

# A TIME-DIVISION MULTIPLEXING SYSTEM for ASYNCHRONOUS HIGH-RATE TELEMETRY

R. L. Appel and C. T. Pardoe

*A time-division multiplexing system has been developed that is significantly different from commercially available equipments. Multiplexing is achieved by interleaving blocks of data from each source preceded by a short coded bit pattern specifying the source of that particular block of data. Data blocks are transmitted as they become available from each particular source. Hence, synchronism of the input and output rates is not required. The prototype system can accommodate input rates as high as 461 kbps on a single input channel and output rates as high as 953 kbps. The system makes possible use of an excess of 90% of channel capacity for information transmittal.*

## Introduction

ONE OF THE PROBLEM AREAS encountered in adapting the Manned Space Flight Network to support the telemetry requirements for the Earth Resources Technology Satellite (ERTS) Program is that from one to three bit streams at rates from 1 to 32 kilobits per second (kbps) must be relayed from the Corpus Christi tracking station to the Goddard Space Flight Center for

processing in approximately real time. It is possible that a general-purpose digital computer such as the Univac 642-B, currently used at remote sites, could perform this task, but little processing time would remain for doing any other computations in the computer.

Multiplexers, per se, are neither new nor hard to obtain. However, after a cursory investigation of available equipments, it appeared that none precisely met the requirements of the ERTS problem. Several manufacturers were agreeable to making modifications to their standard line equipment, but these efforts would have required considerable expenditures of both time and money.

---

The development of the Time-Division Multiplexer was sponsored by the National Aeronautics and Space Administration. The funding for the work was provided by NASA through APL's contract with the Navy.

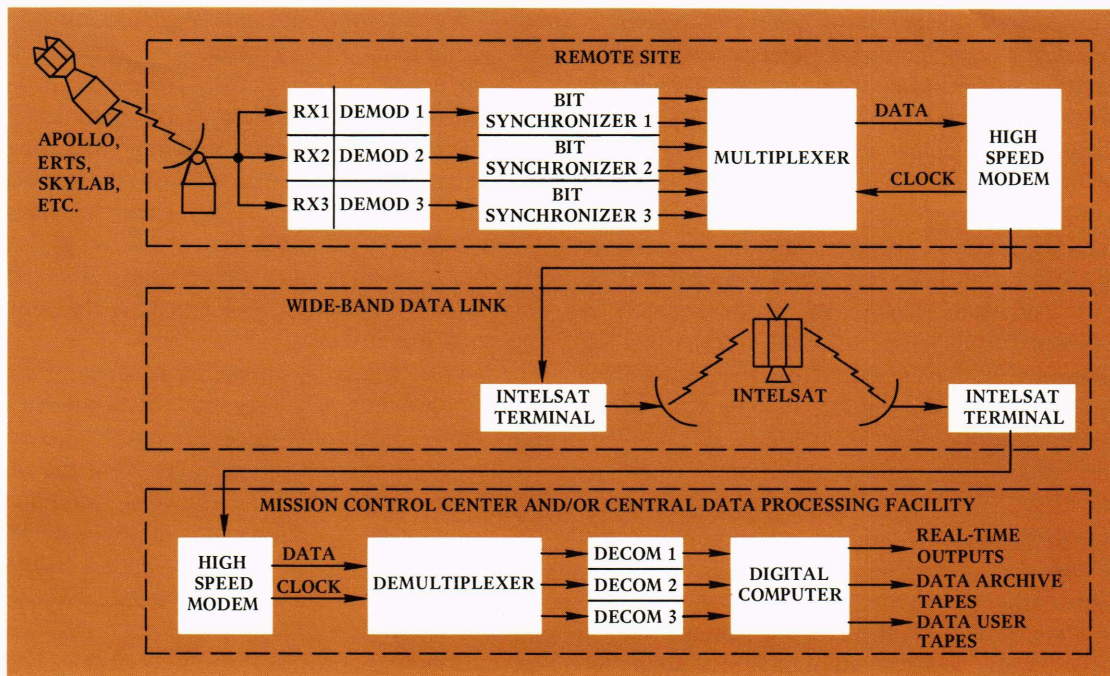


Fig. 1—Proposed application of APL time-division multiplexing system.

At the request of the Manned Flight Planning and Analysis Division of the Goddard Space Flight Center, it was agreed that the Applied Physics Laboratory would design and construct a prototype multiplexing system to satisfy the requirements for ERTS telemetry. This task was undertaken to demonstrate that a small, relatively inexpensive device could be used for the proposed application.

Further impetus was given to the project by the requirements for a similar multiplexing system in a wideband data link/central data processing facility concept (see Fig. 1) presently under consideration. In this application, it is necessary to accommodate a greater number of input channels and a higher output rate than for the ERTS program. The concepts used for the ERTS multiplexing system should, however, be easily expandable to satisfy these requirements.

### System Design Considerations

**Bit-Stuffing Techniques**—All of the commercially available equipments studied in the early stages of this project were of the so-called “bit-stuffing” type. These systems were intended to be used for multiplexing a large number (typically 20) of relatively low rate (typically 150 bps) sources. Furthermore, the sources frequently had

to be synchronous. To accommodate input rates greater than the specified input limitation, a sufficient number of inputs were connected together until the required rate total was achieved.

