FAA-RD-74-197

Project Report
ATC-43

DABS Channel Management

E. J. Kelly

8 January 1975

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

Technical Report Documentation Page

1. Report No. FAA-RD-74-197	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle DABS Channel Management		5. Report Date 8 January 1975
		6. Performing Organization Code
7. Author(s) E.J. Kelly		8. Performing Organization Report No. ATC-43
9. Performing Organization Name and Address Massachusetts Institute of Technology, Lincoln Laboratory, P.O. Box 73, Lexington, Massachusetts 02173		10. Work Unit No. (TRAIS) 45364
		11. Contract or Grant No. IAG DOT-FA-72-WAI-261
12. Sponsoring Agency Name and Address Department of Transportation, Federal Aviation Administration, Systems Research and Development Service, Washington, D.C. 20591		13. Type of Report and Period Covered Project Report
		14. Sponsoring Agency Code

15. Supplementary Notes

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology under Air Force Contract F19628-73-C-0002.

16. Abstract

This report provides a full discussion of the design of the DABS channel management function. Channel management regulates all activity on the RF channel, scheduling ATCRBS interrogations and DABS roll-call interrogations and replies. Channel management arranges for multiple interrogations when required to satisfy both the surveillance and the communications functions of the sensor, and for reinterrogations in the event of link failure.

The report consists of three major parts: Part I, Channel Control; Part II, Roll-Call Scheduling; and Part III, Transaction Preparation, Target List Update, and Transaction Update. Together they encompass the five basic subfunctions of channel management.

17. Key Words DABS, Channel Management, Interrogation Scheduling, ATCRBS Scheduling		18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 246	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

			Page
Introduction			1
Part I		Channel Control	15
	I-1	Overview	15
	I-2	Time Line Structure and Frame Management	18
	I-3	Communication with the Hardware	25
	I-4	ATCRBS Scheduling	30
	I-5	DABS Roll-Call Schedule Management	37
	I-6	A Channel Control Algorithm	48
Part II		Roll-Call Scheduling	63
	II-1	Overview	63
	II-2	The Active Target List	66
	II-3	Message Formatting and Control Command Formulation	71
	II-4	Allocation and Target Selection	78
	II-5	The Schedule Precursor	86
	II-6	Interrogation/Reply Scheduling	91
	II-7	Scheduling Synchronized Transactions	100
	II-8	Scheduling Downlink ELM Transactions	106
	II-9	Interrogation Scheduling without Reply Processing	109
	II-10	A Roll-Call Scheduling Algorithm	111
Part III		Transaction Preparation, Target List Update and Transaction Update	133
	III-1	Overview	133
	III-2	Design Principles	134
	III-3	Structure of the Target Transaction Block and the Active Target List	140
	III-4	Preparation of the Target Transaction Block	143
	III-5	Updating the Active Target List	161
	III-6	Reply Transfer and Reply Appraisal	165

TABLE OF CONTENTS (Continued)

III-7	Range Completion and the Altitude Echo	173
III-8	Azimuth Completion	177
III-9	ATCRBS Identity Request Protocol	181
III-10	The Pilot Button Channel Protocol	185
III-11	The Comm-B Channel	189
III-12	Extended Length Message Protocols	195
III-13	Transaction Block and Active Target List Header Update	200
III-14	Algorithms and Flowcharts	202
References		238
Appendix A	The Isolation Problem	239

INTRODUCTION

The major software functions in the DABS sensor are channel management, surveillance processing, data link processing, and network management. Channel management regulates the use of the RF channel, and this task is the subject of the present report. The interaction of channel management with the other functions is also discussed. Surveillance processing carries out tracking and position reporting, data link processing regulates the transfer of messages on the DABS data link, and network management coordinates the sensors of the network.

Any sensor which repeatedly transmits pulses (radar) or interrogations (beacon) and then listens for echoes or replies exercises "channel management," the regulation of activity on the RF channel. In a radar/beacon system of conventional type, channel management is very simple and its design amounts to little more than application of the radar scanning equation. In the DABS sensor, channel management is more complex. The major reasons are the following:

a) The DABS sensor must operate with two classes of transponder -- ATCRBS and DABS transponders -- which share the ATCRBS frequencies on uplink and downlink. Thus, one channel is used to support two surveillance systems at the same time.

b) DABS interrogations address single aircraft discretely, therefore the sensor must be able to predict when the aircraft will be within the antenna beam.

c) Channel time must be allocated to each individual DABS interrogation and reply. Therefore, a prediction of aircraft range is required.

d) DABS surveillance procedures often require more than one interrogation to each aircraft, and the needs of data link will sometimes require still further calls. If a given interrogation fails to produce a useable reply, the sensor must be able to repeat the attempt as long as the aircraft remains in the beam.

The channel management strategy used to meet the above requirements has two main features. First, the RF channel is time-shared between the ATCRBS function and the DABS function as shown in Figure 1. ATCRBS interrogations are transmitted at a rate which is low enough to leave time between successive ATCRBS interrogation/listening periods. This time is used to perform the DABS functions. Second, the sensor maintains a regularly updated list of DABS targets within the antenna beam and it utilizes the DABS time to make repeated passes through this list, scheduling discretely-addressed DABS interrogations and replies on a non-conflicting basis. A single aircraft may appear on one or more of the resulting schedules of interrogations and replies, so that multiple surveillance and communications tasks can be accomplished. In the case of reply failure, the repeated scheduling of interrogations to an aircraft provides a high overall surveillance/communications reliability.

Channel management is supported by the other software functions in several ways. Surveillance processing provides the predicted position of

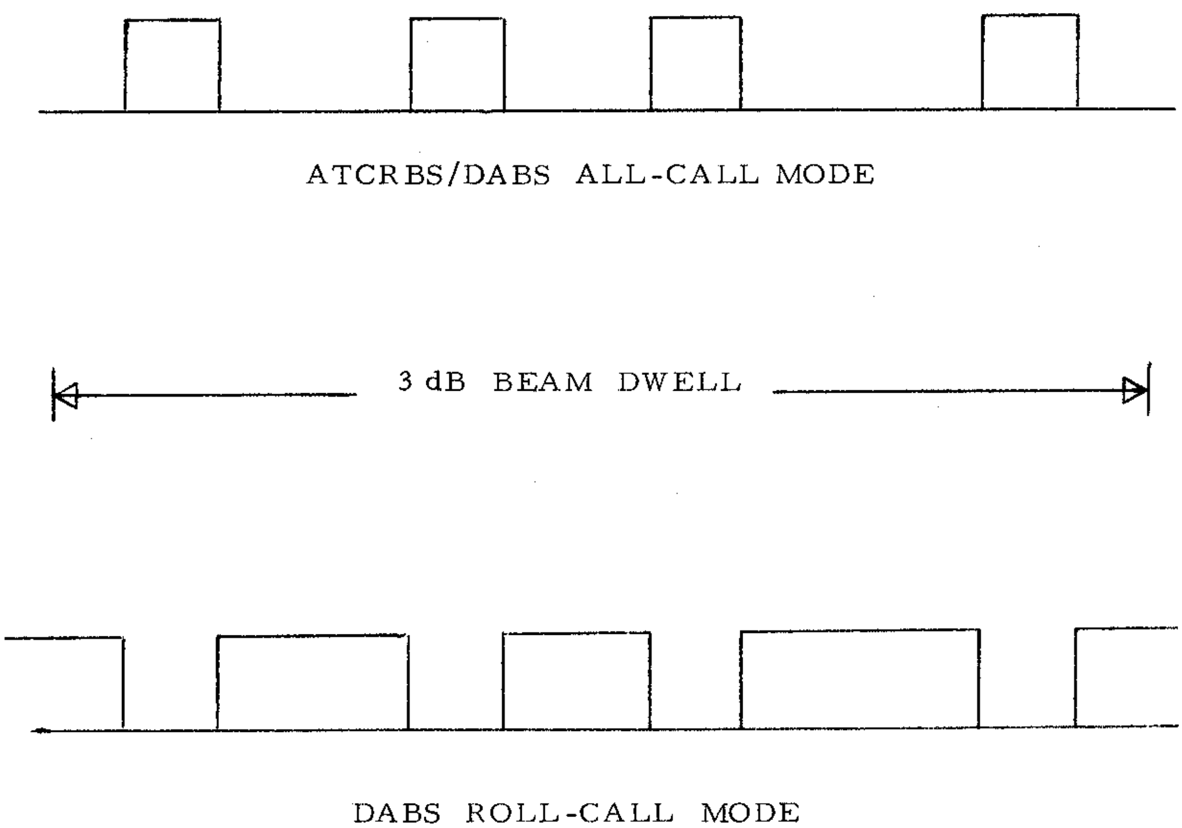


Figure 1. ATCRBS/DABS Time Sharing

each DABS target in track, so that the aircraft may be interrogated while it is within the antenna beam. Predicted range is also needed in the scheduling process so that replies can be anticipated and received without interference from interrogations or other replies. Data link processing provides organized lists of pending messages for each DABS-equipped aircraft so that channel management can determine in advance the number and character of the interrogation/reply pairs to be scheduled to each target when it appears within the beam. Network management exercises dynamic control over the kinds of service, both surveillance and communications, to be afforded each aircraft.

Channel management has complete control over the transmitter/modulation control unit and the DABS and ATCRBS reply processors. Channel management communicates with these units by means of a stream of interrogation and reply control commands, issued by channel management, and by a stream of DABS reply data blocks, issuing from the DABS reply processor. When a target leaves the beam, a record of channel activity and downlink message data is passed on to the other software functions. Channel management is also in contact with the real-time clock and the antenna azimuth register.

Channel management is organized into five subfunctions, as follows:

1) Channel control
2) Roll-call scheduling
3) Transaction preparation
4) Target list update
5) Transaction update

Parts I and II of this report are devoted to the first two subfunctions, respectively, and discussions of the remaining three are combined in Part III. The interaction of the five subfunctions is summarized in Figure 2, which illustrates the data flow (solid lines) through processing functions and data buffers, and control (dotted lines) emanating from channel control. The remainder of this introduction consists of brief summaries of the operation of each of the channel management subfunctions.

Channel control is the heart of the channel management function. It monitors the system real-time clock and the antenna pointing direction, seeing to it that all ATCRBS and DABS activities take place at the proper time and in the proper sequence. Channel control regulates the other channel management subfunctions, activating them periodically. Also, channel control manages the flow of control commands to the hardware units and it directs the transfer of DABS reply data blocks from the reply processor to channel management memory.

At regular intervals, channel control directs transaction preparation to provide a list of targets about to enter the beam. Transaction preparation consults the surveillance file which contains predicted position, the pending uplink messages and downlink message requests placed there by data link processing, and control information generated by network management.

If basic surveillance, synchronous service, and various messages are all pending for a target entering the beam, then transaction preparation will determine the number and type of interrogations required to accomplish this task. Each interrogation elicits a definite type of reply, and a particular

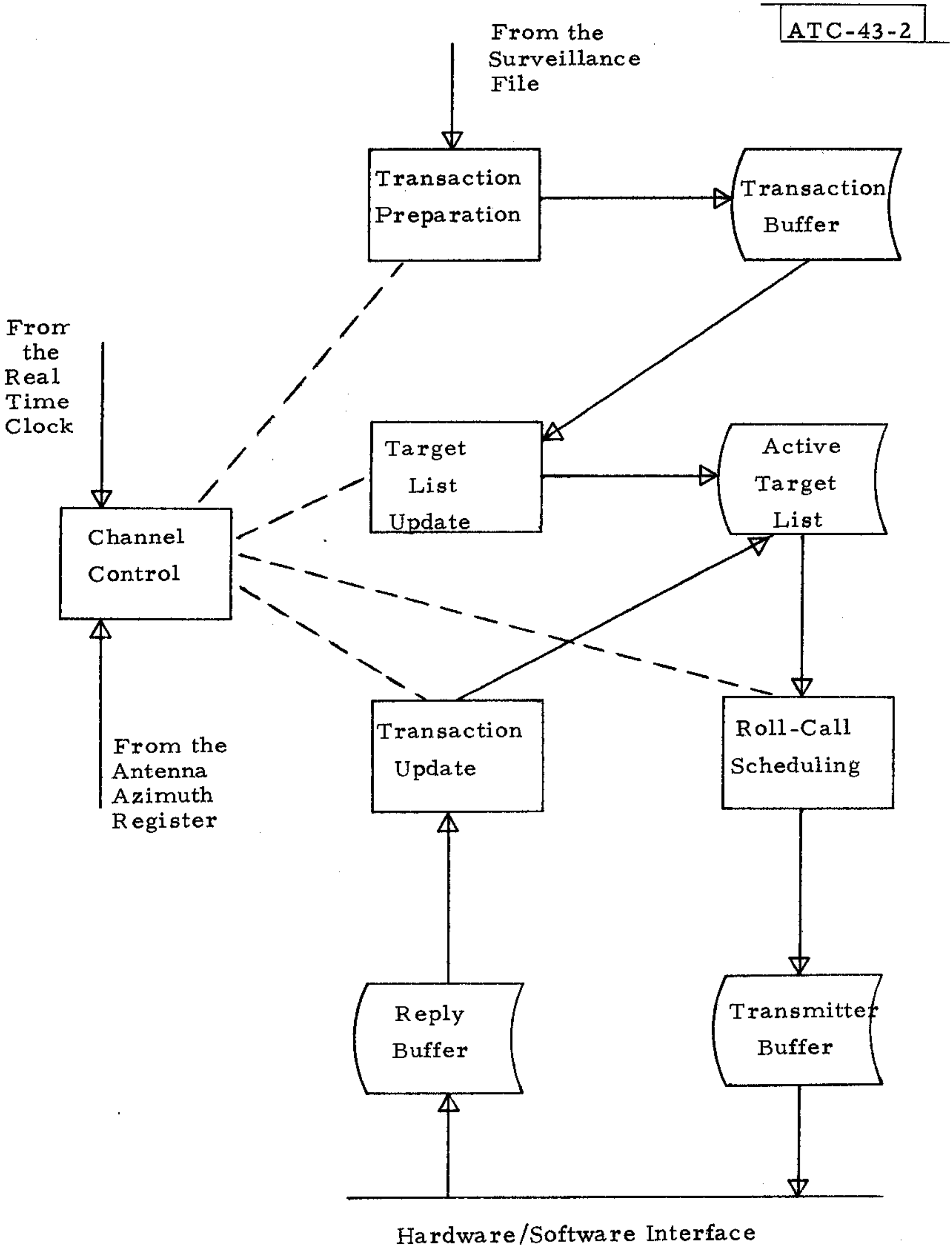


Figure 2. DABS Channel Management Block Diagram

pairing of interrogation and reply types is called a "transaction." A transaction may involve a single interrogation paired with a single reply, multiple interrogations paired with a single reply, or a single interrogation paired with multiple replies.

