

---

The Space Congress® Proceedings

1966 (3rd) The Challenge of Space

---

Mar 7th, 8:00 AM

## Application of General Purpose Computer Techniques to the Design of a Programmable PCM Telemetry Decommulator

D. J. Hermen

*Radio Corporation of America, Missile & Surface Radar Division*

J. O. Horsley

*Radio Corporation of America, Missile & Surface Radar Division*

Follow this and additional works at: <https://commons.erau.edu/space-congress-proceedings>

---

### Scholarly Commons Citation

Hermen, D. J. and Horsley, J. O., "Application of General Purpose Computer Techniques to the Design of a Programmable PCM Telemetry Decommulator" (1966). *The Space Congress® Proceedings*. 2.

<https://commons.erau.edu/space-congress-proceedings/proceedings-1966-3rd/session-7/2>

This Event is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in The Space Congress® Proceedings by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

**EMBRY-RIDDLE**  
Aeronautical University™  
SCHOLARLY COMMONS

APPLICATION OF GENERAL PURPOSE COMPUTER TECHNIQUES  
TO THE DESIGN OF A PROGRAMMABLE  
PCM TELEMETRY DECOMMUTATOR

D. J. Herman  
J. O. Horsley  
Radio Corporation of America  
Missile & Surface Radar Division  
Moorestown, New Jersey

### Summary

This paper describes design of a programmable decommutator for multi-format PCM telemetry data. The decommutator organization includes a program counter, instruction decoding registers, and an arithmetic unit, together with pattern comparison logic, and synchronization decision networks.

The various types of input formats are discussed and their implications on the organization are determined. The input data formats include (a) normal PCM commutation with frame synchronization and subcommutation synchronization, (b) mixes of different rate PCM commutation, (c) word synchronization and (d) data address tags with each data word.

The decommutator is organized around a coincident current memory which serves for both program storage and data buffering. A system of program interrupts is also provided so that (a) data words can be requested from the buffer by an asynchronous device, and (b) new programs can be loaded into unused sections of memory. The second feature permits immediate changes of input format by initiating only one manual command. The address and control fields of the program instructions are defined to allow all of the synchronization parameters (i.e. error tolerance) to be contained as part of the program.

### Introduction

In order to meet the demands for increasing quantities of real time telemetered data the AFETR soon will have in operation the Real Time Telemetry Data System. The purpose of this system is to collect and select telemetry data at various down range stations, reformat the selected data for transmission to the central site and then at the central site distribute the data to the range users. See reference 3 for a discussion of the system background. The remote site equipment includes data selection units which provide an interface to accept data from AFETR's normal decommutators. The data selectors supply the data desired for retransmission to a PCM formatting buffer which in turn drives a conventional data modem. The data output of this formatting buffer will at times contain information from several telemetry links which are not at the same sampling rates. The results is a composite

output containing all of the prime rates plus their respective sub-commutation rates. The decommutator described in this paper is specifically designed for operation with this format as an input signal. The Real Time Telemetry Data System can retransmit data at the rate of 200 kilobits per second, with the words being of variable length (four to 12 bits). The maximum word rate is therefore 50 kilo words per second. Synchronization patterns are four to 64 bits in length for either frame or subcommutation use. Also, because of the data mixing at the remote site, a maximum of 15 independent sub-commutation rates are allowed. Reference 2 contains details of the system specifications.

The first problem of a decommutator is that of searching for synchronization patterns in the presence of noise and also maintaining synchronization through fade periods. The decommutator must also strip each data word from the data stream and provide a temporary storage location for the word until the word can be forwarded to the proper data user.

In order to find the frame synchronization pattern, the decommutator must perform a bit by bit search of the incoming data stream for that pattern and also allow some percentage of the pattern's bits to be in error. Once the initial search finds the pattern a verify mode is entered. This verify mode checks the pattern through a predetermined number of frames allowing a larger percentage of bits to be in error. When the correct number of patterns have been checked, successfully a lock mode is entered. In the lock mode, the pattern is checked each frame, allowing again a different percentage of bits in error in each pattern. The lock mode is continued and the correct and incorrect pattern results tabulated and compared against tolerances limiting the number of incorrect patterns allowable. These modes conform to the requirements of sync detection techniques as described in reference 1.