Multiplexing by bit-stuffing, in its simplest form, is no more or less, than commutation. The input signals are sampled at the output rate in an established sequence. If all sources are exactly synchronous with each other and with the multiplexer output rate, the system is highly efficient in regard to the amount of hardware required. However, when this synchronism is not a characteristic of the input signals (which is typical for space communications), considerable effort must be taken to avoid the ambiguity that occurs when an input signal changes state while it is being handled. This problem is compounded if the input rates are subject to reasonably large variations around their nominal value (this variation is typically as much as  $\pm 10\%$  of the nominal rate for space telemetry) and especially if more than two signals must be multiplexed, one of which is greater than one-half the output rate.

**Block Multiplexing Techniques**—The underlying concept of the APL multiplexer system is that data will enter the device and be stored as a block. At some later time this block will have an identifying label appended and the data will be

transmitted at a faster rate than it was received.

*Simplest Form.* Numerous techniques for implementing this basic concept are possible. If the nominal input rates are integer multiples of the lowest rate, one might consider a scheme using twice the integer number of storage registers for each input. These storage registers would then be emptied in a cyclic manner, half of the storage registers from each source being emptied during each output cycle. The length of the registers is determined by allowing sufficient storage for the maximum variation expected on the input rate. Information as to the number of bits of information in each block must be included as a label.

This system is attractive if the input rates are equal, since, then, only two storage registers for each source are required. Also, decommutation is relatively easy since the sources are interlaced synchronously. However, if the input rates differ by a large factor, or if they are not an integer multiple of the slowest rate, such a system becomes very cumbersome.

*Storage Register Pairs.* Another approach might be to use pairs of storage registers for each input source. By counting modulo the storage register length, one could recognize that a register for a given source had filled. This storage register would then be emptied as output while the incoming information was being stored in the other storage register of the pair.

If no input storage register was filled, then output information would be provided by an auxiliary source. This source might contain a synchroniza-

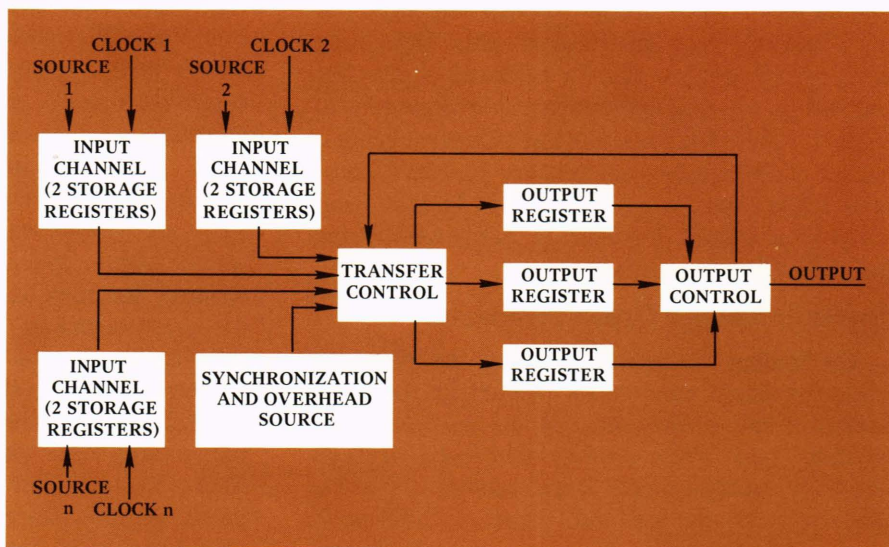
tion word (or words), message routing information, or low priority data (e.g., station status, dump telemetry, etc.).

The output bit stream would then consist of blocks of data (with an identifying label appended to each block) from the various sources, but the blocks would not necessarily occur in any specified order. Decommutation for such a system becomes somewhat more difficult since the label bits must now be examined to ascertain the source of a particular block.

Such a system is appealing because of its simplicity and modularity. It could be expanded easily by adding additional pairs of storage registers and a minimum of control circuitry. Also, the rate restrictions imposed in the previous technique and in elementary bit-stuffing are removed since data blocks do not appear as output until a fixed number of bits has been accumulated. However, it does have a major inadequacy; no input rate may be greater than one-half the output rate. If this is exceeded, data will be lost by attempting to fill and empty a storage register at the same time.

*Storage Register Pairs Plus Output Registers.* The input rate restriction can be removed if additional storage registers are used and if a high rate internal clock is provided to transfer the data from a filled input register to an output register (see Fig. 2). If these additional registers are filled and emptied in a cyclic manner, the output data will be in proper time sequence. Providing that a filled input register from the highest input rate source is given transfer priority over all other sources, and

Fig. 2—Improved storage register pairs system.



not allowing for the periodic insertion of synchronization information, three additional registers are sufficient for this purpose. If synchronization information must be periodically included (e.g., if every 20th block must contain the synchronization word), then the number of output registers required is four regardless of the number of input sources.

By using the high-rate internal transfer clock, the restriction of the previous system (that no input rate exceed one-half the output rate) is now improved to the point that no input rate exceed approximately one-half the transfer rate. Since this rate can easily be on the order of 1 to 10 MHz, this restriction poses no real limitation for telemetry data.