A number of "standard transactions," which combine the possible message lengths used by the system for single interrogations and replies, are illustrated in Figure 3. Extended length message (ELM) transactions are shown in Figure 4.

In general, more than one transaction is planned for a given target in a given scan. The output of transaction preparation is a list of "target transaction blocks," one for each new target, containing a complete specification of the required set of transactions needed to accomplish all pending surveillance and communications tasks.

The basic DABS sensor utilizes a rotating fixed-beam antenna. This antenna will be of the fan beam type, covering 45° or more in elevation, and only a few degrees in azimuth. Thus a given target is "visible" on the order of one percent of the time, and channel management must execute all the planned transactions to this target while it is within the beam. This is accomplished by maintaining an "active target list," which contains information on every target which is presently within the beam and with transactions still pending execution. The entries on this list are target transaction blocks, supplied in groups by transaction preparation. The active target list is maintained by target list update, which is activated intermittently to accept a new group of target transaction blocks and

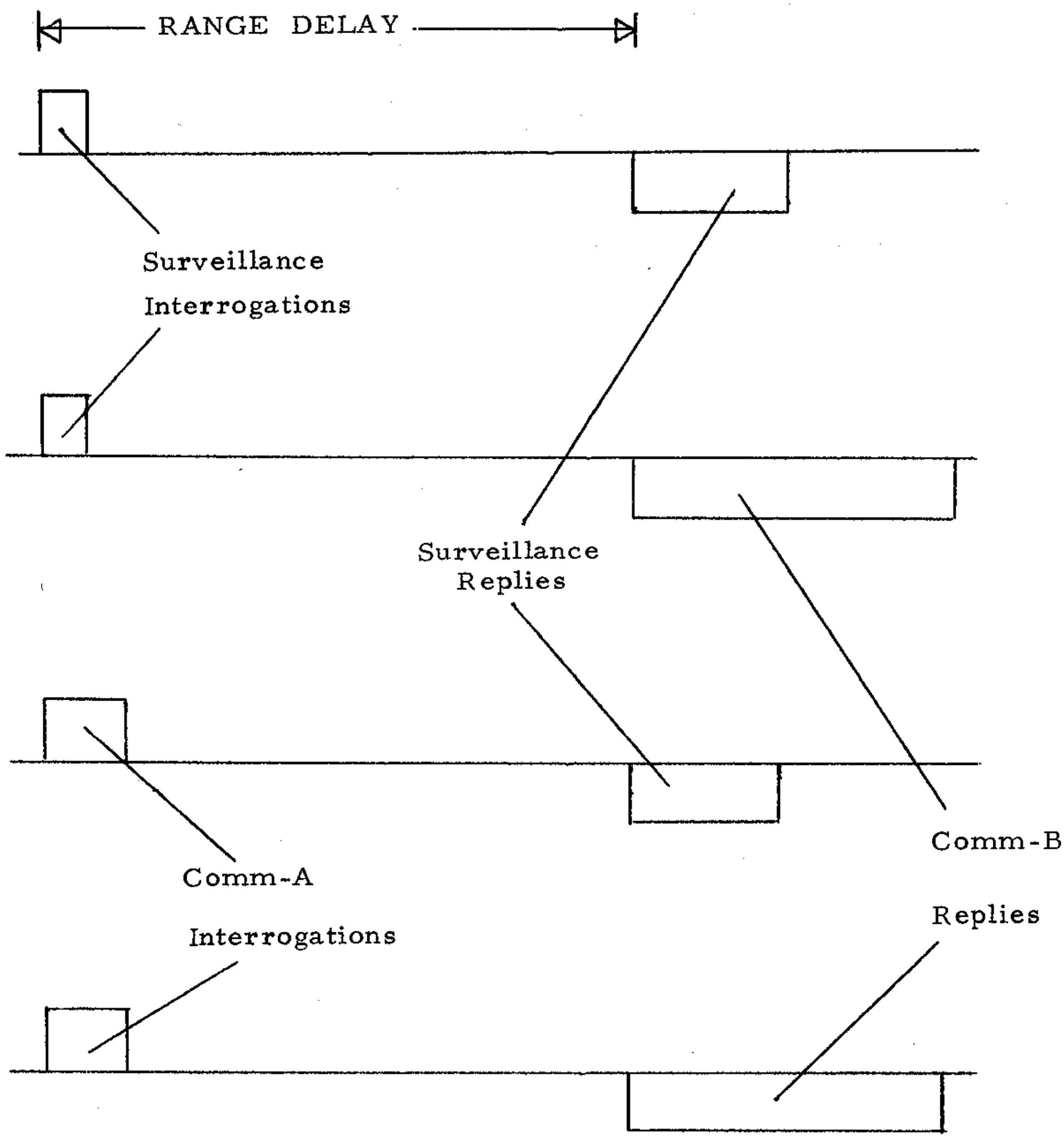


Figure 3. Standard Transactions

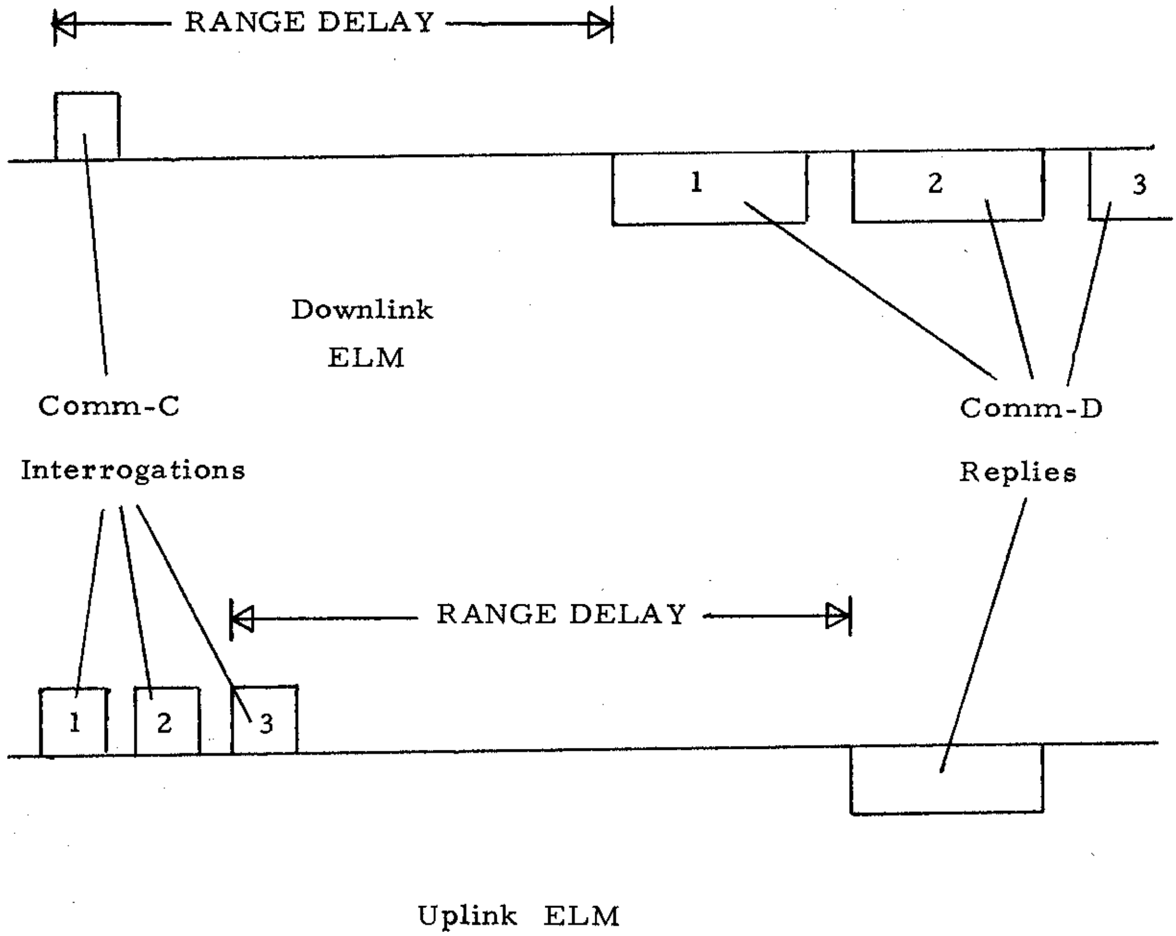


Figure 4. ELM Transactions

merge them with those on the existing list. Target transaction blocks of targets which have either been fully serviced or have fallen behind the beam are removed by target list update at this time. The active target list is organized in order of decreasing target range, for the benefit of roll-call scheduling, and this order is maintained as new targets are added to the list.

Within each target transaction block is a pointer which indicates which transaction is to be executed next for this target. It is the responsibility of roll-call scheduling to produce a set of interrogation and reply times, or "roll-call schedule," which allows the sensor to carry out the indicated transaction for some or all of the targets on the active target list. Intervals of time devoted to this activity are the DABS periods illustrated in Figure 1. The active target list is updated in advance of each DABS period, and during such a period, one or more roll-call schedules are produced.

To produce a schedule, roll-call scheduling will begin with the first (longest-range) target on the list, scheduling an interrogation at an assigned start time of the schedule, and it computes the expected reply arrival time and provides for a suitable listening period. Subsequent targets are scheduled by placing their reply listening periods in sequence and computing the corresponding interrogation times. The process continues until an interrogation, so scheduled, would overlap the first reply. Instead, this interrogation is deferred to start a new "cycle," as the set of interrogation and reply times just produced is called. A cycle is illustrated in

Figure 5, top line. This cycle contains four transactions, and would be part of a roll-call schedule which consists of several such cycles, as illustrated in Figure 6. Since the aircraft on the active target list are in various stages of completion, with respect to DABS activity, each one is likely to be represented on a given schedule by a different kind of transaction. The cycle of Figure 5, top line, illustrates this feature by including long and short interrogations, coupled with long and short replies.

The cycle also includes one synchronized transaction, shown cross-hatched, which differs from the others in that the reply is separated from its neighbors by time buffers. These buffers are required for synchronized transactions in connection with subepoch reply timing and the prevention of interference on the air-to-air link. The other cycles shown in Figure 5 illustrate the inclusion of uplink and downlink ELM transactions.

Roll-call scheduling is activated by channel control, which provides the actual time at which the schedule is to start and the remaining time available in the DABS period. If there is not enough time available to schedule the current transaction for each target on the list, then certain priority rules are exercised in the allocation of available time.

All replies are examined by transaction update and, if a transaction is successful, this subfunction will modify the target transaction block so that the next planned transaction is pending execution. If there are no remaining transactions, the target transaction block is flagged for removal from the active target list. If a transaction fails, transaction update will arrange for it to be repeated. For targets newly entering the beam, the