After frame synchronization verify is entered, search is begun for each of the fifteen subcommutation patterns in their appropriate windows in the data stream. Each pattern can independently proceed through a search, verify and lock status. If frame synchronization ever returns to the search status, all subcommutations are also returned to a search status.

The basic design approach for the decommutator involves the use of a stored program

operating a coincident current memory system. The stored program will identify the particular format expected, store all of the error tolerances and provide executive space for all counters and error histories. The only information stored permanently in hardware external to the memory is the synchronization status. The memory, in addition to providing program storage, is also used for data storage.

The program used for decommutation is organized around program counters which call out operating instructions for the disposition of prime rate channel words. If a prime channel is subcommutated, that instruction will call a jump to a sequence of instructions for the particular subcommutation sequence. Also, each subcommutation sequence has a secondary program counter or indexing function.

An arithmetic unit is used to modify the program contents as the format proceeds, and also to calculate error overflows.

### Input Formats

The input formats expected for this decommutator are the various formats utilized in the Real Time Telemetry Data System. The primary format (see Figure 1a) is a simple PCM format of one basic frame rate (with a frame synchronization code) and up to fifteen independent subcommutation rates. Any number of channels may be subcommutated. However, each subcommutation pattern is limited to 64 bits. Also, prime rate channels can be subcommutated without having its own synchronization pattern utilizing another subcommutated channel as a synchronization reference. A subcommutation channel of this type is referred to as a slaved channel. Channels containing synchronization patterns are referred to as independent channels.

If more than one telemetry link is being retransmitted in the message from one remote site, several prime sample rates will be present in the data stream. Only one frame synchronization pattern will be searched for, the remaining rates are handled through the use of multiple program counters and jump instructions. Figure 1b shows a mixed format of two links (x and y). Three channels of x information and two channels of y information are present in that format. In this mode a prime channel data instruction designates from which link the next word is derived and creates a jump to that corresponding program counter. A maximum of seven program counters are allowed in any format.

The Real Time Telemetry Data System can also handle formats utilizing word synchronization. In this format each data word is preceded by a word synchronization pattern of a fixed number of bits (see Figure 1c). To process this format the search procedure is

used on the word synchronization pattern. Once word synchronization is obtained the frame synchronization pattern is searched for in the normal data windows. Also, the word synchronization patterns are treated as a data link group of one word and its program counter, with the PCM data operating from a second program counter.

Another alternate format is used for implementing data smoothing or a data compaction mode. This format (Figure 1d) contains data words preceded by 9 bit address tags. The decommutator strips the address and transfers it to the memory address bus, and on the next "data complete" signal, provides storage of the associated data word in the memory location specified.

## Logic Organization and Description

### General Logic Description

To provide for the many operational modes and secondary requirements resulting for them, this decommutator has been designed as a programmable device. As such, it has been necessary to consider all of the specific operational requirements and then to develop an instruction repertoire and programming procedure which is as simple as possible. The primary instructions of this device are (a) master instructions which detail the disposition of each prime channel of the input format, and (b) subcommutated word instructions which detail the disposition of each subcommutated channel word. The basic functions of synchronization data decommutation and storage, quality assessment, and data read-out are accomplished by sequencing through a group of instructions in a manner which considers the required function, priority, and available time. A close analogy exists between this decommutator and more conventional decommutators when it is realized that the stored instructions of this decommutator are equivalent to the counters and control logic of non-programmable decommutators. Also, the storage of synchronization patterns and error tolerances in this decommutator have their equivalent in simpler decommutators where analog variables are normally set for pattern recognition and synchronization evaluation.