**Formatting and Channel Efficiency**—Up to this point, efficiency has been discussed only from the aspect of the amount and complexity of hardware required to perform a given multiplexing and demultiplexing task. Another, and perhaps more important, aspect of efficiency is concerned with the amount of multiplexed information that can be transmitted over a channel with a known capacity. For this report, multiplexer efficiency will be defined to mean the ratio of the number of information bits to the minimum number of bits required by the demultiplexer to properly process these information bits times 100%. Symbolically, multiplexer efficiency  $E_m$  is defined as:

$$E_m = \frac{d}{d + \ell} \times 100\%,$$

where

$d$  = number of information bits and  
 $\ell$  = minimum overhead for “ $d$ ” bits.

It should be noted that this definition removes the effects of periodic synchronization words, automatic routing codes, fixed frame length requirements, and input total rate to output channel rate ratios on the efficiency of the multiplexer unless this information is required by the demultiplexer. It further assumes a perfect (i.e., error-free) transmission channel.

**Bit-Stuffing.** Representative formats for bit-stuffing multiplexer systems are shown in Fig. 3 (a), (b), and (c) for various input conditions. It can be seen that in the fully synchronous case, in which all input rates and the output rate are derived from a single clock, the efficiency can be as high as 100%. However, as this synchronism is

removed, and the input rates allowed to vary independently, the amount of information that must be supplied by the multiplexer to enable the demultiplexer to function properly can be significantly large. This inefficiency can be overcome, but probably at the expense of considerable time delay through the system and/or very complex control circuitry.

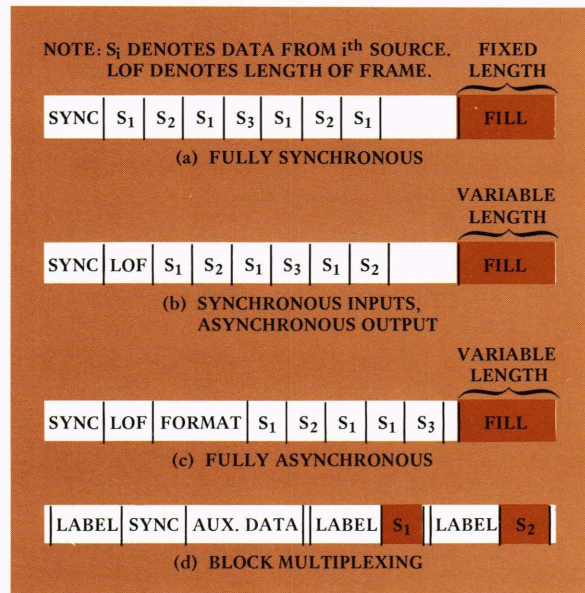


Fig. 3—Typical multiplexing formats.

**Block Multiplexing.** The general format for a block multiplexer is quite different from that for bit-stuffing, as is shown in Fig. 3(d). Each fixed length block of data from a particular individual source has appended to it a number of bits (referred to as a “label”) with a specific pattern that the demultiplexer uses to route the data bits to the appropriate output. If no restrictions on time delay are imposed, the efficiency of a block multiplexer system can be made to approach 100%, limited only by the cost of additional storage registers. The designer must, therefore, choose a storage register size that is compatible with these factors of channel efficiency, time delay, and cost.

The number of bits used as a label has some effect on the efficiency, but since it will generally be much less than the number of bits in the data block, this effect can be kept rather small. However, since a misinterpretation of the relatively few bits in the label will result in the improper routing of the entire block of data bits that follow, the

number of bits in the label should generally be considerably greater than the minimum. This is done so that error detection and correction coding can be included in the labels. The number of bits to be added is determined by the bit-error-rate and over-all error characteristics of the transmission channel that is to be used.

**Demultiplexing Techniques**—The function of the demultiplexer is to separate the multiplexed composite bit stream into the original data sources. Hence, the design of the demultiplexer is essentially dictated by the format, or manner, in which the data are multiplexed. The first problem, of course, is for the demultiplexer to establish synchronization on the composite bit stream. This is generally accomplished by sending a synchronization word, or specific bit sequence, that has good correlation properties to minimize the probability of obtaining false synchronization. This synchronization word may be sent periodically (which is necessary for bit-stuffing multiplexing systems) or at random intervals (which may be used in block multiplexing systems). The important point is that it be transmitted often enough to minimize the time to reacquire synchronization once it has been lost.

Once synchronization has been established on the incoming bit stream, the data may then be separated, or unpacked, in the manner prescribed by the multiplexer design. For example, a bit-stuffing demultiplexer for nonvarying, synchronous sources and output clock is essentially identical to a demultiplexer. The demultiplexer simply counts bit-by-bit after synchronization is acquired, and each count routes the given bit to the proper output channel. The main advantage of this technique is that it is relatively simple to implement.

If the sources are rate-variable and nonsynchronous, the design of the bit-stuffing demultiplexer may or may not be more complex. For this case, the bit-stuffing multiplexer complexity increases significantly. However, once the data have been multiplexed and a concomitant data clock generated, the demultiplexer really cannot tell if the sources were synchronous or nonsynchronous. The dynamic range of the fill bits generally will be increased for this case, and, hence, the demultiplexer decoding circuitry may correspondingly increase.