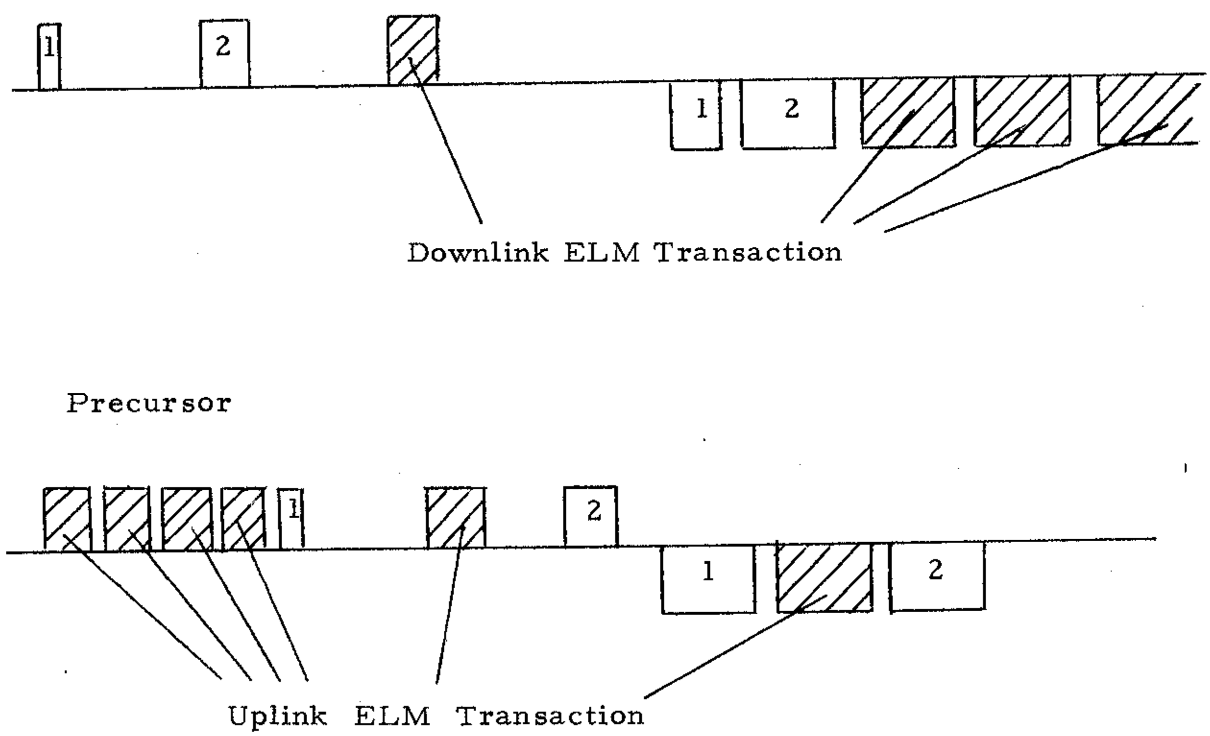


Figure 5. Schedule Cycles

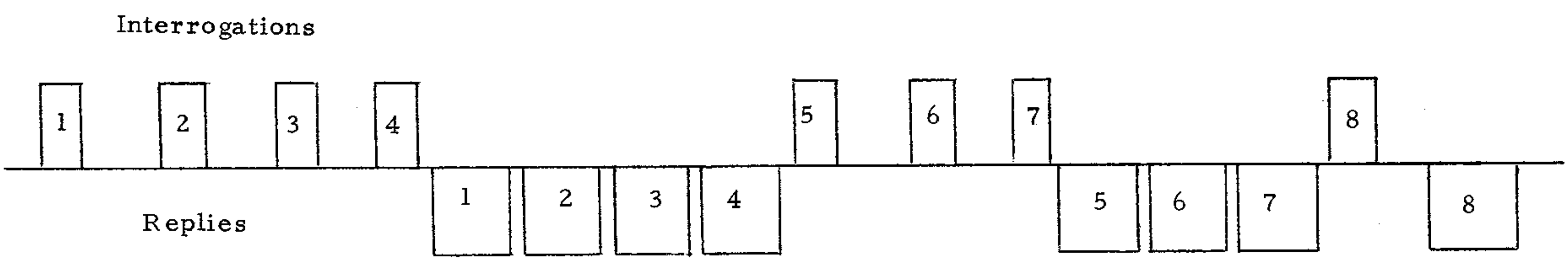


Figure 6. A Roll-Call Schedule Containing Three Cycles

number of attempts to execute the first transaction is limited, and re-interrogations will be made in a special high-power mode. This procedure is also regulated by transaction update. In certain cases, the presence of flags in a DABS reply will cause transaction update to arrange for an extra interrogation, not planned in advance of the target's entry into the beam. This feature is used to service pilot-originated requests to transfer messages or emergency indicators from the aircraft to the ground.

The DABS sensor is specified in detail in Reference 1. Although this reference (the Engineering Requirement for the DABS sensor) is not generally available, cross-referencing to it has been included here for the benefit of some readers. This description given here is self-contained and complete to a considerable level of detail, however, and does not depend upon access to the Engineering Requirement.

Reference 2 is generally available, and it covers the signal formats and data link protocols associated with them.

PART I

CHANNEL CONTROL

I-1. Overview

Certain basic concepts and definitions are introduced here followed by brief summaries of the operations of the four subfunctions controlled by channel control.

The DABS system utilizes a multiplicity of interrogation and reply types. There are two interrogation signal durations (short and long) and two reply signal durations (short and long). Numerical values, in microseconds, are given in Table 1.

Table 1. Signal durations.

	Short	Long
Interrogation	18.5	32.5
Reply	64.0	120.0

As described in Reference 2, there are several message types for each of the signals whose durations are given in Table 1. Particular message types are used to accomplish various surveillance and communications tasks, usually combining position determination and message delivery. A particular pairing of interrogation and reply message types is called a "transaction." The manner in which transactions are chosen to accomplish DABS surveillance and communications objectives is described in Part III of this report.

A "roll-call schedule" is a time-ordered list of interrogation times and corresponding reply listening periods, which together represent a schedule of transactions. One or more roll-call schedules are computed and executed during each period of time in which the RF channel is devoted to DABS roll-call activity.

The transaction preparation subfunction produces a target transaction block for each target before the target enters the antenna beam. The basis of this activity is information contained in the surveillance file, which includes the so-called "active message lists" of pending uplink messages and downlink message requests. The surveillance file contains predicted position and various control bits, updated by surveillance processing and network management. The active message lists are maintained by data link processing. The surveillance file is azimuth-ordered and transaction preparation is given a "new azimuth limit" each time it is activated. Transaction preparation will retrieve targets with predicted azimuths between this new azimuth limit and the previous azimuth limit from the surveillance file. Target transaction blocks are thus produced in groups, ready for merging into the active target list. Within each block, one transaction will be designated for execution.

The target list update function is activated intermittently along with transaction preparation. Target transaction blocks generated previously are merged into the active target list, preserving the ordering on predicted range, by target list update. This subfunction also removes targets (i.e., it deletes the target transaction blocks) from the active target list if they have no further pending transactions or if they have fallen behind the beam. Removed target transaction blocks constitute the "released target list," which is used as input for the next cycle of surveillance and data link processing.

The active target list is updated in advance of each period of DABS roll-call activity. The roll-call scheduling function is called upon next to produce a roll-call schedule from the active target list. Certain priority rules are exercised by this subfunction so that the most important transactions are executed in the limited time that may be available. No target is represented by more than one transaction in any given schedule. After the replies from one schedule are received and processed, another schedule is formed within the same roll-call period until the time available is insufficient for another schedule.

After each schedule is computed, transaction update is activated to assess the reply data. If a transaction is completed successfully, the corresponding target transaction block will be modified so that the next transaction planned by transaction preparation can be executed. In certain cases, new transactions are generated and inserted by transaction update. Unsuccessful transactions are simply repeated. If no further transactions remain pending after a successful reply, the transaction block is marked for deletion from the active target list by the next action of target list update. Whenever transaction update is finished, the active target list is ready for the generation of a new schedule.

The coordination of these activities is the responsibility of channel control. Sections I-2 and I-3 are devoted to descriptions of the time-line structure used by the DABS sensor and the means of communication with the clock, azimuth shaft encoder, transmitter/modulation control unit and the reply processors. Sections I-4 and I-5 concern ATCRBS scheduling and DABS roll-call scheduling, respectively, and Section I-6 contains an illustrative channel control algorithm and flow chart.

I-2. Time Line Structure and Frame Management

The concept of a structured time line is a simple generalization of the notion of a regular or staggered PRF in a radar or beacon system. In a DABS sensor at any given time, the RF channel is committed to either ATCRBS or DABS roll-call activity or else it may be idle. It is assumed that these three cases occur in some periodic, patterned sequence, and this sequence defines the time line structure. The time interval over which this pattern repeats is called the frame duration and the pattern itself is called the frame. An ATCRBS period is an interval of channel time which contains an ATCRBS interrogation and lasts at least as long as the longest ATCRBS listening period in use by the sensor. A DABS period is an interval of channel time devoted to DABS roll-call scheduling activity. A frame may contain one or more ATCRBS periods, one or more DABS periods, and one or more idle periods.

It is intended that the DABS sensor be flexible enough to operate with an arbitrary time line structure so long as the corresponding frame is of reasonable duration and is consistent with surveillance requirements and the antenna in use at the time. These points are discussed in more detail below. To attain this flexibility, the DABS sensor is designed to accept a time line structure as a variable input. Channel control has the responsibility of timing and sequencing all DABS channel activities in accord with the frame in use at any given time. In particular, given a real-time reading, channel control must be able to determine the nature and start time of the next forthcoming period of channel activity. This exercise of this capability may be referred to as "frame management."

A simple technique of frame management is based on the use of a "frame table." This table is a file, the entries of which correspond, in sequence, to the active periods which make up the frame. The frame table has a header which contains the number, K, of frame table entries and the duration of the frame in time. The frame duration, as well as all other time parameters in the frame table, are represented as 16-bit words whose least significant bit represents 16 μsec. The relation of this unit of time to the DABS real-time clock will be explained presently. A particular point in the time line pattern will be designated as the start of a frame. The frame itself is a sequence of contiguous periods. Each active period is represented by a frame table entry. Each entry contains a word which represents the start time of the corresponding period relative to the start of the frame, a type code, and a data field, as illustrated in Table 2.

In Table 2, the relative start time of the kth period in the frame is represented as T(k). The TYPE field identifies ATCRBS, and DABS periods. Also, TYPE identifies the ATCRBS "delay mode", which defines the method by which actual ATCRBS firing time is computed, and the DABS "schedule mode", which determines what kinds of transaction (synchronized, unsynchronized or both) are to be included in the schedule. The character of the DATA field depends upon the value of TYPE (ATCRBS or DABS) and the separate cases are discussed in later sections of this report.

Table 2. Frame table.

Header	K	Frame Duration	
1	T (1)	Type (1)	Data (1)
2	T (2)	Type (2)	Data (2)
3	T (3)	Type (3)	Data (3)
.			
.			
.			
K	T (K)	Type (K)	Data (K)

Channel control always maintains the current values of two quantities: (1) the clock time at which the frame in progress began, and (2) the index, k, of the frame table entry which corresponds to the period currently in progress. The channel control algorithm includes a means of sensing the end (or approaching end) of any period within a frame. In addition, a real-time clock reading is made available to channel control at, or immediately after, such a period end is sensed. Thus, upon sensing a period end, channel control can increment k, determine the relative start time of the next forthcoming period, and predict the real time at which this period will commence. Comparison with the real-time reading provides the time-to-go, or "waiting time," until the start of the next period. If, through some failure, the next period start time is already past, channel control can continue to increment k until its next future responsibility is determined. Whenever the incrementing of k causes it to exceed K, a frame has ended. In this case, k is reset to unity and the frame duration is added to the stored "frame start time," maintaining the currency of this parameter. Subsequent channel activities will be executed at actual times which are first computed relative to the stored frame start time. In this way, channel control maintains the periodic scheduling of frames and periods of activity within the frames.

In order to complete the discussion of frame management, it is necessary to describe the DABS sensor clock, or, more precisely, that portion of the sensor timing subsystem with which channel management interacts. A more complete description of the DABS timing subsystem may be found in References 1 and 3. This clock consists of a precisely-controlled 16 MHz frequency source driving four counters in sequence. The two fastest counters are ordinary eight-bit binary counters, while the two slowest counters reset to zero every 125 input counts. The entire clock resets to zero every 64 sec, since

$$125^2 \times 2^{16} \times 16^{-1} = 64 \times 10^6 \quad .$$

The 64-sec output is, in effect, synchronized to an external time-of-day standard so that all DABS sensor clocks are synchronized. The slower 125-counter is not accessible to the software and the faster 125-counter is implemented as a seven-bit binary counter which clears on the next input count after it reads 124. If we add an eighth bit, always zero, to the left end of the seven-bit field representing the state of the faster 125-counter, then we can represent the state of the accessible portion of the clock as a set of three (eight-bit) bytes. We use the following notation for time, T, as represented by the clock:

$$T = (\ell \,|\, m \,|\, n) \quad .$$

The low-order bytes are represented by integers m and n, which range from zero through 255, while the high-order byte, ℓ, ranges from zero through 124. The clock period, or least significant bit, is called a "range unit," denoted Ru. One Ru represents a time interval of 62.5 nsec. Clearly, the time represented by T in the expression above is

$$T = (65536\,\ell + 256\,m + n)\ \text{Ru}$$

$$= \ell \times 4.096\ \text{msec}$$

$$+ m \times 16\ \mu\text{sec}$$

$$+ n \times 62.5\ \text{nsec}$$

In order to deal with time in 16-bit words, the sensor makes use of an unsymmetrical exchange of time information between hardware and software. When channel control requests real time, or is given real time as part of another hardware-to-software response, only the two higher-order bytes are transferred. Such a time can be represented in the form

$$T = (\ell \,|\, m| \quad ,$$

which uniquely identifies the bytes being transferred. "Time words" of this kind are thought of as being in "time units," which are 16 μsec in magnitude. On the other hand, software-to-hardware commands which reference a time for execution, contain only the two lower-order bytes. Such a time can be uniquely represented in the form

$$T = |m \,|\, n) \quad ,$$

which is still in range units. The ambiguity associated with these two low-order bytes is 4.096 msec. A command containing an execution time in the above form will be executed as soon as the two lowest-order bytes of the real-time clock agree with the bytes transferred in the command. Thus, a special procedure is required when channel control must issue a command to be executed more than 4.096 msec in the future. This procedure is described in Section I-3 in connection with the so-called "pseudo-ATCRBS" interrogation control commands.

Real-time clock readings (time words) in the form

$$T = (\ell|m| ,$$

are ambiguous by 0.512 sec (=125 x 4.096 msec), and this must be considered whenever future times are predicted. For example, suppose that T, as expressed above, represents the clock time of the beginning of the current frame, and the end of the last period within the frame is sensed. Channel control will then add the frame duration expressed as

$$\text{Frame duration} = (\ell_o|m_o| ,$$

to T in order to update this variable. Let the new value be

$$T_1 = (\ell_1|m_1| ,$$

and suppose that ℓ_1 turns out to exceed 124 (say, $\ell_1 = 126$). The arithmetic is valid but the new time is not, and it must be corrected by subtracting the clock period, 0.512 sec, which is equal to 32,000 time units. The clock period is represented in the form

$$\text{Clock period} = (p|0| ,$$

where the low-order byte is all zeros, and the high-order byte is

$$p = [0\ 1\ 1\ 1\ 1\ 1\ 0\ 1].$$

The negative of this period, represented in two's complement form, is

$$-\text{Clock period} = (p' | 0| ,$$

where

$$p' = [1\ 0\ 0\ 0\ 0\ 0\ 1\ 1].$$

It is easy to see that if ℓ_1 lies in the range

$$124 < \ell_1 \leq 127 ,$$

then adding $(p' | 0|$ to $(\ell_1 | m_1|$ will produce the correct future clock reading without overflow. If the computation of T_1 leads to an effective time value in excess of 32767 ($=2^{15} - 1$) time units, then the ℓ_1 -byte will indicate a negative number (i.e., an eight-bit count in excess of 127 in the high-order byte), and overflow will have occurred. As before, the correct time is obtained by adding $(p' | 0|$, and ignoring the overflow which will occur. It is assumed here that the frame duration never exceeds 0.5 sec, hence T_1 will never carry out of the high order byte of the time word.