In the following explanation of the decommutator operation, the reader is referred to a block diagram (Figure 2), a flow chart (Figure 3), and a timing diagram (Figure 4) of the subject decommutator.

### Deccommutator Block Diagram Description

An overall block diagram of the decommutator is shown in figure 2. Prior to the decommutation of any data channel, a set of instructions are read from memory and temporarily stored in their respective registers. These instruction words are read from memory in the manner illustrated in the flow chart shown in Figure 3 and the timing diagram of Figure 4.

The I instruction is always read from memory first and starts the sequence of instruction loading. The I, A, B, and C instructions contain bits used to address memory locations, directly or indirectly, and control bits to specify the sequence of operation in decommutating the forthcoming data word. The type P and SP instructions similarly require a particular storage register to implement the sync logic, although these and type B instructions never require modification and are immediately rewritten to memory.

If indirect addressing of a memory cycle require the use of a C instruction, the parallel digital adder adds the sync index number of the C auxiliary location to the contents of the data location of the A instruction in order to find the current B instruction. The resultant sum is used to read a B instruction from memory. The adder is also used to modify the content of the instructions before re-entry into the memory and to make various comparisons required in the system for proper operation.

A status flip-flop storage register is used to keep track of the synchronization status of each independent synchronization rate and sub-rate. This register contains two bits of storage for each rate or sub-rate synchronization and specifies search, verify, or lock mode for each rate or sub-rate. Decoding the "Sync Index" field of the A instruction selects the information about the appropriate channel (i.e. search, verify, or lock mode).

A timing slot generator is provided for timing of the required control and transfer cycle within the system. The slot generator is started by the bit per word counter reaching zero and is terminated at the end of each word decommutation cycle. A complete cycle of this type never exceeds 20 microseconds. Within this 20 microsecond period, priority interrupts are allowed at certain times with a maximum of two interrupts for either program load input or data output in each complete cycle which requires memory access. No demand for interrupt must wait more than 12 microseconds before being serviced.

A digital comparator is provided to check the incoming pattern against the expected pattern. Shift registers are used to transfer the number of discrepancies to an error counter whose content is transferred to the adder/subtractor, at the appropriate time slot, for comparison against the allowable tolerance.

#### Auxiliary Functions

Provisions in the decommutator are made to initialize all executive memory location to known starting places prior to the start of a program or whenever synchronization is lost

and must be reacquired. During the initialization cycle, all I instructions are first read and modified to insure that the first A instruction will be read during the first program run. Similarly, all A, B, and C instructions are initialized to known starting points (i.e. accumulated errors reset to zero, etc.).

Prior to the start of synchronization cycle, the expected pattern is transferred to the P register and all other necessary instructions decoded to set up the condition for a synchronization cycle. When the command to initiate a sync cycle is generated, the tolerances to be used in the decision logic is transferred to the adder/subtractor for comparison against the actual counted errors. If the actual errors counted are within the allowable tolerance, synchronization has been accomplished. Each time synchronization is found to be within limits, the "In-Tolerance" field of C instruction is incremented. When this "In-Tolerance" field contains a count greater than the "In-History Tolerance" field of the P instruction, then the decommutator is allowed to advance to the next mode of operation (i.e. verify mode to lock mode). Similarly, continued incrementation of the "out-tolerance history" will eventually cause the decommutator to revert to the frame sync search mode.

In the process of pattern synchronization, the decommutator reviews the primary frame synchronization patterns and determines the compliance of each received synchronization bit with the expected synchronization pattern. In the process, individual triggers are generated which represent compliance and non-compliance with each bit in the pattern. These triggers will then be used to determine the magnitude of the bit error rate on the serial data received over the communications media.

A built-in simulator allows known data to be generated internal to the decommutator. The simulator serves as another input to the interrupt logic and is serviced at the appropriate time. The known data is loaded into the memory from the console and is available to the simulator upon request.