## **Description of the Multiplexer/ Demultiplexer System**

**Multiplexer Design**—The system selected to demonstrate the feasibility of achieving block multiplexing with a small special-purpose equipment is that shown in Fig. 2. It was constructed with three input channels to meet the requirements for ERTS telemetry. In operation, this system stores incoming data in one of the two registers available to each source. The transfer control section recognizes when a register is filled and shifts the data, with label appended, into the next available output register. During this time, input data are being stored in the other register of the input channel. The output control section needs to merely cycle through the output registers, sequentially shifting data out of each.

*Input Channel.* The logical flow of information for an input channel is shown in Fig. 4. The data bits are entered into either shift register  $SR_{IA}$  or  $SR_{IB}$  at the data source clock rate. Selection of the A or B register is accomplished by input and output selector circuits, as well as clock selectors for each register. All selectors for a single input channel are under the control of a single bistable element that changes state after each “ $d$ ” transition (in one direction) of the data clock.

An additional “ $l$ ” bits are added to the data block via binary-coded-octal thumbwheel switches on the front panel. These switches generate the correct inputs to a parallel input shift register,  $SR_L$ . This information is used as a label for the “ $d$ ” data bits so that correct demultiplexing may be accomplished. The label information is entered immediately after completely filling an input buffer and is not released until just prior to transferring the “ $l + d$ ” bits from an input channel register to an output register.

The remaining input channel logic is used to establish a “fill command”,  $F_i$ . The presence of this signal in the “true” state indicates that an input register for this channel has been filled, that a periodic synchronization data block is not required ( $n$ ), and that this input channel has been selected for transfer of data to an output buffer.

In order to avoid the simultaneous occurrence of the  $F$  signal for more than one channel that would result in an ambiguous decision, the channels are sequentially strobed by an  $MC_i$  signal. Each application of this signal, which is derived from the master clock, occupies a distinct and unique time position. (Specifically the  $MC_{i+1}$  signal has a duration of one-half the master clock

period and occurs one-half a master clock period after the termination of  $MC_i$ .) The  $MC_i$  signal also provides the necessary priority for any input with a rate greater than one-half the output rate providing that source is connected as source #1.

**Transfer Control.** The major purpose of the transfer control section is to recognize that an input channel has a full register and, under the proper conditions, to transfer the contents of that register to the next available output register. In order to ease rate restrictions, this must be done at a high clock rate, which is generated internally.

Also, only the data from one input channel can be transferred at a time and data from a high rate source must be given priority over sources of a lower rate. Both of these functions have been accomplished by generating "strobing" pulses from the internal clock signal ("MC" in Fig. 5). When the transfer control section completes the transfer of " $l + d$ " bits (detected by the transfer bit counter), the input to a modulo 5 counter is enabled and the five strobing pulses are generated ( $MC_0, MC_1, \dots, MC_4$ ). The first pulse,  $MC_0$ , resets to zero all the bistable elements generating the  $F_i$  signals, particularly the one  $F_i$  signal that has been in a true state during the transfer of data just completed. The next pulse,  $MC_1$ , strobes the conditions for setting the  $F$ -generating bistable element in channel 1. If all conditions are met (a full input register and no periodic synchronization block required), the  $F$  signal will be set, data will be transferred from input channel 1, and no further  $MC_i$  pulses will be generated.

If a register has not been filled in channel 1, no

$F$  signal is generated in that channel, and the  $MC_2$  pulse strobes the conditions in channel 2.  $MC_3$  does the same in channel 3. When no input channel has data to be transferred and the next output register to be emptied has not yet been filled, the  $MC_4$  pulse causes synchronization and overhead source data to be entered into the output register.

The "2-bit comparator" in Fig. 5 is used to prevent the transfer of information from an input register to an output register when the output register still contains data from a previous transfer. This is accomplished by detecting the equality of the states of the transfer cycle counter and the read cycle counter. When these counter states are identical, the modulo 5 counter that generates the strobing pulses is held in a reset condition. Once the output register is emptied, this reset signal is removed, and the generation of strobing pulses can be resumed.

The proper routing of information from an input channel to a selected output register is accomplished by the "4:1 selector" followed by a "1:3 selector" to route data and a second "1:3 selector" to apply a clock signal to the proper output register. The 4:1 selector is an AND-OR gate in which each of the  $S_i^*$  signals is gated with its corresponding  $F_i$  signal so that a "data bus" line is formed as the input to the 1:3 selector.

In the event that a periodic synchronization word is required, it can be provided by counting the blocks of data that are transferred and generating a signal " $n$ " for the appropriate block. This signal then inhibits the generation of  $F$  signals for the data channels and causes  $MC_4$  to