When the interval of time between a clock reading and the projected time of occurrence of a future event is computed, a large negative result can be obtained if a clock zero occurs between the two times. We test for this case by checking to see if the computed "waiting time" is negative by more than half the clock period. (It is assumed that no waiting period should be as large as half a clock period.) If this is the case, the correct waiting time is obtained by adding the clock period, $(p | 0|$. If the computed waiting time is negative by an amount smaller than half the clock period, it is assumed that the system somehow fell behind in its responsibilities, and the time of occurrence of the "future" event is already past. We assume that such a failure will be detected before channel control falls behind by as much as half the clock period, and recovery is accomplished by moving forward in the frame until an event is found which is actually in the future. A similar situation arises when the system falls behind, but a waiting time appears to be large and positive, again because a clock zero lies between the two times. This is corrected in a similar fashion, as shown in the flow charts of Section I-6.

I-3. Communication with the Hardware

The hardware elements with which channel management interacts are the following:

1. The real-time clock.
2. The antenna azimuth register.
3. The transmitter/modulation control unit.
4. The ATCRBS reply processor.
5. The DABS reply processor.

The nature of the interface between channel management and the hardware elements will obviously affect the internal structure of the channel management software, especially that of channel control. The assumed properties of this interface are described in the present section, and the discussion in subsequent sections is based upon these assumptions. The intention is to give a point of departure, i.e., an example of how channel management can be accomplished in one reasonable configuration.

Channel control has the capability to execute a "read time" command, which results in the transfer of a time word, of the type $(\ell|m|$, from the clock through the interface. The time word, and the notation used in connection with it, has been explained in Section I-2.

A similar capability is the "read azimuth" command, which results in the transfer of an "azimuth word" through the interface. An azimuth word represents the current contents of the antenna azimuth register (a 14-bit register driven by the antenna shaft encoder), right-justified in a 16-bit word. The least significant bit of the antenna azimuth register is called an "azimuth unit," Au. One azimuth unit represents (2^{-14})-th part of a circle, or approximately 0.022°.

Channel management communicates with the transmitter/modulation control unit by sending it "interrogation control commands," which contain the time of the intended interrogation. Time, in an interrogation control command, is expressed in range units, in the form $|m|n)$, in the notation of Section I-2. An eight-bit control field identifies the command as an ATCRBS or DABS interrogation control command, as well as the transmitter power, DABS message length, ATCRBS mode and ATCRBS transmission type (transmit-on-time or transmit-on-trigger). Further details are given in the following Sections. DABS commands contain the uplink interrogation message bits, while ATCRBS commands contain the trigger window and the P_1 delay time (in the transmit-on-trigger mode) and the ATCRBS reply listening interval. The listening interval is transferred from the transmitter/modulation control unit to the ATCRBS reply processor, hence channel management need not communicate with this reply processor directly.

Channel management communicates directly with the DABS reply processor, sending it reply control commands and receiving reply data. There will be one reply control command for each expected DABS roll-call reply. A reply control command contains the DABS address of the aircraft (to be used in message decoding), the earliest expected arrival time of the reply and the reply type. A time window is also included in this command in a special format (five bits, with LSB = 4 μsec) to control the interval within which DABS reply preambles will be accepted. The earliest expected arrival time is a standard 16-bit word in range units, representing time in the form $|m|n)$, just as in the representation of interrogation times.

DABS interrogation and reply control commands are generated by the roll-call scheduling subfunction (see Part II) in blocks corresponding to portions of a roll-call schedule called "cycles." A cycle is a time-ordered string of interrogations, directed to a set of DABS targets followed by a time-ordered string of replies from these same targets. Sometimes a roll-call schedule begins with a so-called "precursor," which may be thought of as a special kind of cycle in which the replies are absent. In general, the number of interrogations in a cycle is not equal to the number of replies, due to the possibility of extended-length downlink messages. In the computation of a schedule cycle, roll-call scheduling proceeds one target at a time, generating both interrogations and reply times for this target before proceeding to the next one. These commands must be transferred across the interface, as data, in the exact sequence in which they are to be executed, which is not the sequence in which they are computed. Therefore, the interrogation control commands and reply control commands are stored, in separate data blocks, until the computation of a cycle is complete. Interrogation control commands require 10 bytes for the short interrogation, and 17 bytes for the long interrogation, while reply control commands of either type require six bytes. These data blocks are then eligible for transfer, by means of block data transfers, to the hardware (interrogations followed by replies), under control of channel control*. As described in detail in Section I-5, roll-call scheduling is activated by channel control. After certain preliminary computations, roll-call scheduling will produce the first cycle (or, perhaps, a precursor), completing the data blocks containing interrogation and reply control commands. It will then formulate the control words for the corresponding block data transfers, and return to channel control. Channel control is thus made aware of the completion of each cycle of the schedule, and it responds by controlling the transfer of the command data blocks and re-activating roll-call scheduling until the schedule is complete. The regulation of the transfer of data blocks of DABS interrogation and reply control commands is discussed in Section I-5.

Channel control also regulates the transfer of DABS reply data* from the DABS reply processor into storage, where this data is available to channel management and the other sensor functions. All of the replies anticipated in a roll-call schedule can be moved in a single block data transfer into a designated section of storage. The transfer is initiated just before the schedule begins execution, so that replies can be moved as soon as they are received. The final output of the schedule computation is a statement of the amount of

*In this regard, the example described here differs slightly from the design specified in Reference 1.

storage required to contain the reply data words expected from the schedule. The DABS reply processor will generate a "reply data block" for each expected reply, even if the reply is not received. Short replies require 13 bytes for the reply data, while long replies require an additional seven bytes in the reply block.

In certain circumstances, channel control requires a notification that a pre-set time has arrived, or that a given interval of time has elapsed since the last reading of the real-time clock. This is accomplished by using the so-called "pseudo-ATCRBS" interrogation control command. Whenever the transmitter/modulation control unit executes an ordinary ATCRBS interrogation control command, it sends a time reading to channel control immediately after the transmission. The time word in this three-byte message (one byte contains a control field) contains the P_1 pulse transmission time, in the same format as a normal response to a "read-time" command. In response to a pseudo-ATCRBS interrogation control command, the time reading is reported when the interrogation time arrives, but no RF transmission occurs. Using this command, channel control can provide itself with an interrupt message at any future time within 4.096 msec of command delivery. In order to provide for an interrupt in the more distant future, channel control issues a pseudo-ATCRBS interrogation control command containing the most recent real-time clock reading as an "interrogation" time. The elapsed time since this last clock reading until the command is unimportant (assuming it will always be less than 4.096 msec), and the process may be repeated until no ambiguity remains and the final pseudo-ATCRBS command is sent containing the desired response time. This technique is used in several circumstances by channel control, as explained in Sections I-4 and I-5.

I-4. ATCRBS Scheduling

The DABS sensor provides surveillance of aircraft equipped only with ATCRBS transponders by transmitting ordinary ATCRBS interrogations (combined with DABS All Call) in some regular pattern. The pattern is fixed by the frame structure in use, which provides one or more ATCRBS periods per frame. Each ATCRBS period will contain one ATCRBS interrogation, hence ATCRBS interrogations will recur at some average repetition frequency. The effective "ATCRBS runlength," or the average number of ATCRBS replies from one target in a scan, depends upon this effective PRF (determined by the frame structure) and the antenna dwell time (determined by the beamwidth and scan rate). The need to provide at least some minimum number of ATCRBS hits on each target acts as a constraint relating frame structure and antenna characteristics. The DABS sensor makes use of an amplitude-comparison monopulse technique to determine aircraft azimuth and also as an aid to the degarbling of overlapped replies. As a result, DABS can operate with a shorter runlength for ATCRBS replies than the conventional ATCRBS system. The lower limit will be set by detection and decoding requirements and the fact that at least two ATCRBS modes (3/A and C) will generally be used. At present, it appears that the required number of hits will lie in the range four to six hits per target per scan.

As an example, suppose that a DABS sensor is equipped with an antenna which has a 3-dB beamwidth of 2.4°, rotating at 15 rpm. If we require an average of four ATCRBS hits within the 3-dB dwell time of 26.7 msec (which will provide at least six hits within the 10-dB beamwidth), then the average

ATCRBS PRF must be at least equal to 150 Hz. If, in this example, the ATCRBS period is 2 msec in duration, then the sensor will devote 30 percent of its RF channel time to ATCRBS, regardless of the detailed frame structure, leaving the remaining time potentially available for DABS roll-call activity. The simplest possible frame structure consistent with the constraints of this example would alternate 2-msec ATCRBS periods with 4.7-msec DABS periods in a 6.7-msec frame.

The DABS sensor has nine ATCRBS modes, listed at the end of this section. The sensor accepts a table of modes as an adjustable input, and it will cycle repeatedly through the table, producing any desired pattern of mode interlace.

Although the ATCRBS periods within a frame are, in general, all of equal duration, the actual listening periods, or time intervals devoted to ATCRBS reply processing, are variable. Each sensor is provided with a map which defines the area within which the sensor is responsible for ATCRBS surveillance. The map is stored in the form of a table called the "ATCRBS/radar range mask," which is also used to define the sensor's area of responsibility for the processing of reports from a collocated primary radar. The table is entered with azimuth and it returns the listening period duration in time units (only one byte is returned). The table has 64 entries, corresponding to an azimuth quantization of 256 azimuth units, or approximately 5.6°, and it is maintained and occasionally changed (in the event of failure of a neighboring sensor) by the sensor network management function. In order to use the table, channel control must predict the approximate azimuth of each forthcoming ATCRBS period based upon actual readings of the antenna azimuth register and the real-time clock. The resulting table entry is incorporated in the ATCRBS interrogation control command as one byte, with maximum value 4.096 msec, which is ultimately used to control the operation of the ATCRBS reply processor.

An important feature of ATCRBS scheduling which remains to be discussed is the possibility of delay of ATCRBS interrogations relative to the start times of ATCRBS periods. The DABS sensor has considerable flexibility in this respect, being capable of generating random or programmed delays, or of responding to external triggers for the timing of ATCRBS interrogations.

The simplest scheme involves no delay at all so that each ATCRBS period begins with an ATCRBS interrogation. If, as in the example given above, the ATCRBS periods occur periodically, then the result will be strictly periodic ATCRBS interrogations. Another simple case involves a fixed delay, assigned to the sensor, or a deterministic, repeating sequence of delays. In the latter case, the delays can be obtained from a separate table which channel control cycles through repeatedly. A pseudo-random delay sequence can also be used with the delays computed sequentially according to a standard algorithm for generating uniformly-distributed random numbers. In this case, the range of random delay (i.e., minimum and maximum values) must be input along with the frame structure.

Another possibility for programmed delays makes the delay a function of azimuth. In this case, regular messages are exchanged among sensors to permit each sensor to maintain knowledge of the approximate pointing directions of the antennas of its neighbors, and ATCRBS interrogation delays are to be computed according to a rule (not specified here) which recognizes the pointing directions of all the sensors in some fixed region. It is expected that programmed ATCRBS interrogation delays may be used to coordinate the activities of a number of sensors which are sharing the surveillance load in a high-density terminal or metroplex environment. This coordination is exercised by synchronizing the real-time clocks of the sensors and operating them with a coordinated set of frame structures and programmed ATCRBS interrogation delay rules. The coordination itself is, in general, transparent to the individual sensor. Reference 3 contains a discussion of synchronization procedures.

In any case, the ATCRBS transmission time is the intended time of transmission of the P_1 pulse. The delay sequence can be coordinated with the ATCRBS mode interlace in use in order to regulate the delay of P_3 relative to the ATCRBS period start time.

In case the DABS sensor is collocated with a (primary) radar, it may be desired to synchronize the ATCRBS interrogations with radar transmissions. The radars operate with a periodic or staggered PRF, and in the present beacon system, ATCRBS interrogations are made to coincide with every second or third radar pulse in a fixed sequence. The DABS sensor will accept radar pre-triggers from such a radar and use them to synchronize its ATCRBS transmissions in the desired manner. The DABS sensor must be provided with a frame structure which corresponds to the radar stagger pattern so that ATCRBS periods will occur at the appropriate intervals. Channel control will send ATCRBS interrogation control commands to the transmitter/modulation control unit with the start times of the ATCRBS periods as interrogation times. The control fields in these commands will indicate "transmit on trigger" and a trigger window is included. The value of the trigger window is obtained from the frame table and it determines the interval during which the transmitter/modulation control unit is responsive to external pre-triggers. The transmitter/modulation control unit will transmit the P_1 pulse of each ATCRBS interrogation at a controlled delay after receipt of the pre-trigger, which is sent directly to the transmitter/modulation control unit. The controlled delay, called the P_1 delay time, is computed by channel control by subtracting the P_1 - P_3 interval for the appropriate ATCRBS mode from an adjustable system parameter, PTRIG, which represents the desired interval from pre-trigger to P_3 transmission time. The timing can be arranged to permit simultaneous display of radar and ATCRBS video for monitoring purposes. The expected transmission time of each ATCRBS interrogation in the pattern is known relative to the start of the frame. One or more of these transmissions will be used, each frame, to correct the stored frame start time to keep the DABS frame synchronized with the radar pulse pattern. The ATCRBS transmission times are known to channel control by way of the usual response of the transmitter/modulation control unit upon transmission of an ATCRBS interrogation.

