partial printouts of the end of the preceding and the beginning of the succeeding parts can be overlapped and matched. After the match is made, the duplicate lines in each record are deleted. The parts to be joined are then copied into a logical file in their proper order.

Program SPLICE scans the lines in the logical file until it encounters a read error. The format for a line in the record is very different from the format for the title that is inserted at the beginning of every record during the first stage of transcription. At the read error, the program backspaces over the title, writes a / (which is a comment line starter in our usage, hence ignored by our data processing programs) and a comment in standard line format over the title location in the logical file and over the next line in the file, which is the column heading line (see Figure 1). It then asks the operator to enter the number of missing sweeps on the Teletype. The number is added to the last frame count found before the read error and the sum is assigned to a constant. The program resumes scanning lines from the file. If the frame count of the current line is less than the frame count of the previous line (as it will be for all lines after the first part in the file), the constant is added to the current frame count, the file is backspaced, and the line with its new frame count is written in the logical file over the place of the line and its old frame count. SPLICE has been used only five times in some 550 h of data.

Program REFRAM

Program REFRAM is an SSR system program that is used to correct the frame counts of lines in a record in which the sweep rate of the encoding keyboard was other than exactly (to six significant digits) 20/second. Timing errors are usually due to inaccurate settings of the potentiometers governing the output of the multivibrator. Such errors are essentially constant throughout a record. To take account of the possibility of timing errors, standard operating procedure for use of the keyboard includes entering a time statement at both the beginning and the end of every record. Time statements can be entered with high accuracy by pressing the ! as the sweep second hand crosses a prescribed numeral on a timepiece, and then entering the T and the numbers for hour and minute as usual.

Program REFRAM scans the record for the first and last time statements. If it finds at least two time statements, it computes the number of minutes between them and then the number of sweeps there should have been. The ratio of the number there should have been over the difference between the frame count of the first time statement and the frame count of the last is the reciprocal of the rate at which the keyboard was operating. This ratio is used to correct the frame counts of all lines in the record. The logical file is rewound and after each line is read, the file is backspaced, the product of the ratio times the frame count is formed, the real-time equivalent is formed, and the line is rewritten into the file with a corrected real time and with the product as its revised frame count.

Program COMBO

Program COMBO is a third-stage data processing program that organizes the raw records of the primary and secondary observers from the priority of access to water test described above (see AN APPLICATION) into the output depicted in Figure 1. Each observer's record is copied into a logical file on the disk. The program scans each record for its first and last time statements. If at least two time statements are found in each record, the program is ready to reframe them so that they can be put into registration.

Reframing within each record is done in the manner described above for program REFRAM, except that after the ratio between the expected sweep count and observed sweep count is computed, the product of the ratio times the frame count of the first time statement is subtracted from the integer 10,000 and the difference is assigned to a constant. The logical file into which the record was placed is then rewound and, as each line is read, the frame count is multiplied by the ratio and the product is added to the constant so that the relative time of each line is corrected and put into the register on the first time statement, which has been arbitrarily assigned the frame count of 10,000. At the end of file, the program

switches to the second record and operates upon it in like manner, so that the relative time of each of its lines is corrected and in register on the first time statement, which again has a frame count of 10,000. Since the ! of the first time statement in each record was entered by one observer pushing the ! key on each keyboard simultaneously, the two records are now in registration with each other.

To output the two records in registration, the logical files are rewind, and, reading a line from each record as required, the frame counts of the current lines from each record are compared. If they are within .5 sec (10 sweeps) of one another, the lines are printed together with the real-time equivalent of the frame count from the primary observer's record. The next line in each record is then read and the frame counts compared. If the frame counts are more than .5 sec different, the line with the lower frame count is printed with its real-time equivalent, and the next line in that record is read. Thus, the data from the two records are output together in the temporal relationship in which they were originally recorded.

This technique of combining records in temporal registration is also being applied to repeated playbacks of cine filmed and video taped sequences of behavior among monkeys and children. Each individual in the visual record is followed as a focus subject and its behavior is encoded on an SSR system keyboard. The visual record is then rewind and the next individual is followed as focus subject. To insure a constant time base for the multiple playbacks, the markers at the beginning and the end of the visual record are entered on each individual's keyboard encoded record. To verify the time base of the keyboard, time statements are also entered at the beginning and the end of the keyboard record. The only time error that cannot be controlled is that due to variation in the user's reaction time. After all individuals have been followed, their records can be corrected and put into registration for parallel printing, evaluation, and analysis.