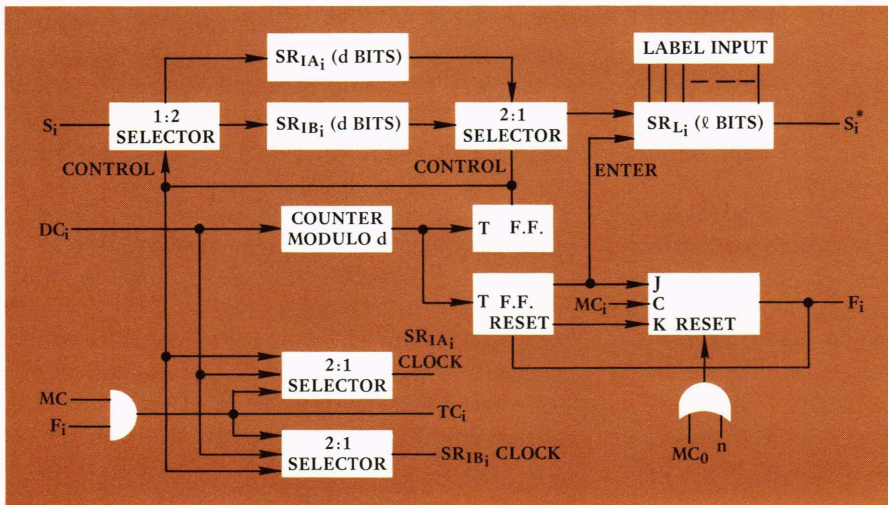
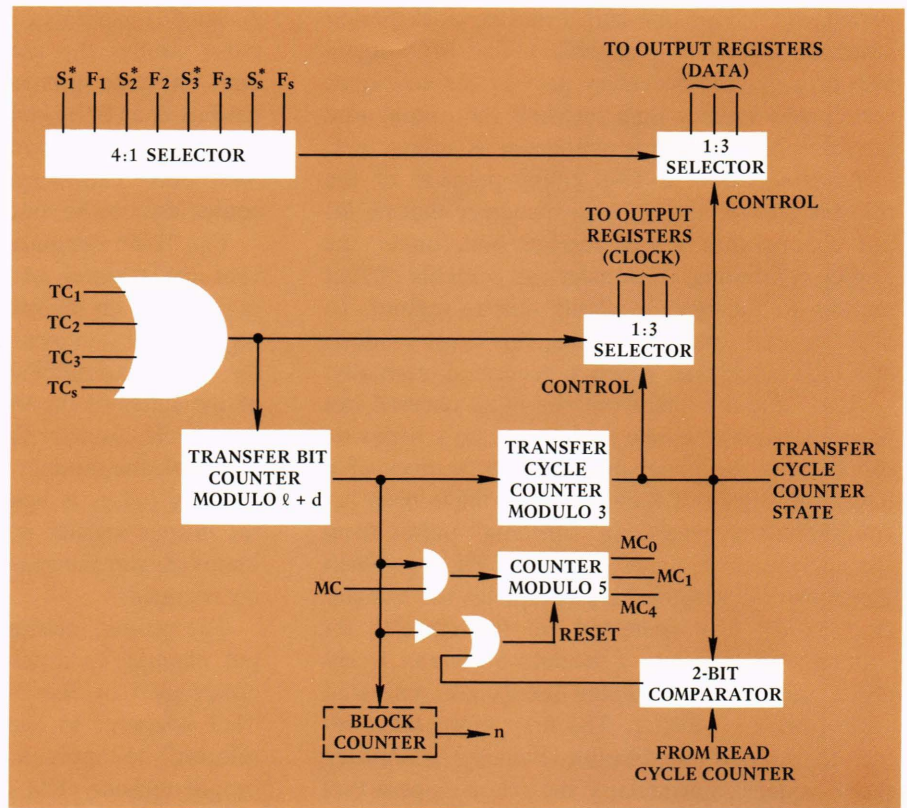


Fig. 4—Multiplexer input channel section.

Fig. 5—Transfer control section.



initiate the transfer of information from the synchronization and overhead source channel.

**Data Formatting**—The selection of the multiplexer output data format is important in that it dictates the design of the demultiplexer and determines to a large extent the efficiency of utilization of the output channel capacity. Many factors influenced the choice of the format for the APL multiplexing system, some of which are discussed in the following.

*Synchronization and Overhead Source.* When the condition arises that no input channel has a storage register filled with data and data are required by the output, the necessary “filler” information is provided by the synchronization and overhead source. Since in block multiplexing this condition persists for a complete block of bits rather than occurring as pseudo-random bits as it does in bit-stuffing, some use can be made of this otherwise wasted transmission channel capacity. If nothing else, at least synchronization information can be transmitted so that the demultiplexer can verify its synchronism at frequent, although aperiodic, intervals.

The condition requiring the insertion of information from the synchronization and overhead source is detected by a comparison circuit, which compares the states of the read cycle counter in the output control section and the transfer cycle counter in the transfer control section. Whenever the transfer counter state is only one greater than the read state, a command is given to transfer information from the synchronization and overhead source to the output register. A second condition for inserting synchronization data is the requirement that such information be supplied periodically. As discussed previously, this is accomplished with the “*n*” signal inhibiting all  $F_i$ 's except  $F_s$  and causing  $F_s$  to be set.

*Block Length.* The selection of a block length for any transmission system generally involves a number of tradeoffs. From an intuitive standpoint, the block should not be so short that overhead information is being sent frequently and the channel capacity is inefficiently utilized, nor so long that the delay time from reception to multiplexer output will become significant for low rate data.

For the APL multiplexing system, a third factor

entered into the block length selection. In this system, it is essential that the block labels be received by the demultiplexer correctly to protect against erroneously routing the data from one source to the output channel of another source. To accomplish this, a study was made of error-correcting codes that would be short enough so as not to significantly decrease the efficiency of the system, yet powerful enough to correct a significant number of errors.