When channel control determines that the next active period is an ATCRBS period, it will assemble the information required to formulate an ATCRBS interrogation control command. This information consists of:

(a) Interrogation time (P1 pulse)

(b) ATCRBS mode

(c) ATCRBS power

(d) Listening interval

(e) Transmission type (i.e., transmit on time in item (a) or in response to pretrigger).

In the external pretrigger case, the trigger window and the controlled pretrigger/P1 pulse delay are also required.

The delay mode, which identifies the algorithm for determination of interrogation time, is found in the TYPE field of the appropriate frame table entry. The firing delay can thus be computed and added to the ATCRBS period start time to determine interrogation transmission time. If this time is less than 4.096 msec in the future, the interrogation control command will be formulated and sent to the transmitter/modulation control unit for execution. Otherwise, one or more pseudo-ATCRBS commands will be used, as described in Section I-3, to span the time until the actual interrogation control command can be issued.

ATCRBS mode is determined by advancing through the stored mode table. The interlace pattern is a repeating sequence, formed from the modes

(1) ATCRBS Mode A/DABS All-Call

(2) ATCRBS Mode C/DABS All-Call

(3) ATCRBS Mode 2/DABS All-Call

(4) ATCRBS Mode 1

(5) ATCRBS Mode 2

(6) ATCRBS Mode 3/A

(7) ATCRBS Mode B

(8) ATCRBS Mode C

(9) ATCRBS Mode D

Modes numbered (4) through (9) above are intended to be exercised for experimental purposes only.

ATCRBS power is determined by the adjustable system parameter, APWR, which assigns to ATCRBS either of the two power levels used for DABS.

In order to determine the ATCRBS listening interval, channel control must predict the antenna pointing direction for the forthcoming interrogation. The pointing direction of the last ATCRBS interrogation is known, since channel control always reads azimuth when notified of the execution of an ATCRBS interrogation. This azimuth, θ_{old}, and the corresponding time are saved as inputs to the azimuth prediction algorithm. Let WT be the time interval from the last ATCRBS interrogation to the forthcoming one, just computed by channel control. Then the new pointing direction, θ_{new}, is computed according to the formula

$$\theta_{new} = \theta_{old} + WT/SCAN$$

The antenna scan rate parameter, SCAN, is defined and discussed in Section I-5.

Transmission type is contained in the frame table entry (in the TYPE field) along with the trigger windows (in the DATA field), in the external pre-trigger mode. In that mode, channel control will determine the P1 - P3 interval for the forthcoming interrogation (perhaps stored in the mode table) and subtract this value from the system parameter PTRIG in order to find the P1 firing delay required after the arrival of the pre-trigger. This firing delay is included in the ATCRBS interrogation control command.

The actual format of the ATCRBS interrogation control command is specified in Reference 1, Table 3.4.2-1.

I-5. DABS Roll-Call Schedule Management

DABS periods, like ATCRBS periods, recur periodically in a pattern determined by the frame structure in use. DABS periods may all be identical, except (possibly) in duration, or they may cycle through a sequence of "schedule modes," in which case different DABS periods would be devoted to different surveillance/communications objectives. In Section I-1 we described the active target list which contains information on DABS roll-call targets currently in the antenna beam. This list is updated for each DABS period so that it will contain all targets which will be within the beam throughout the DABS period, except for those for which no further transactions are pending (see Part III for full details). Any given target will remain on the active target list for several DABS periods, depending on the relation of the beam dwell time to the frame structure. For example, suppose that a target enters the beam (defined by its 3-dB points) at time T_1 and leaves the beam at T_2. Then this target will qualify for the active target list for each DABS period which lies entirely within the interval (T_1, T_2). Roll-call scheduling does not check azimuth, and a target on the active target list may be scheduled at any time within a DABS period. Hence, a target is not allowed on the list for a particular DABS period unless it will remain within the beam for the entire period. If there are many DABS periods within the beam dwell time, and if each period is short compared to that dwell time, then a target will be on the active target list for almost all of its period of visibility. On the other hand, if the DABS periods are considerably longer, i.e., an appreciable fraction of the beam dwell time, then much of the period of visibility of a target can be lost. The former situation is costly in computation time since the active target list must frequently be updated while the latter situation wastes access time to the target. A compromise between these extremes must be met by the chosen frame structure in relation to the antenna beam and scan rate. This is the DABS analog of the ATCRBS runlength constraint discussed in Section I-4. It appears that a reasonable compromise will result in each target remaining on the active target list for an average of about three DABS periods. If other considerations forced the use of a frame structure much different from this compromise (e.g., a frame with very long DABS periods), then it is obvious that the approach just described would require modification (such as checking azimuth in connection with scheduling).

It is expected that two or more roll-call schedules will be executed in each DABS period. The actual number will vary with azimuth depending upon the instantaneous target load. The average number of schedules per period will depend upon the relation of the frame structure to the average target load and the efficiency of the roll-call scheduling algorithm. If, for example, there are two schedules per period, on the average, and a target remains on the active target list for three DABS periods, then the sensor will have six opportunities to interrogate that target. This number would provide the opportunity to carry out many transactions while the target is within the beam provided the link is reliable, or else it provides an equivalent number of attempts for a target with an unreliable link. The maximum possible number of tries to a non-responding target is clearly a function of actual target load, but the average value of this number is a basic system parameter which must be considered in frame design. In Part I, we are concerned only with certain aspects of the management of the active target list and the authorization of roll-call schedules by channel control. The scheduling algorithm is described in Part II, and the target list update procedure in Part III.

Each DABS period will, in general, be preceded by an ATCRBS period or by an idle period. In either case, the fact that the next active period on the channel will be devoted to DABS roll-call scheduling will become evident to channel control some appreciable time in advance. In the case of an ATCRBS period, channel control has no further responsibility for ATCRBS activity after the interrogation control command is sent. The remaining time is used to update the active target list for the forthcoming DABS period. Target transaction blocks for targets to be added to this list will have been generated by the transaction preparation subfunction at an earlier time. In fact, if D_1 and D_2 stand for two successive DABS periods, then just before the onset of D_1, channel control will authorize the updating of the active target list with previously prepared target transaction blocks and it will also direct transaction preparation to generate new target transaction blocks, in advance, for targets that will be accessible for the first time during period D_2. In order to determine which targets should be processed by transaction preparation, channel control computes an azimuth, θ_{new}, called the "new azimuth limit," and passes this value to transaction preparation. The new azimuth limit is simply the direction of the leading edge of the beam at the start of period D_2, and it delimits the set of newly visible targets that will be visible during all of D_2.

The surveillance file is sorted on azimuth so that transaction preparation can fetch targets between the new azimuth limit just received from channel control and the previous value of this quantity. Target transaction blocks are stored in a buffer (the "transaction buffer") by transaction preparation, and this buffer is emptied by the target list update function whenever the latter function operates. It is intended that transaction preparation complete its task before its output is required by target list update, but the latter function will empty the transaction buffer in any case.

The target list update function, besides merging in the new target transaction blocks, must also remove the target transaction blocks of targets which will fall behind the beam before the end of the forthcoming DABS period (D_1). To accomplish this, channel control passes to target list update an azimuth, θ_{cut}, called the "cutoff azimuth." The cutoff azimuth is the direction of the trailing edge of the beam at the end of the period D_1. The roles of predicted azimuth and measured azimuth in the addition and deletion of targets from the active target list are discussed in the subsequent Parts of this report.

In order to compute θ_{new} and θ_{cut}, channel control must read the real-time clock and the antenna azimuth register, it must determine the time intervals to the end of D_1 and start of D_2, and it must compute the required azimuths, taking account of the scan rate of the antenna in use. To facilitate this computation, the duration of DABS period D_1 and the time interval between the start times of D_1 and D_2 are stored in the DATA field of the frame table entry corresponding to D_1. Suppose that channel control has read the time (T) and the antenna boresight azimuth (θ), and that it has also computed the waiting time, WT, from time T to the start time of period D_1. Then, the two stored quantities can be added to WT to produce WT_1, the waiting time to the end of D_1, and WT_2, the waiting time to start of D_2. The desired azimuths are then calculated by means of the formulas:

$$\theta_{cut} = \theta - \theta_{half} + (WT_1/SCAN)$$

$$\theta_{new} = \theta + \theta_{half} + (WT_2/SCAN).$$

In these formulas, θ_{half} is an adjustable parameter representing the effective half-beamwidth and SCAN represents the inverse scan rate of the antenna. The nominal value of θ_{half} is one-half the 3-dB beamwidth but experience with the sensor may suggest a somewhat different value. In any case, θ_{half} must be adjustable to accommodate a variety of antennas.

The parameter SCAN requires some further explanation. It would be more natural to multiply the time interval by a scan rate parameter but in the units natural for the sensor, i.e., azimuth units per time unit, all reasonable scan rates are very small numbers. Since high accuracy is not required in these calculations, it seems reasonable to divide by "inverse scan rate," and approximate the divisor by a power of two, so that the division can be accomplished by a right shift. One minute per revolution corresponds to 228.88... time units per azimuth unit, hence we have the values given in Table 3.

Table 3. SCAN Parameter Correspondence

Scan Rate (rpm)	SCAN Parameter
15	15.259 ...
6	38.147 ...
5	45.776 ...

We therefore use the approximate value 16 for SCAN to represent a rotation rate of 15 rpm and put SCAN = 32 for the other two cases. The resulting azimuth errors are shown in Table 4 for a 10 msec waiting time, which should be a representative value.

Table 4. SCAN Errors

Scan Rate (rpm)	Error (WT = 10 msec)
15	0.0337°
6	0.0691°
5	0.1292°

The computation of θ_{new} and θ_{cut} and the activation of transaction preparation and target list update, are preliminaries to the main business of channel control in connection with DABS periods, which is the authorization of roll-call schedules. Channel control will authorize schedules, iteratively, until time runs out in the DABS period. The first schedule of a DABS period (sometimes called the "prime schedule") is authorized after the target list update function reports completion of its task to channel control. Subsequent schedules are authorized after channel control receives notification of completion by the transaction update subfunction. This latter function is activated after the computation of each schedule. Transaction update assesses the results of each schedule and modifies the active target list according to the success or failure of each transaction. Transaction update cannot add or delete targets from the list, but it does flag targets with no further pending transactions, so that these will be ignored during subsequent scheduling activity within the DABS period.

Before authorizing any schedule, channel control will determine the schedule start time and the time remaining from this moment until the end of the DABS period. This interval is called the "available time," and a schedule is authorized only if the available time exceeds a certain minimum value. The minimum value is an adjustable system parameter and represents at least the 196 μsec required to schedule a single target at zero range. The schedule start time is determined by adding a fixed processing delay interval to the real-time clock reading which is obtained by channel control upon receipt of a completion notice from either target list update or transaction update. In the case of a prime schedule, the schedule start time must not precede the beginning of the DABS period.

When activated for a schedule other than the prime schedule, the roll-call scheduling subfunction makes an independent check of the feasibility of producing a schedule for the period of time remaining, taking account of the actual ranges of the targets on the active target list. Once computation of the schedule begins, roll-call scheduling will report completion of each cycle to channel control, having prepared all the relevant commands and formulated the control words for the block data transfer of these data. Roll-call scheduling is reactivated after each cycle until the last, which includes a completion notice to channel control and an indication of the storage required for all of the scheduled replies. As soon as the schedule computation is complete, channel control activates transaction update, whose completion in turn causes the entire cyclic process to repeat until the time available is no longer sufficient to support a schedule.

It remains to discuss the timing relationships between the computational tasks, mentioned above, and the activity on the RF channel. As we have pointed out already, channel control will begin preparation for a DABS period during a preceding ATCRBS period (there may also be an idle period between the ATCRBS period and the DABS period in question). This means that channel control will have at least one millisecond (probably more) of computation time before the DABS period begins. The preparatory tasks of computation of θ_{new} and θ_{cut}, and the activation of transaction preparation and target list update will require very little time. It is assumed here that one computer is dedicated to the channel management tasks of channel control, target list update, roll-call scheduling, and transaction update. Transaction preparation, being much less time-critical, will be carried out in the so-called "scan-to-scan" processor as described in Reference 1 (Section 3.3). Channel control must therefore only wait until the completion of the target list update task which is expected to occupy only a fraction of the available computation time preceding the DABS period.

By the time the DABS period starts, one or more complete cycles will have been generated by roll-call scheduling. Channel control will be cognizant of the generation of these cycles and of the fact that the interrogation and reply control commands are ready to be transferred to the hardware. In fact, the block data transfer control words will already have been formatted. Channel control is responsible for controlling the transfer of these commands to the hardware and it must wait until ATCRBS reply processing is finished before it initiates any new data transfers. Depending on the frame structure, for each DABS period, there will be a "schedule release time," which represents the earliest moment when DABS commands (i.e., interrogation and reply control commands for a DABS roll-call schedule) can be released prior to that DABS period. The time interval by which this release time precedes the DABS period will be stored in the DATA field of the DABS period entry in the frame table. As soon as channel control commences preparation for a DABS period, it arranges to be interrupted when the schedule release time arrives. This is accomplished by the use of one or more pseudo-ATCRBS commands, as described in Section I-3. When the schedule release time arrives, channel control first sends an open-ended block data transfer ("read") command to the DABS reply processor, enabling the hardware-to-software transfer of DABS reply data blocks, when they arrive, and assigning a starting location in memory for the storage of these replies. Later, when the schedule computation is complete and the total reply storage area is known, channel control will amend the "read" command to delimit the length of the transfer.