#### Operator Functions

Except for power turn-on and maintenance all operator functions are performed at a central console. Programs may be loaded by either paper tape, magnetic tape, or manual insertion with the decommutator in any mode of operation except frame sync search mode. All program mode changes (i.e. Standby, Operate, or Test) are initiated from the console and cause the decommutator to reinitialize its instructions.

When there is a need for trouble-shooting, a portable maintenance unit may be plugged into the decommutator which allows step-by-step



operations to be checked. The maintenance unit displays the contents of the I, A, and B registers and indicates the slot number of the Time Slot generator.

### Program Instructions and Operating Codes

#### General

The operating logic of the decommutator uses three basic types of instructions, arbitrarily designated I, A, and B, and three types of auxiliary instructions designated as C, P, and SP. The use of the auxiliary instructions depends upon the operating mode and status of the decommutator as well as the operating codes of the basic instructions. The instruction formats are shown in Figure 5.

#### Index Instruction "I" used as a Program Counter

The index instruction always starts a sequence of operations. The "A Instruction Location" field of the I instruction contains the current A instruction memory address. This field is incremented by one before restoring the I instruction to memory unless a "Last A" Control Bit is raised in the current A instruction. In the latter case, the "First A Instruction Location" field is substituted for the current A instruction location before restoring the I instruction to memory.

Two error tolerance fields are included in the I instruction for use during the initial search for frame synchronization or when reversion to the Frame Sync Search Mode is flagged by the decision logic.

Control Bits Y and Z of the I instruction are used in conjunction with program load and initialization functions.

#### Master Instruction "A"

The A instruction contains the essential information related to each prime channel. There is one A instruction for each channel to be decommutated.

Each I instruction uses the contents of the "Current A Instruction Location" field to read an A instruction from memory. The A instruction has a "Sync Status Control" field that provided answers to the following questions:

- (1) Is this a prime frame channel?
- (2) Is this a prime frame synchronization channel?
- (3) Is this a continued prime frame synchronization channel? (Patterns of more than 12 bits are treated in a syllable technique.)

- (4) Is this a subcommutated channel?
- (5) Does this subcommutated channel contain synchronization information (i.e. is it an independent subcommutated channel)?
- (6) Is this subcommutated synchronization pattern a continued pattern?

Control Bits Y and Z are used during the Data Compaction and Asynchronous modes respectively. When control bit Y is raised, the data is to be stored in the memory specified by the data tag address register. When control bit Z is raised, the data is raised, the data is transferred to a Magnetic Tape Recorder for storage.

A sync index field is used to address sync index instructions C.

A control bit designated "Last A" when raised causes an unconditional jump to the "First Channel Location" when the next I instruction is read from memory.

The "Data Location" field of the A instruction is sometimes used to specify the location of other instructions. The use of this field varies according to the interpretation of the "Sync Status Control" field, and are as follows:

- (1) If prime channel is indicated, the field specifies the location to store data.
- (2) If prime channel frame synchronization is indicated, the field specifies the location of a pattern.
- (3) If an independent subcommutated data channel is indicated, the field specifies the location of the "First B" instruction (programmed constant non-incremented field).
- (4) If a dependent subcommutated data channel is indicated, the field specifies the "Current B" instruction location (incremented field).

The "Word Length" field is used to control decommutation of variable or fixed word lengths and are transferred to the bit per word counter. Each decommutation cycle must wait for this counter to become zero.

The "Next I" field is used to specify the next I instruction for the next channel to be decommutated in the main frame. In the absence of mixed data, the same I code is always used. However, if data from different systems are mixed, the I code will specify one of several I instructions stored in memory.

### Sub-Commuation Instruction "B"

A "B" instruction is read from memory if the "Sync Index Code" of the "A" instruction indicates the channel to be subcommutated.

Subcommutated data channels are either independent subcommutated data channels which contain synchronization data in some of the subcommutated data words, or dependent subcommutated data channels which are slaved to one of the independent channels.

An independent B instruction uses the "Data Location" field to specify the location in memory to store data or the location of a synchronization pattern. If the location of a synchronization pattern is specified, then a portion of the "First B Instruction Location" field is used to specify the Sub-Com Search Tolerance.