Four labels or code words were required for the APL system (three source and one synchronization label), and an 8-bit near-equidistant code for four messages was selected. Equidistant codes are those for which the distance between pairs of code words are all equal to the maximum possible mean distance. By selecting the code word length to be 8-bits, a minimum distance of five was obtained between the four labels. This minimum distance allows the correction of up to two errors. Thus, three errors would have to be made in the eight bits of a label before invalid decoding of a label would occur.

The probabilities of *not properly* identifying a label (i.e., the label cannot be decoded as one of the four known labels) and *incorrectly* identifying a label (i.e., the label is decoded as the label from a different source) have been calculated. The probability that a label is not properly identified is approximately  $56p^3$ , where  $p$  is the bit error probability. The probability that the label is incorrectly identified is approximately  $20p^3$  in the worst case, tending toward  $10p^3$  in more favorable cases. With a bit error probability of  $10^{-6}$ , these probabilities are of the order of  $10^{-16}$  to  $10^{-17}$ . Even with a bit error probability as high as  $10^{-3}$ , the probability of not correctly recognizing the label is  $5.6 \times 10^{-8}$ , and the probability of incorrect identification is 1 or 2 times  $10^{-8}$ .

Since the label length is considered overhead information and thus affects the efficiency of utilization of the output channel capacity, the block length of the APL multiplexing system was chosen to be 104 bits. This length was selected so that the overhead would contribute less than 10% of the total block length. This block length appears reasonable in that a channel capacity utilization in excess of 90% may be obtained and the time delay from multiplexer input to output is kept rather small (e.g., for a 1 kbps data source and a 50 kbps output rate, the maximum time delay is approximately 100 ms).

**Block Formats.** The final decision regarding the multiplexing system data format was to transmit a synchronization block 104 bits long so all blocks would be of equal length. This not only simplifies the demultiplexer hardware, but provides some additional flexibility. The format of a synchronization block is shown in Fig. 6(a). The block consists of a special 8-bit synchronization label, the 24-bit synchronization pattern, three 8-bit words to indicate the source, destination, and format of the data, and 48 bits that may contain additional housekeeping information or data from an auxiliary source.

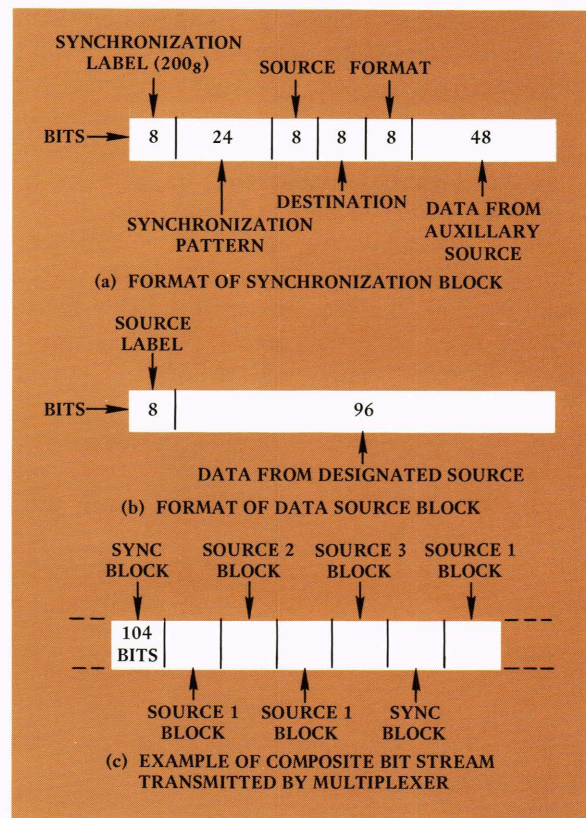


Fig. 6—APL multiplexing system format.

Figure 6(b) shows the format of a data source block, and Fig. 6(c) shows an example of the composite bit stream transmitted by the multiplexer. Note that the multiplexer output is simply a series of equal length blocks, each block being decodable by a distinct label that indicates either a synchronization block or one of the three data source blocks. The number of synchronization blocks transmitted per second is determined by the



multiplexer output line rate and the arithmetic sum of the input sources it is multiplexing. For example, if three 10-kbps sources are multiplexed and transmitted via a 40-kbps data line, approximately 20%, or one out of every five blocks, will be synchronization blocks. This includes the 8% overhead to transmit each data source. If no data are present to be multiplexed, the multiplexer simply sends a series of synchronization blocks, an operation analogous to the generation of an idler pattern by a modem when no data are present for transmission.

**Demultiplexer Design**—As previously discussed, the functions of the demultiplexer are to establish block synchronization on the composite bit stream transmitted by the multiplexer (Fig. 6(c)) and to separate the composite bit stream into the three original data sources. This section describes the basic operation of the demultiplexer on a functional or logic basis rather than on an individual component level.