After issuing the "read" command, channel control will send a "write" command directing the transfer of the block of interrogation control commands of the first cycle (or precursor) of the roll-call schedule. The individual words (interrogation commands) of the data block will be transferred, one at a time, under hardware control. As each "write" command is executed, i.e., as each block data transfer is completed, channel control will be notified and it will respond with another "write" command until no more are available. Concurrently, roll-call scheduling is generating new data to be transferred until schedule computation is complete. In the case of the prime schedule, schedule computation may be completed before the start of the DABS period. Even if this is not the case, schedule computation will be completed well before the receipt of the last scheduled reply by the hardware.

As soon as the schedule computation is complete, channel control will activate the transaction update subfunction. If no replies have as yet been received, as may be the case for the prime schedule, transaction update will simply await their arrival. If replies are already in storage, and arriving still, transaction update will begin its processing of replies, and eventually catch up, so that before the end of the actual schedule, transaction update will be processing the replies directly upon their arrival. It is expected that transaction update will require no more computation time, per reply, than the reply duration itself, and that it can catch up on any backlog during the time intervals when interrogations are being transmitted.

When transaction update is finished with the last reply of a schedule, the entire active target list will have been prepared for the next schedule. Upon receipt of notice of completion by transaction update, channel control will determine the present real time and add it to a processing delay interval in order to fix a schedule start time for the next roll-call schedule. After checking the available time for the feasibility of further scheduling, channel control will again activate roll-call scheduling. For all schedules after the prime schedule, channel control can release "read" and "write" commands without delay, since all previously issued commands will have been executed already. With schedules after the prime schedule, the schedule computation will be carried out concurrently with schedule execution, but this computation will be completed before the last scheduled reply is received, since it is anticipated that schedule computation time will be significantly less than schedule execution time. Thus, transaction update will always be activated while the schedule execution is in progress, and this subfunction should be able to catch up to the arriving replies before the receipt of the last one.

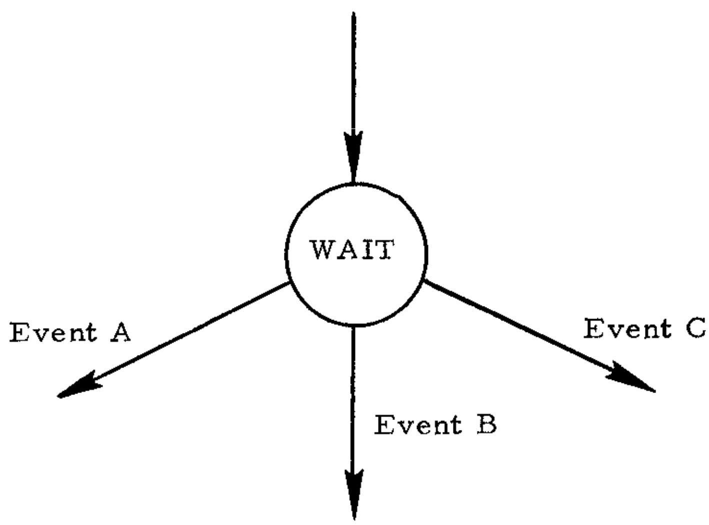
The DABS schedule mode, obtained from the frame table, is simply passed by channel control to roll-call scheduling. This mode determines which kinds of transactions may or may not be included in the schedule. The application of this information to the "target qualification" procedure will be explained in Part II. However, one particular schedule mode (scheduling without reply processing), causes a modification of the procedures described above. In this case replies are to be ignored by the sensor and no reply control commands will be produced by roll-call scheduling. Only one schedule is produced in such a DABS period. After schedule computation is complete, channel control will activate transaction update, passing to that subfunction the fact that no reply processing is to take place. Transaction update will respond by reacting as though all scheduled transactions were successful and it will then return control to channel control, perhaps before the end of the schedule. Although replies are not processed, a schedule of this kind is considered to last until the last reply arrives at the sensor. Upon receiving the completion notice from transaction update, channel control will consider the DABS period activity finished and proceed as though insufficient time were available for another schedule.

The general design described here is expected to be more than adequate, in regard to surveillance and communication capacity, for the initial stages of DABS inplementation. The design leaves room for improvement in many areas and it is expected that a continual evolution of algorithms will take place in the future, guided by experience with the system and the needs of related programs, such as IPC. One specific design detail which reduces channel capacity in the algorithm described here is the processing delay which occurs between schedules. There are several ways to avoid this, each of which requires roll-call scheduling to begin schedule generation before the active target list is fully prepared for the next schedule. The main issue here relates to the allocation of schedule time to high-priority transactions and in the present design this is not attempted until the results of the previous schedule are all known.

I-6. A Channel Control Algorithm

The operation of channel control can best be summarized by a sequence of flow charts, beginning with a gross overview chart, and continuing with successive expansions of processing blocks. The charts presented here conform to the detailed example of channel control given in the preceding sections. This example was based on specific assumptions regarding the transfer of data between hardware and software, presented in Section I-3. These assumptions affect many of the details evident in the flow charts, and it must be kept in mind that other possibilities exist and may be preferable. The understanding of the general problem made possible by exploring this example should make it easy to implement channel control with other configurations and constraints.

Channel control is basically an infinite loop, entered by means of a start-up procedure. Within the loop, channel control frequently enters an inactive state, to be awakened by the occurrence of one or more events. The action taken by channel control in these cases generally depends upon the nature of the triggering event. The schematic representation given here is a combination flow chart and state diagram. Each inactive period is represented as a WAIT state, entered by channel control and left in response a specific event, as follows:



The arrows leaving the WAIT state lead to other sections of the flow chart (or other WAIT states) just as in the case of a multiple decision branch. From the flow-chart point of view, the WAIT states are decision points in which the decision is dependent upon external triggers instead of internal data. From the state-diagram point of view, each WAIT state is a different state of the system.

The basic task of channel control is to sequence through the active periods of the frame. Whenever ATCRBS or DABS period scheduling is complete, channel control will determine current time and azimuth and determine the nature and start time of the next active period. After a test to verify that the next period start time is still in the future, channel control proceeds with the appropriate activity for the period. Figure 7 is a flow chart of this activity. The processing boxes labelled A, B and C are expanded in subsequent charts.

At all times, channel control maintains the current value of TIMREF, the clock time at which the present frame began, and the current value of k, the index which identifies the present active period within the frame. These quantities are updated every time channel control executes box A. The start-up procedure is essentially an assignment of values of TIMREF and k, followed by a reading of time (T) and antenna azimuth (θ). One possibility is to initialize TIMREF to a future time and arrange to have the system begin executing a frame when this time arrives. In the synchronized mode of operation, the activities of several sensors are coordinated, and the allowable start-up times (frame start times in actual time-of-day) are restricted. Synchronization procedures are discussed in Reference 3.

Figure 8 is a partial expansion of processing box A of Figure 7. The active period index, k, is incremented and tested against the maximum value, K, to see if the old frame is completed. If this is the case, then k is reset to unity and TIMREF is updated. Channel control then computes TNEXT, the start time of the forthcoming active period and WT, the waiting time from T (the last real time reading) to TNEXT. The nature of the DABS timing system complicates these steps slightly, and processing boxes A1, A2 and A3 are accordingly expanded in Figures 9, 10, and 11. The procedures involved are exactly those explained in Section I-2. In Figure 10 the array T(k) contains the active period start times, as shown in Table 2.

ATCRBS period scheduling is straightforward, hence the expansion of processing block B of Figure 7 is a simple list of the steps required to formulate an ATCRBS interrogation control command. The flow chart shown in Figure 12 follows the steps described at the end of Section I-4. The test, "delay required?", implies a check as to whether the interrogation time is less than 4.096 msec in the future or whether a pseudo-ATCRBS command must be used to span the time interval until the actual command can be issued.

Figure 13 is a flow chart of DABS period scheduling, processing box C of Figure 7. The chart is somewhat simplified, but the steps have been explained in Section I-5. The phrase "provide for interrupt at schedule release time" refers to the issuance of one or more pseudo-ATCRBS commands, the last of which provides the desired interrupt. While roll-call scheduling is generating the prime schedule it is repeatedly reactivated by channel control after each cycle is complete. This activity is summarized in Figure 13a by the phrase "channel control accepts notices of roll-call scheduling output." The READ command is the initiation of a block data transfer of reply data from hardware to software, while the WRITE command initiates the transfer of interrogation and reply control command from software to hardware. In Figure 13c, the first test "time for another schedule" is made by channel control, based only on the computed available time. Roll-call scheduling "acknowledgment" is a response indicating whether or not roll-call scheduling will produce a schedule. This response is then used by channel control in the second test. Only WAIT states which are left by the occurrence of a single event are shown in this flow chart, but more complex states (with several triggering events) would arise in the further expansion of the chart.