The dependent B instruction always uses the data location field to specify the location to store data and uses the first B location field of the B instruction to specify the original contents of the word counter for the slaved channel. The first B location is transferred to the data location field of the A instruction when the Last B instruction of a sequence is detected. Thus, the current B location field of the A instruction is always incremented and/or recycled in the same way as the current A location field of the I instruction.

The Time Data Control Flag Bit denotes that the word to be stored is a correlated time word, and that the word will be stored in the memory location specified by the data location field of the B instruction.

The Sync Word Control Flag is raised when the sync portion of the independent subcommutated channel is due. This control bit causes sync patterns to be called up and appropriate transfers of error tolerances to be initiated at the proper time slot.

Control bits to allow storage of data in groups is also provided.

### Sync Index Instruction "C"

An auxiliary location is always read from memory when the A instruction specifies that the channel contains synchronization information or independent channel subcommutated data. The sync index word field of the C instruction is used to locate the current B instruction address. The sync index word is used in the following manner:

- (1) If the independent subcommutated channel sync data routine is in

verify or lock mode and the "Last B" flag bit not raised, then the sync index word is added to the First B Instruction address (Data Location field of A instruction) and the sum transferred to the memory address bus to read the independent B instruction.

- (2) If the independent channel is in the search mode, the sync index word is first cleared.
- (3) The sync index field is cleared when the Last B flag is raised.

The two control flags are provided in the C instruction which effectively double the sync pattern check capability when both the direct code and its complement are to be checked. These flags, when set, may be displayed to the console operator. These two control bits are sync complement code flag and sync code/complement history flag.

The contents of the five remaining fields of the C instruction are as follows:

- (1) Accumulated errors - used to store the accumulated errors counted for each section of a continued sync pattern until the pattern check is complete.
- (2) Verify error tolerance - contents transferred to the adder/subtractor for comparison to error counter.
- (3) Lock Mode Error Tolerance.
- (4) Out-of-tolerance (error) history - counter preserves the number of times a complete pattern check in any single mode exceeds the check tolerance for the mode and channel. When the contents of this counter exceeds the tolerance, the decommutator reverts to the initial type search mode.
- (5) In-Tolerance History - counter preserves the number of times a complete pattern check in any single mode is within the check tolerance for that mode and channel. If the "in-tolerance history" limit is exceeded when in verify or lock mode, the entire mode history is zeroed, and, if the channel is in verify mode, the channel mode progresses to lock mode.

### Synchronization Pattern Instruction "P"

An auxiliary location sync pattern instruction P is read from memory during the pattern recognition cycle of any synchronization mode. The pattern field contains the actual pattern to be used to check against the expected incoming pattern.

An asynchronous control bit is used to flag the word sync portion of the data word and indicates that a bit-by-bit search is to be performed.

Another control bit is used to designate long patterns (i.e. patterns longer than the comparator).

The four remaining fields of the P instruction contain the following data:

- (1) Verify Mode Sync History - Out of Tolerance Limits.
- (2) Verify Mode Sync History - In Tolerance Limits.
- (3) Lock Mode Sync History - Out of Tolerance Limits.
- (4) Lock Mode Sync History - In Tolerance Limits.

The limit value fields for the proper mode are compared to the out of tolerance history counter of the C instruction in the proper time slot of each sync pattern recognition cycle. If the "out-of-tolerance" limit is exceeded before the "in-tolerance" limit is exceeded, the independent channel mode is caused to revert to search mode. If the reverse is true when in verify mode, the channel mode is progressed to lock mode.

#### Special Pattern Instruction "SP"

This instruction is used only for the initial pattern check in the frame sync search mode and for patterns that exceed the maximum word length of normal channels.

In addition to the "Second Section of Pattern for Search Mode" field, there are three fields to properly modify the word length bit counter of the A instruction, for the two-word pattern check.