SYSTEM ADAPTABILITY

As has been indicated throughout this report, the logic of the SSR system is readily adaptable to a wide variety of user problems because most of its features are arbitrary and software defined. The set of characters, the functions of characters, even the number of characters, can be changed. Of course, to change the number of characters would require corresponding changes in the number of stages on the ring counter and in the values of some of the variables that are used for counting in the first stage of transcription. All of the other changes that we have imagined can be done by modifying only the software. For example, to change the time base for proper timing of records entered from cine films projected at other than normal speed, the number of sweeps that equals a second can simply be redefined. Even at the more basic level of different word lengths in different

computer memories, fitting the 48 bits of each sweep into, for example, three words of 16-bit length instead of two words of 24-bit length is fairly straightforward. Once implemented, this capacity for easy software modification of its basic characteristics allows the SSR system to be readily adapted to meet user's changing needs.

REFERENCE NOTES

1. Sykes, R. E. Midcars: Minnesota interaction data coding and reduction system. *Observations*, 1971, 1, 1-8. Available from Observational Research, University of Minnesota, 2122 Riverside Avenue, Minneapolis, Minnesota 55455.
2. Sackett, G. P., Stephenson, E., & Ruppenthal, G. C. *Portable digital data acquisition systems for observing primate behavior in laboratory and field settings*. Paper presented at IVth International Congress of Primatology, Portland, Oregon.
3. Stephenson, G. R., Smith, D. P., & Roberts, T. W. *The SSR recorder: An event recording system with computerized transcription*. Abstract for AAAS oral presentation in American Zoologist, 1970, 10, 483-484.

REFERENCES

- ALEXANDER, B. K., & HARLOW, H. F. Social behavior of juvenile rhesus monkeys subjected to different rearing conditions during the first six months of life. *Zoologische Jahrbuch der Physiologie*, 1965, 71, 489-508.
- BOELKINS, R. C. Determination of dominance hierarchies in monkeys. *Psychonomic Science*, 1967, 7, 317-318.
- BERNSTEIN, I. R. Primate status hierarchies. In L. A. Rosenblum (Ed.), *Primate behavior: Developments in field and laboratory research* (Vol. 1). New York: Academic Press, 1970. Pp. 71-109.
- CLARK, D. L., & DILLON, J. E. Evaluation of the water incentive method of social dominance measurement in primates. *Folia Primatologica*, 1973, 19, 293-311.
- DAWKINS, R. A cheap method of recording behavioural events for direct computer-access. *Behaviour*, 1971, XL, 162-173.
- KAWAI, M. On the rank system in a natural troop of Japanese monkey. I: The basic and dependent rank. *Primates*, 1958, 1, 111-113.
- MISSAKIAN, E. A. Genealogical and cross-genealogical dominance relations in a group of free-ranging rhesus monkeys (*Macaca mulatta*) on Cayo Santiago. *Primates*, 1972, 13, 169-180.
- STEPHENSON, G. R. Cultural acquisition of a specified learned response among rhesus monkeys. In D. Stark, R. Schneider, & H. J. Kuhn (Eds.), *Progress in primatology*. Stuttgart: Fischer, 1967. Pp. 279-288.
- STEPHENSON, G. R. Testing for group specific communication patterns in Japanese macaques. In E. W. Menzel (Ed.), *Symposium of the IVth International Congress of Primatology* (Vol. 1) *Precultural primate behavior*. Basel: Karger, 1973. Pp. 51-75.
- STEPHENSON, G. R. Social structure of mating activity in Japanese macaques. In S. Kondo, M. Kawai, A. Ehara, & S. Kawamura (Eds.), *Symposia of the Fifth Congress of the International Primate Center*. Tokyo: Japan Science Press, 1975. Pp. 63-115.
- VARLEY, M., & SYMMES, D. The hierarchy of dominance in a group of macaques. *Behaviour*, 1966, XXVII, 54-75.
- WEISBARD, C. The effect of parturition and group composition on competitive drinking order in stump-tail macaques (*Macaca arctoides*). *Primates*, 1975, in press.
- WHITE, R. E. C. Wrats: A computer compatible system for automatically recording and transcribing behavioural data. *Behaviour*, 1971, XL, 135-161.