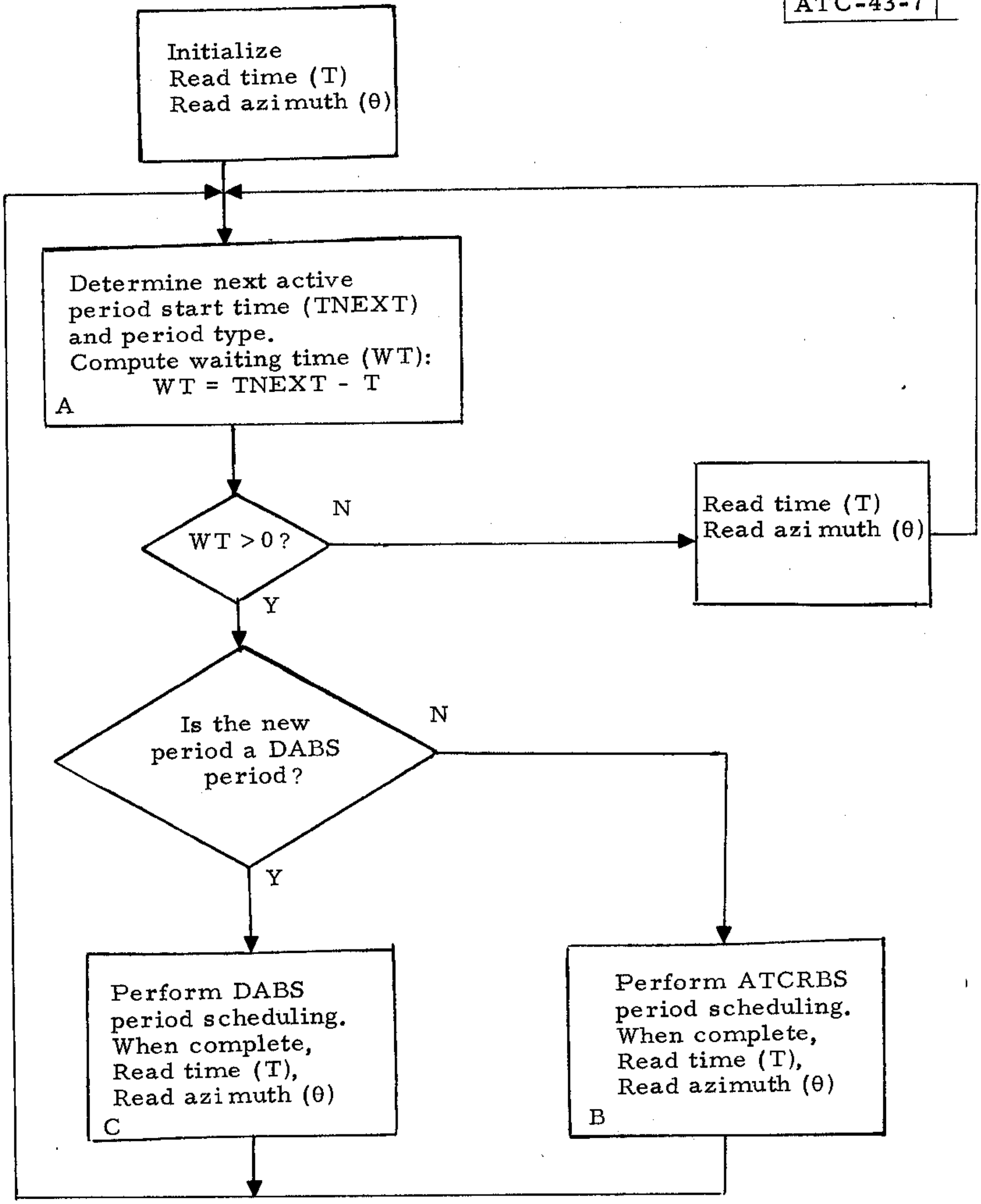


Figure 7. Channel Control

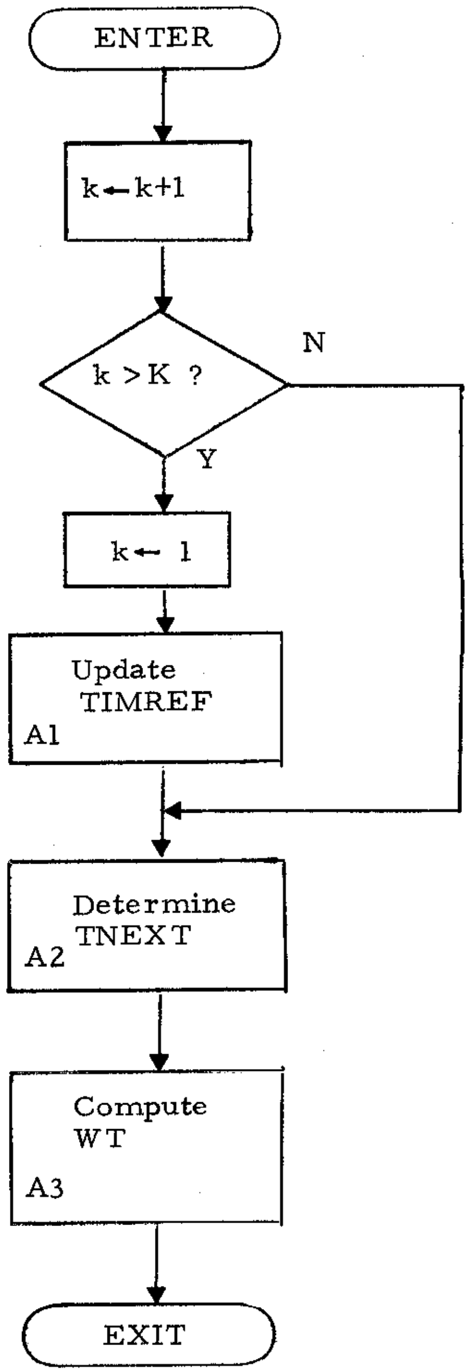


Figure 8. Box A Expansion

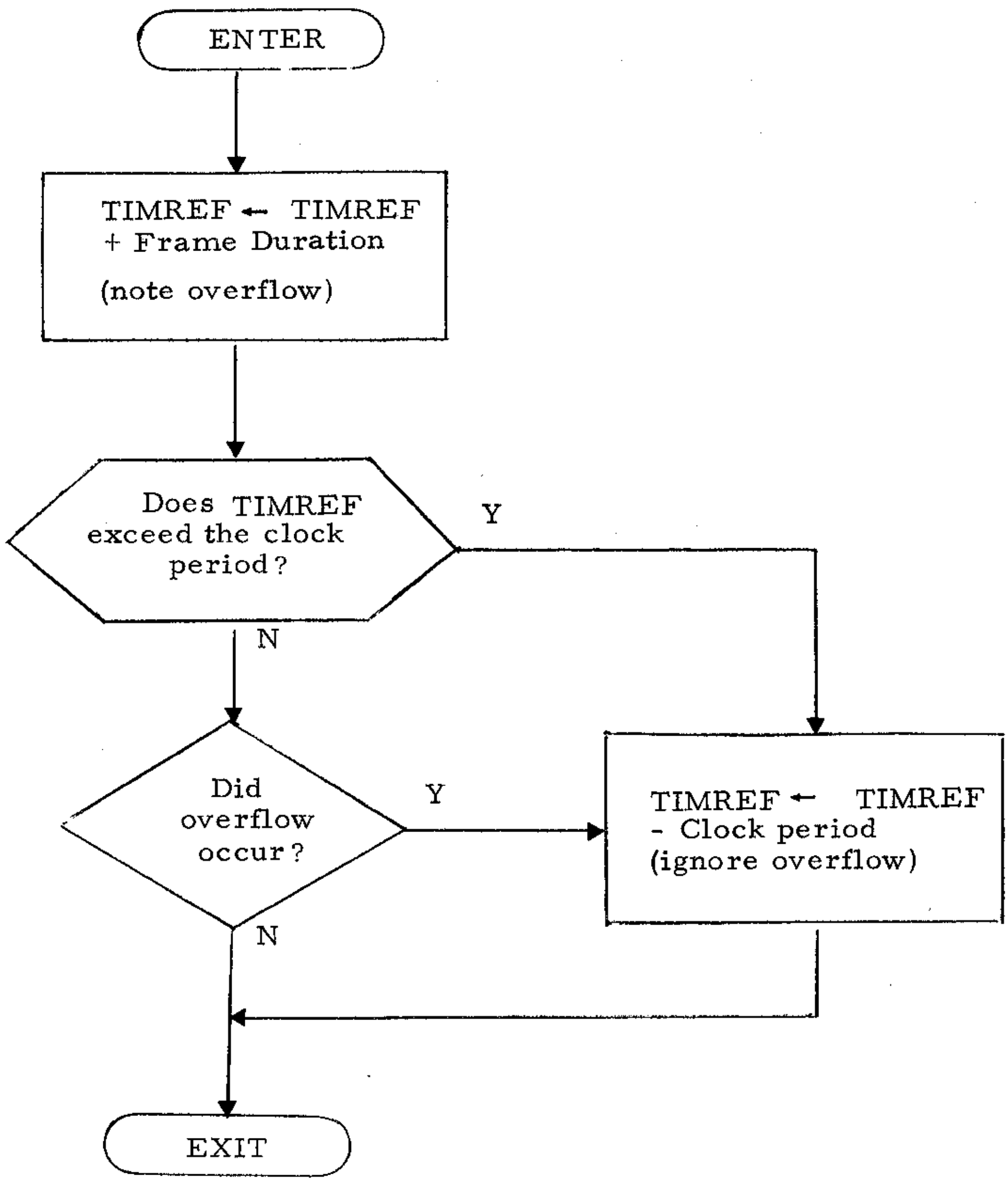


Figure 9. Box A1 Expansion

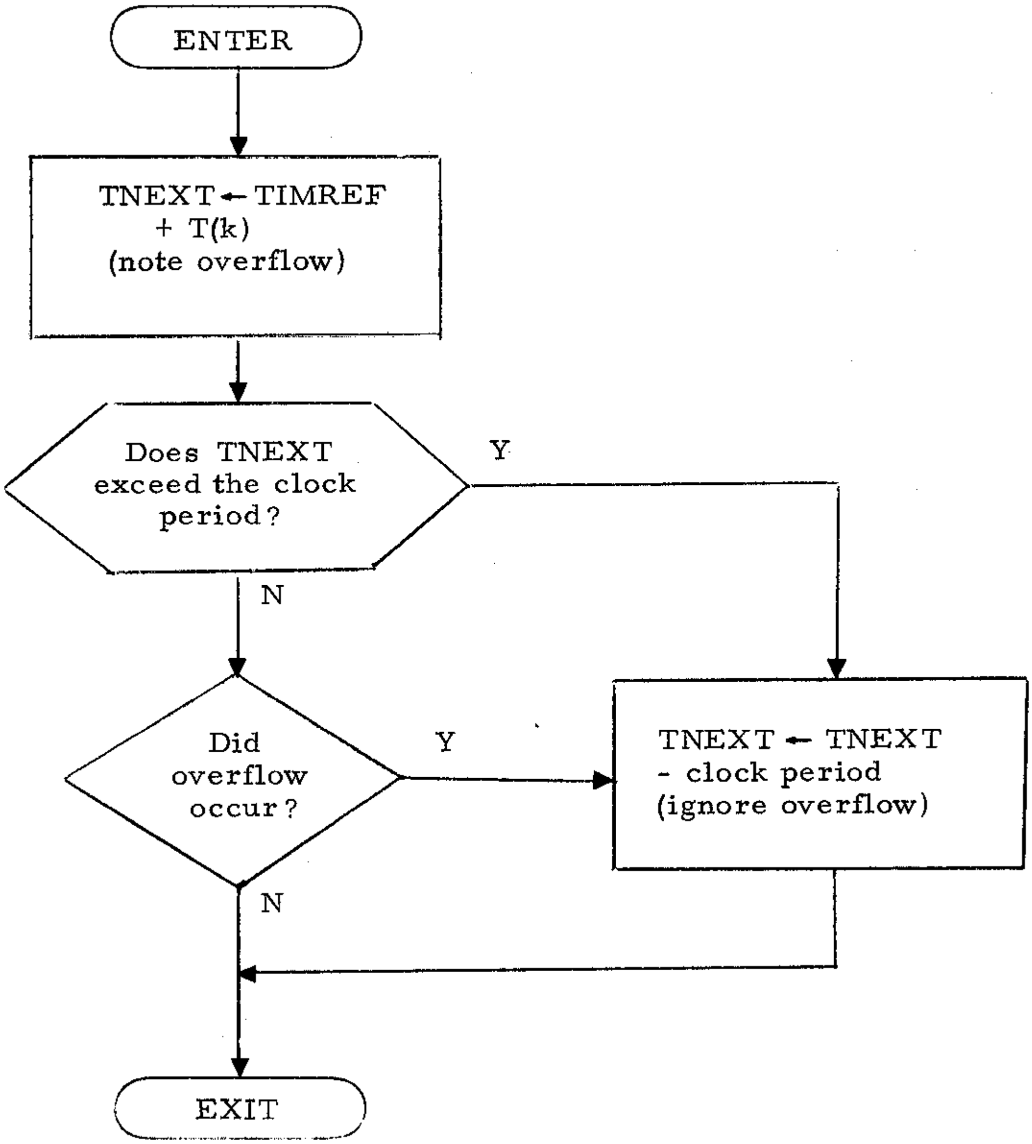


Figure 10. Box A2 Expansion

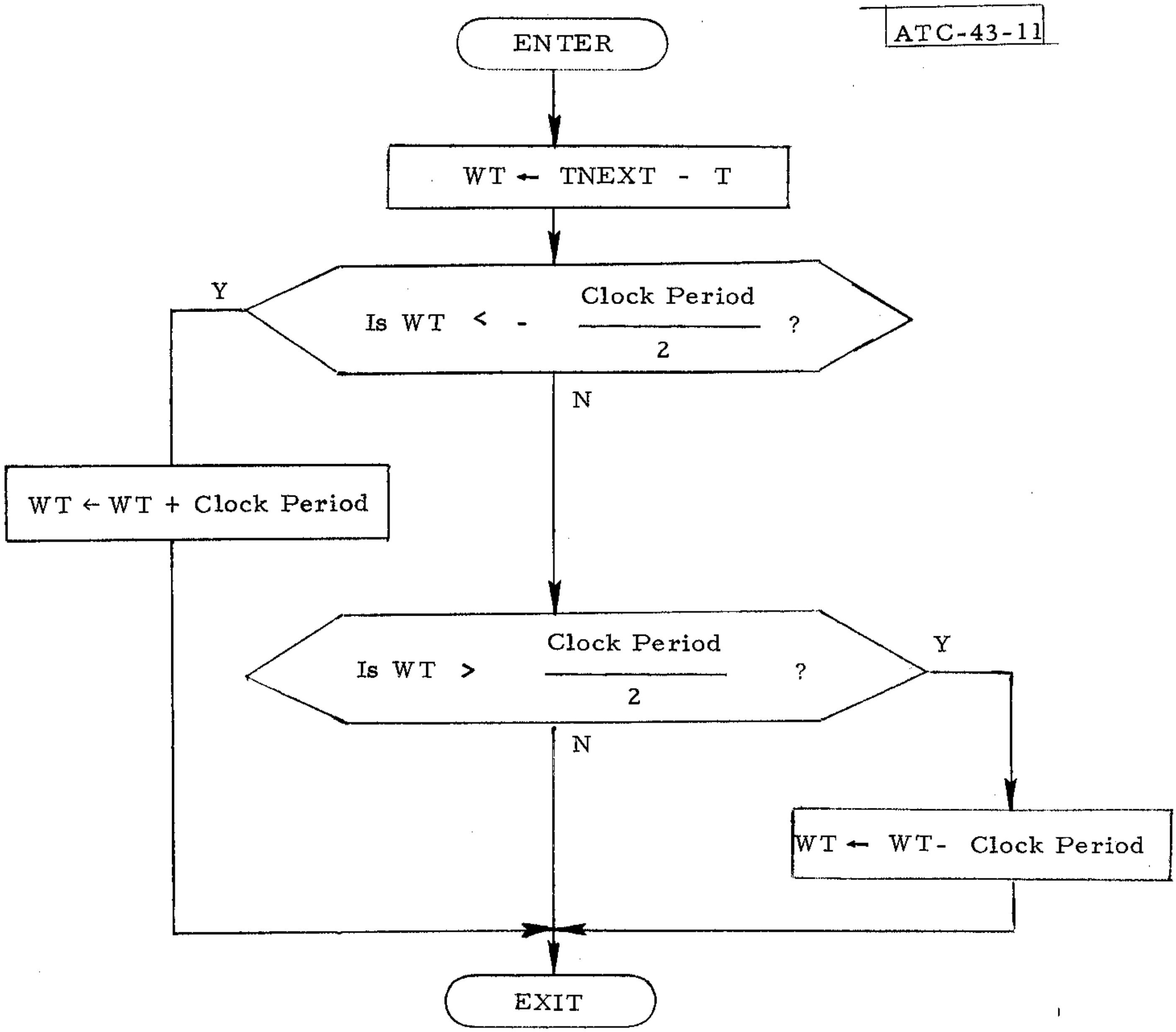


Figure 11. Box A3 Expansion