#### Conclusion

The decommutator described in this paper has been constructed using integrated circuits for the gate and flip-flop elements. The basic circuits are contained on 3" x 4" cards which mount in a rack mounting drawer assembly (see Figure 6). Several areas of the logic utilize pre-wired logic on the 3" x 4" cards. One drawer assembly contains the entire logic package with a 25% spare space for expansion if required; a second package contains the memory with its associated current driver, sense amplifier, address decoding and power supply. Memory, logic, and power control elements are all mounted in a single rack.

A panel which controls the program loading is shared with other units in the Real Time Telemetry Data System and is included in a central operating console. Power supplies are also shared and are included in another rack.

The decommutator is composed entirely of digital logic elements. Synchronization patterns are examined in a digital comparator, errors totaled in a binary counter, and the results subtracted from the tolerances in an arithmetic unit. All error tolerances and patterns are contained in the program; also temporary storage which provide a fly-wheel effect, is provided in executive program locations. Two complete programs can be loaded into memory providing immediate switching from one input format to another. The telemetered data can be stored in memory for interrogation by an asynchronous data user, or transferred immediately with an address tag to external devices.

The present decommutator is limited to a maximum of 200 kilobits per second input due to the frame synchronization search techniques. Minimal logic changes and additions could double the input capability. The use of faster logic circuits could double that figure. The two micro-second memory system limits the input word rate to 50 kilowords per second. Use of a faster memory would produce proportional speed improvements.

The basic functions of a decommutator could be performed by many general purpose computers. There are two configurations for general purpose computers used as a decommutator.

- (1) In a system which enters the telemetry data, in digital form, directly into a computer. All operations performed on the data are accomplished by the computer. This includes synchronization operations, decommutation, data formatting and routing.
- (2) In a hybrid system, where the handling of the telemetry data is shared between fixed hardware and a computer.

For type (1), studies indicate that a system of this type would present the ultimate in decommutator flexibility; however, present computers cannot handle the required telemetry data rates.

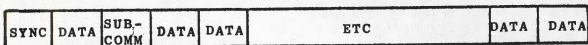
Systems of types (2) (above) have many advantages including maximum use of "off-the-shelf" hardware and flexibility and probably indicates the near future generation of decommutators.

The programmable decommutator described in this paper provides extreme flexibility in the handling of PCM data formats. Many of the logical functions used in this decommutator will be directly adaptable to the hybrid system.

References

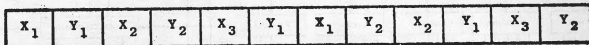
1. Williard, M. W. Mean Time to Establish  
PCM Synchronization  
  
1962 National Symposium on Space  
Electronics and Telemetry
2. Trigg, H. W. The AFETR Real Time  
Telemetry Data System  
  
1965 International Space Electronics  
Symposium
3. Herrburger, E. G. Considerations In the  
Design of a Ground Data Control  
System for Real Time Telemetry
4. Williard, M. W. PCM Telemetry Synchroni-  
zation  
  