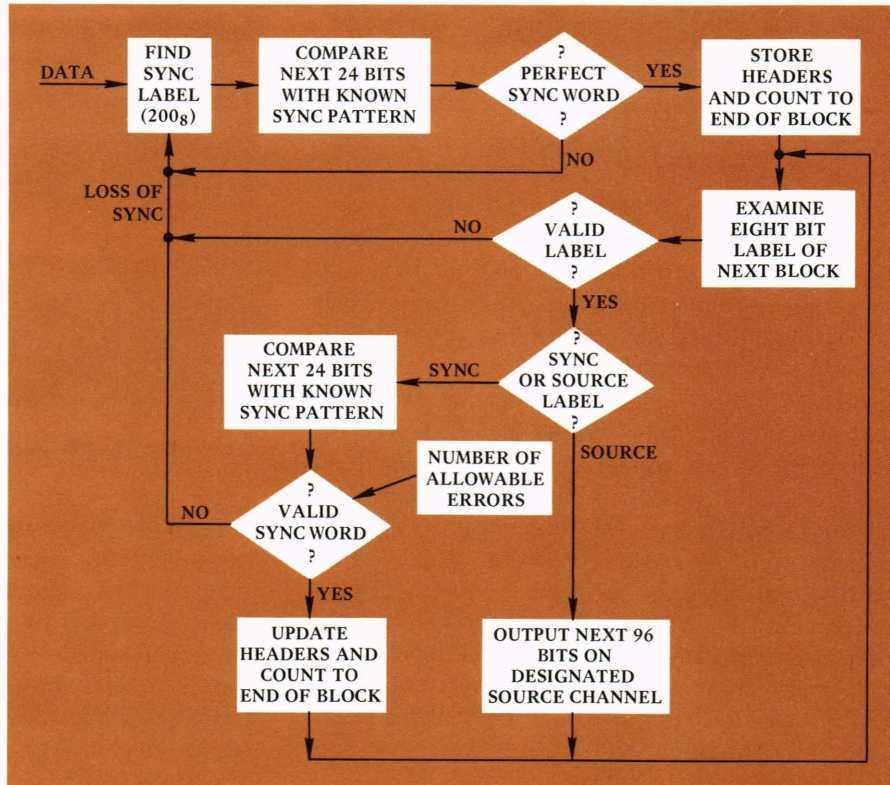
*Basic Operation.* An operational flow diagram for the demultiplexer is shown in Fig. 7. The demultiplexer is designed to search initially for the special synchronization label, which consists of seven “zeroes” followed by a “one”. The com-

posite input bit stream is examined bit-by-bit until a run of seven consecutive “zeroes” is found. The next “one” received activates a bit-by-bit serial comparison of the following 24 bits in the input data stream with the selected master synchronization pattern.

If no errors are found in the received pattern, the demultiplexer is considered to have established synchronization. Upon reception of any bit that does not agree with the known pattern, the demultiplexer is considered to have lost synchronization, and a bit-by-bit search for the synchronization label is resumed. Assuming that a perfect 24-bit pattern has been verified, the next three 8-bit words may be stored for future use if required (see Fig. 6(a)). The remaining 48-bits are presently fill bits, and the demultiplexer simply counts until the end of the block occurs (i.e., 96 bits after the initial synchronization label was found).

Having completed examination of the synchronization block, the next 8 bits in the data stream are examined to determine if the next block is another synchronization block or a data source block. If one of the three possible source labels is found, the next 96 bits are data from that particular source, as shown in the data source block for-

Fig. 7—Operational flow diagram of time-division demultiplexer.



mat of Fig. 6(b). The data bits are then routed to the proper output channel as dictated by the source label.

If the next 8 bits were determined to be the synchronization label, the next 24 bits would again be compared to the known synchronization pattern. In this case, however, some number of disagreements may be tolerated; this allowable number of errors is switch-selectable from zero to seven. Hence, a less stringent bit error rate criterion is allowed once a perfect synchronization pattern has been found.

If the following 8 bits after the synchronization block contain too many errors and cannot be decoded as either the synchronization label or one of the three source labels, synchronization is assumed to be lost, and the bit-by-bit search for the synchronization label and an error-free 24-bit synchronization pattern is resumed.

**System Characteristics**—The multiplexer and demultiplexer (shown in Fig. 8) are each constructed as self-contained units on a standard 19-inch panel 7 inches high. Each has its own power supply integrally mounted with the unit. The electronics for both units consist exclusively of transistor-transistor logic (TTL) circuits manufactured by both Texas Instruments and Fairchild Semiconductor. The multiplexer consists of 11 printed circuit boards containing up to 16 dual-in-line integrated circuit packages per board. Approximately 150 discrete packages are used. The demultiplexer uses four such printed circuit boards and contains approximately 60 discrete packages.

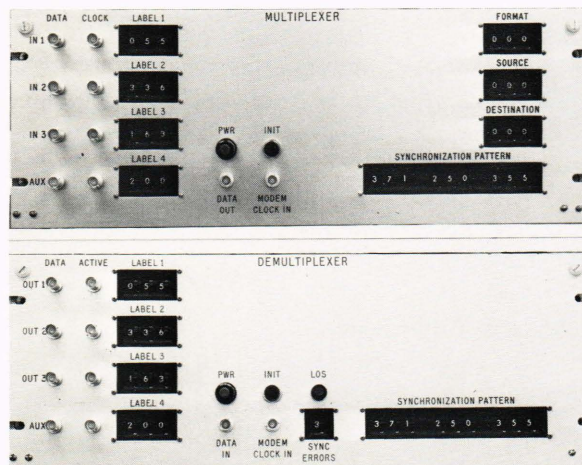


Fig. 8—Multiplexer and demultiplexer units.