1961 National Telemetry Conference



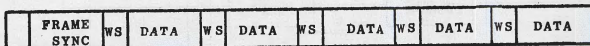


CHANNEL 1 ----- n

(a) NORMAL PCM OUTPUT FRAME

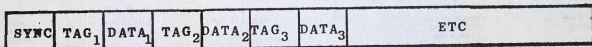


(b) MIXED PCM OUTPUT FRAME



WORD SYNC

(c) WORD SYNCHRONIZATION



(d) DATA PLUS ADDRESS TAGS

FIGURE 1 Input Data Formats

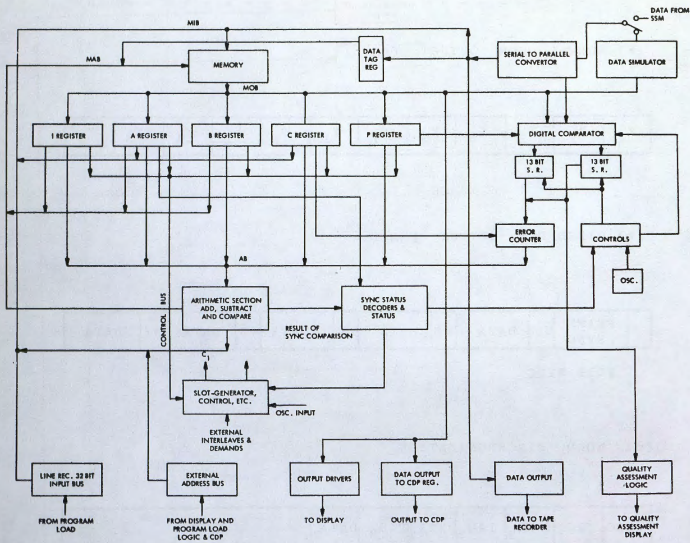