Binary-coded-octal thumbwheel switches are used on the front panel of both units. These switches are used to set the three 8-bit source identification labels, the special 8-bit synchronization label, and the 24-bit synchronization word on the multiplexer and demultiplexer. Additionally, the multiplexer has three 8-bit switches to set the format, source, and destination of the data, and the demultiplexer includes a single switch to set the allowable number of errors in a synchronization word.

The system is capable of multiplexing up to three serial pulse-code modulation (PCM) bit streams and may be used for any non-return-to-zero (NRZ) coded data. The maximum output rate limitation for the multiplexer is 953 kbps with a maximum single-channel input bit rate of 461 kbps. These limitations are imposed by the internal multiplexer clock rate of 1 Mbps and the input register configuration.

## System Expansion Potential

The time-division multiplexing system described in this report meets all of the requirements for the ERTS telemetry application. Moreover, the system may be utilized in a wide range of similar applications in which several high-rate bit streams must be multiplexed into a single output channel (e.g., the Central Data Processing Facility application). Numerous possibilities exist for expanding the capabilities of the APL multiplexing system to satisfy additional requirements.

If an increased number of input channels is required (i.e., in excess of three channels), the most direct approach is simply to duplicate the current equipment (both multiplexer and demultiplexer units) and connect two of each in a cascade arrangement. In this way, the number of inputs that can be accommodated is increased from three, for a single multiplexer unit, to five, for two cascaded units. A suitable clock rate would have to be provided to serve as the output clock from one multiplexer and the input clock for the second. Such an arrangement, for example, would probably be satisfactory for relaying data from a remote site to a Central Data Processing Facility for most Skylab telemetry conditions.

Input capacity expansion by cascading multiplexers does have the disadvantage of decreasing channel usage efficiency. This results from having

to provide synchronization and block labeling information in the output of the first multiplexer and then similar type information again in the second multiplexer. Also, some small increase in the delay encountered by those sources that must go through both multiplexers will occur.

Both of these difficulties can be circumvented by increasing the number of basic input channels available. Physically accomplishing this with the current equipments is a nontrivial task, but it can be accomplished with relative ease in a redesigned system. The most significant change is that the modulus of the modulo 5 counter in the transfer control section (Fig. 5) must be made equal to the number of input sources desired plus two. This, then, allows for the generation of a sufficient number of strobing pulses to accommodate the additional inputs. Additionally, an input channel (Fig. 4) must be provided for each source, and such secondary units as the 4:1 selector and the transfer clock combining gate in Fig. 5 must be expanded.

**Data Rates**—In the event that the absolute single channel input rate limitation on the multiplexer of 461 kbps proves inadequate for future missions, this limit may be increased within the current system by replacing the 1-MHz master clock with a 2-MHz clock. This would effectively double both the single channel input rate limit and the output channel rate limit so that inputs of up to 922 kbps and an output as high as 1.9 Mbps could be accommodated. It is not expected that such a change would involve anything further in the multiplexer.

To exceed the above limits would entail a redesign of the entire system using some form of higher-speed logic. Integrated circuits of the "ECL" (emitter-coupled logic) type are now readily available that can be operated at clock rates in excess of 100 MHz. The use of such circuits would result in a multiplexing system design that could easily handle input rates up to 10 Mbps with an output rate at approximately 20 Mbps. It is well known, however, that the physical construction of systems operating at these rates is a nontrivial task.

**Data Rate Regeneration**—The final area for expanding the capabilities of the APL multiplexing system is in the demultiplexer output portion of the system. At present, the data from the various sources are transmitted in bursts of data at the

input, or modem, clock rate. This type of operation will suffice for most digital processing applications as now envisioned. However, there may be cases where it is desirable to have continuous serial bit streams as the outputs from the demultiplexer (e.g., for analog chart recording, oscilloscope presentations, etc.).

A prototype single channel data rate smoothing system has been designed and constructed to demonstrate the feasibility of obtaining continuous or smoothed serial bit streams from the burst-type demultiplexer outputs. The system basically consists of a long shift register that provides temporary storage for the data which enter the register in bursts at the modem clock rate. Data are continuously emptied from the register at or near the expected nominal rate of the source (i.e., the original downlink frequency prior to multiplexing). A voltage-controlled oscillator (VCO) is used as the source for the output clock frequency.

Control of the rate smoothing operation is accomplished by maintaining the number of data bits in the storage register at approximately one-half the register length. If data are entering the register rapidly so that the register is over half full, the VCO or output clock rate is increased. If data are entering the register at a lower rate, so that the register is less than half full, the output rate is correspondingly decreased. Thus, the output rate is continuously adjusted so that the register is always approximately half full, and a continuous serial bit stream is provided. Rate smoothing by this approach does not restore the original instantaneous rate variation of the source. The method, however, does provide a long-term approximation of the original rate and yields gap-free outputs for analog display purposes.

Since the APL multiplexing system uses 96-bit data block lengths, the prototype rate smoothing system was designed to provide a storage capacity of 10 data blocks, or 960 total bits. The system may be used for smoothing original source rates up to 1 Mbps with a maximum modem clock rate of 2 Mbps.

## Acknowledgment

The authors would like to acknowledge the significant contributions, especially in the area of coding, made by A. G. Carlton, whose technical guidance and direction aided the project immeasurably.