FIGURE 2 Decommutator Block Diagram

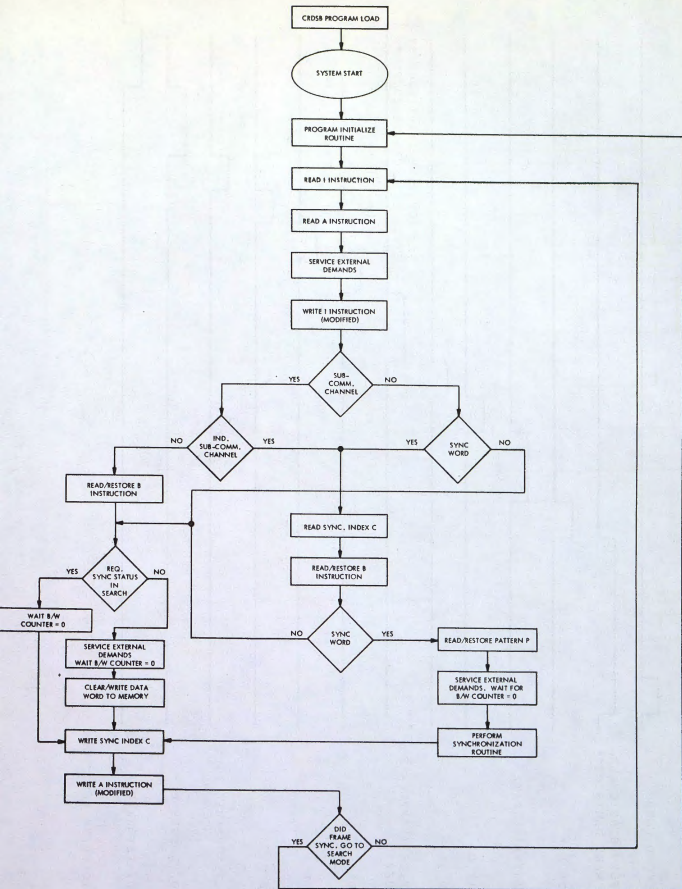


FIGURE 3 Operation Flow Chart

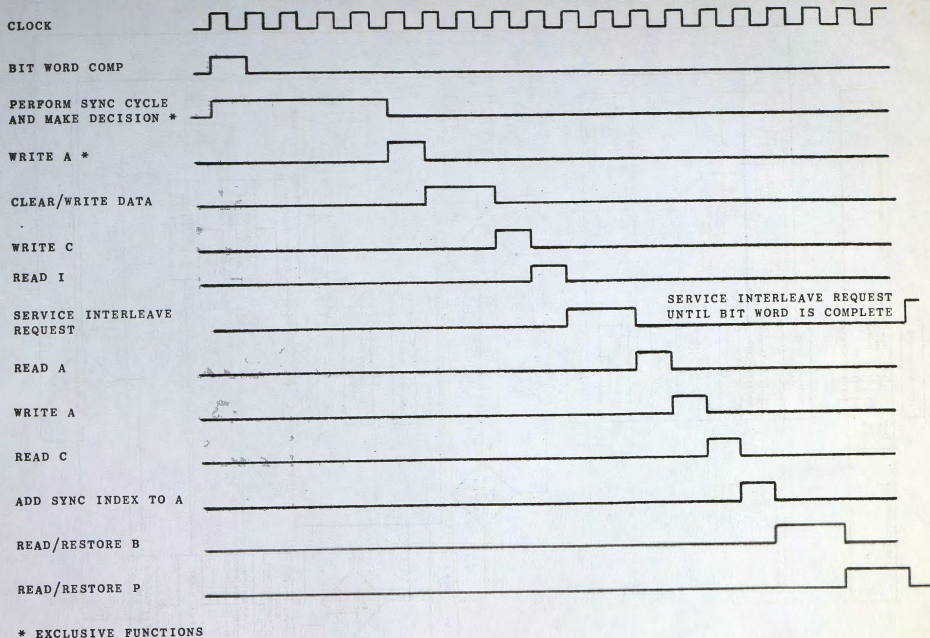


FIGURE 4 Decommutator Timing Diagram



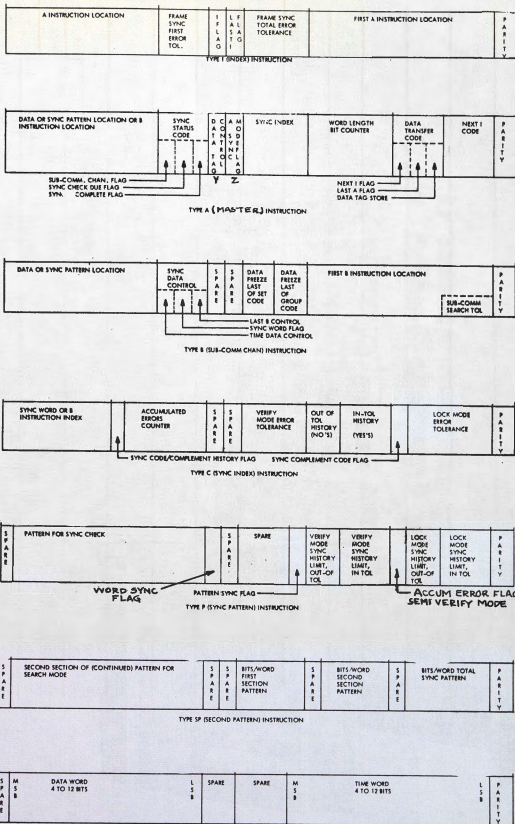


FIGURE 5 Program Instruction Formats

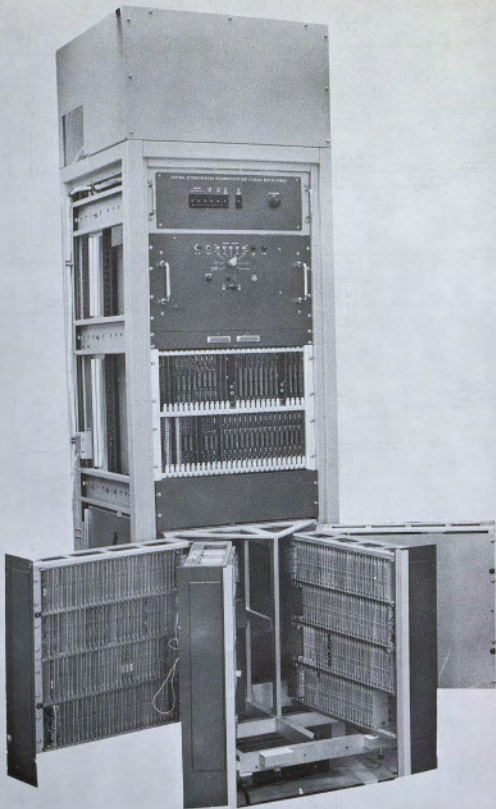


FIGURE 6 Front Elevation of Decommulator Rock