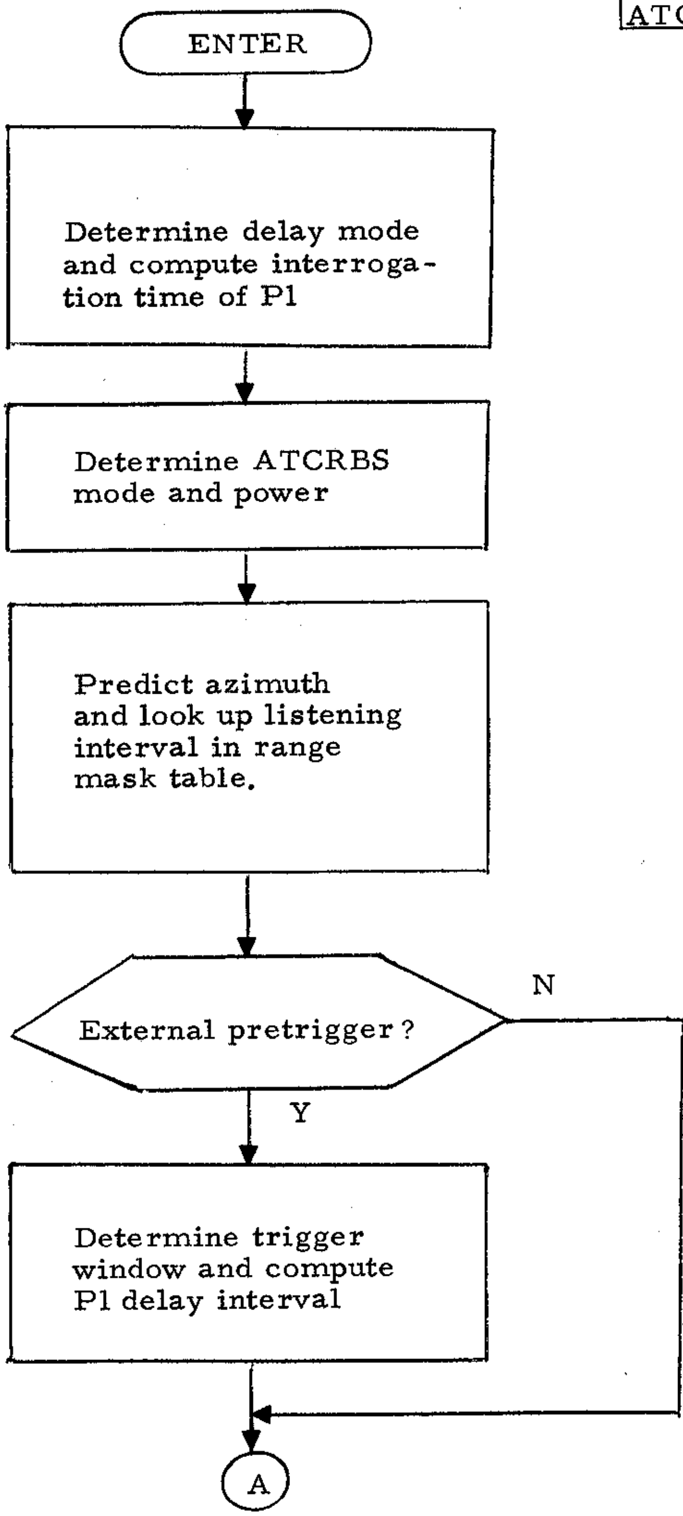


Figure 12a. Box B Expansion (ATCRBS Period Scheduling)

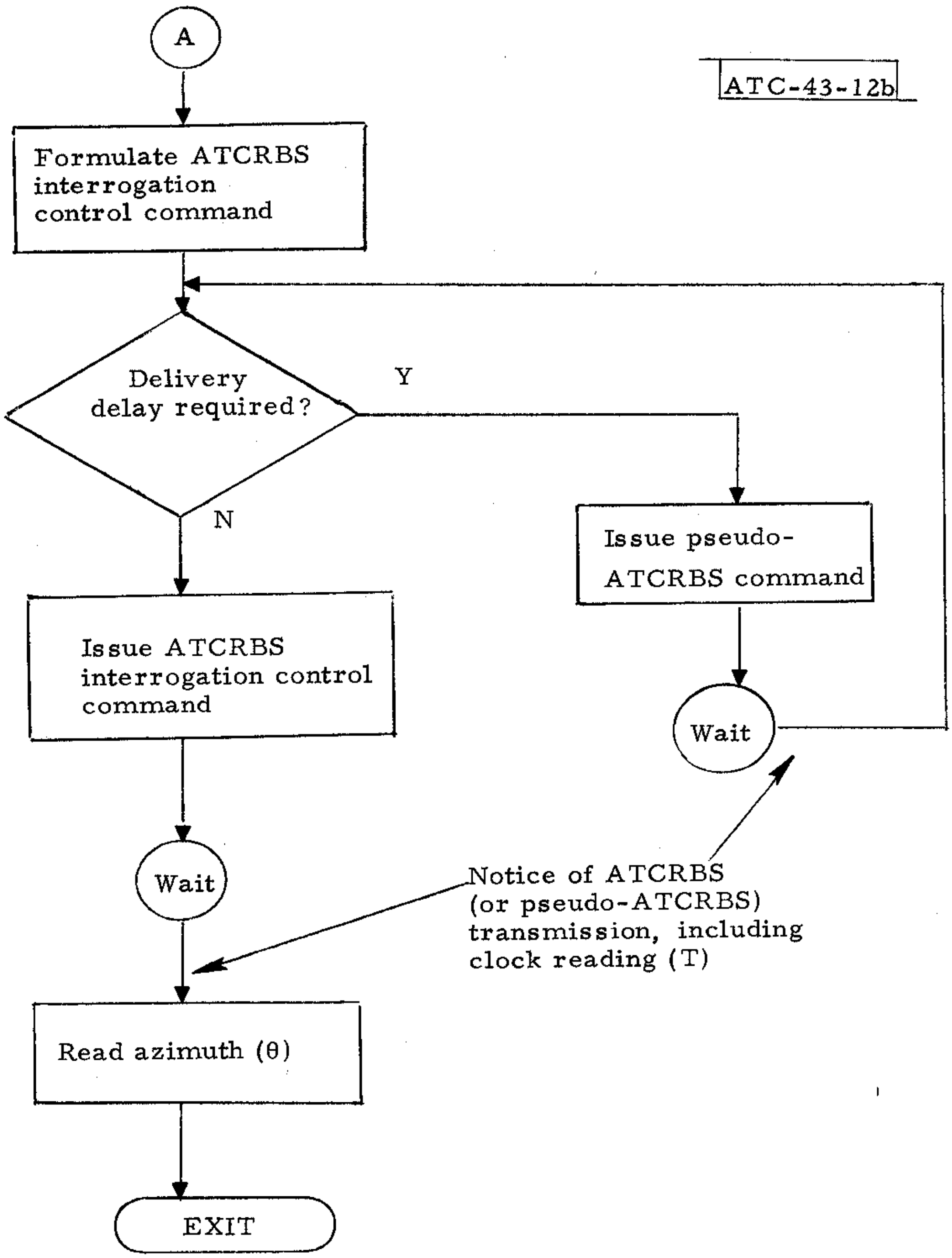


Figure 12b. Box B Expansion (Continued)

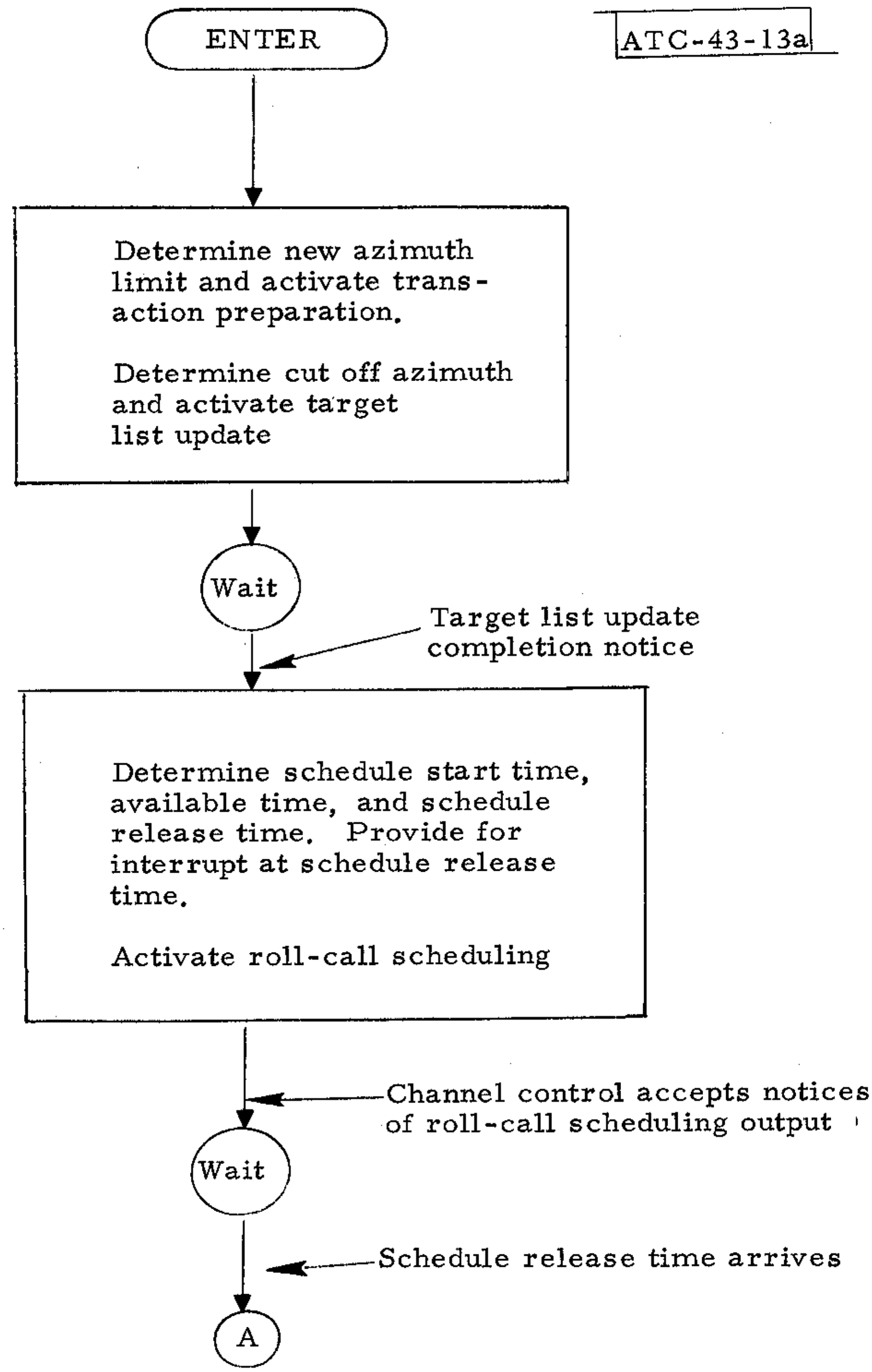


Figure 13a. Box C Expansion (DABS Period Scheduling)

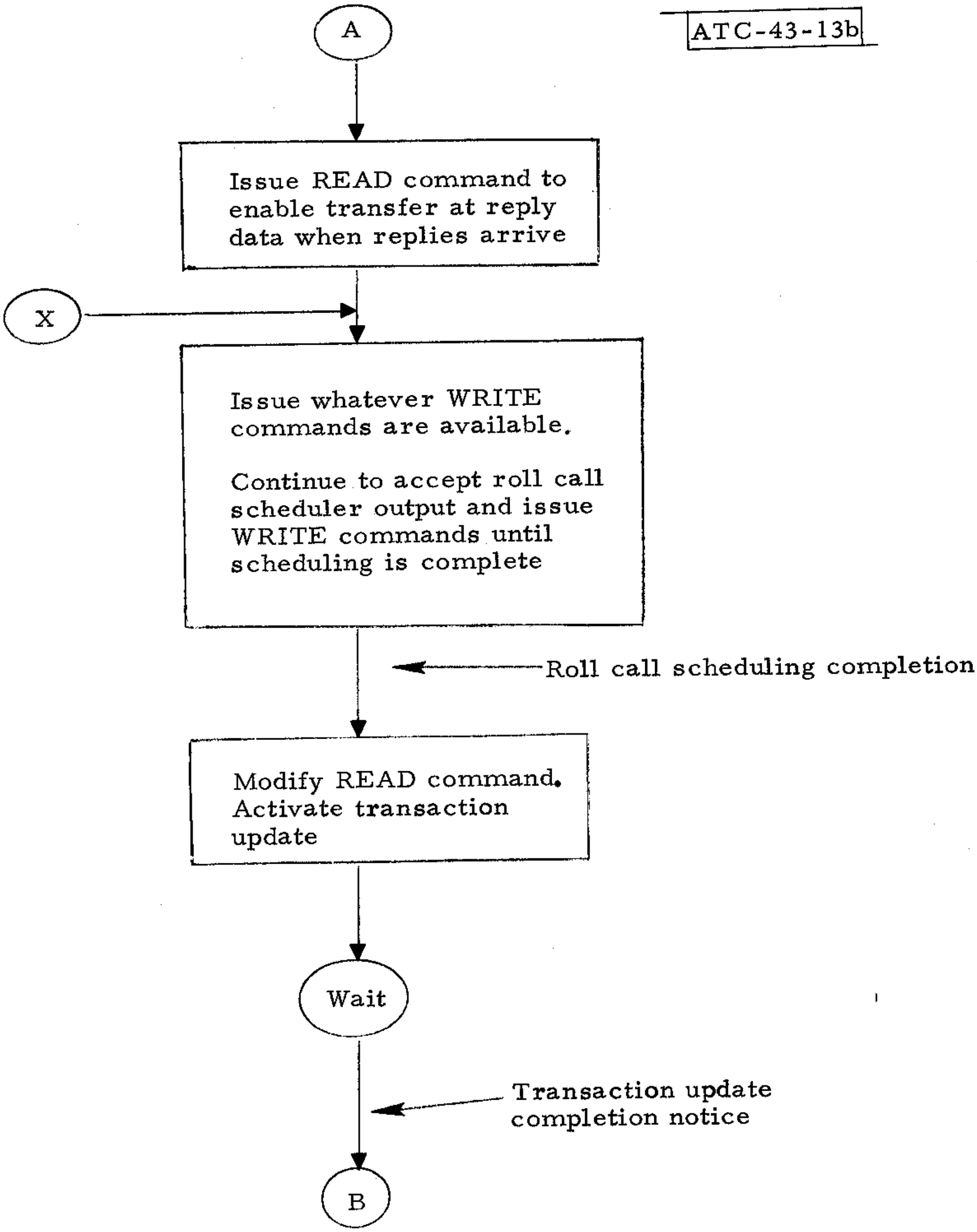


Figure 13b. Box C Expansion (Continued)

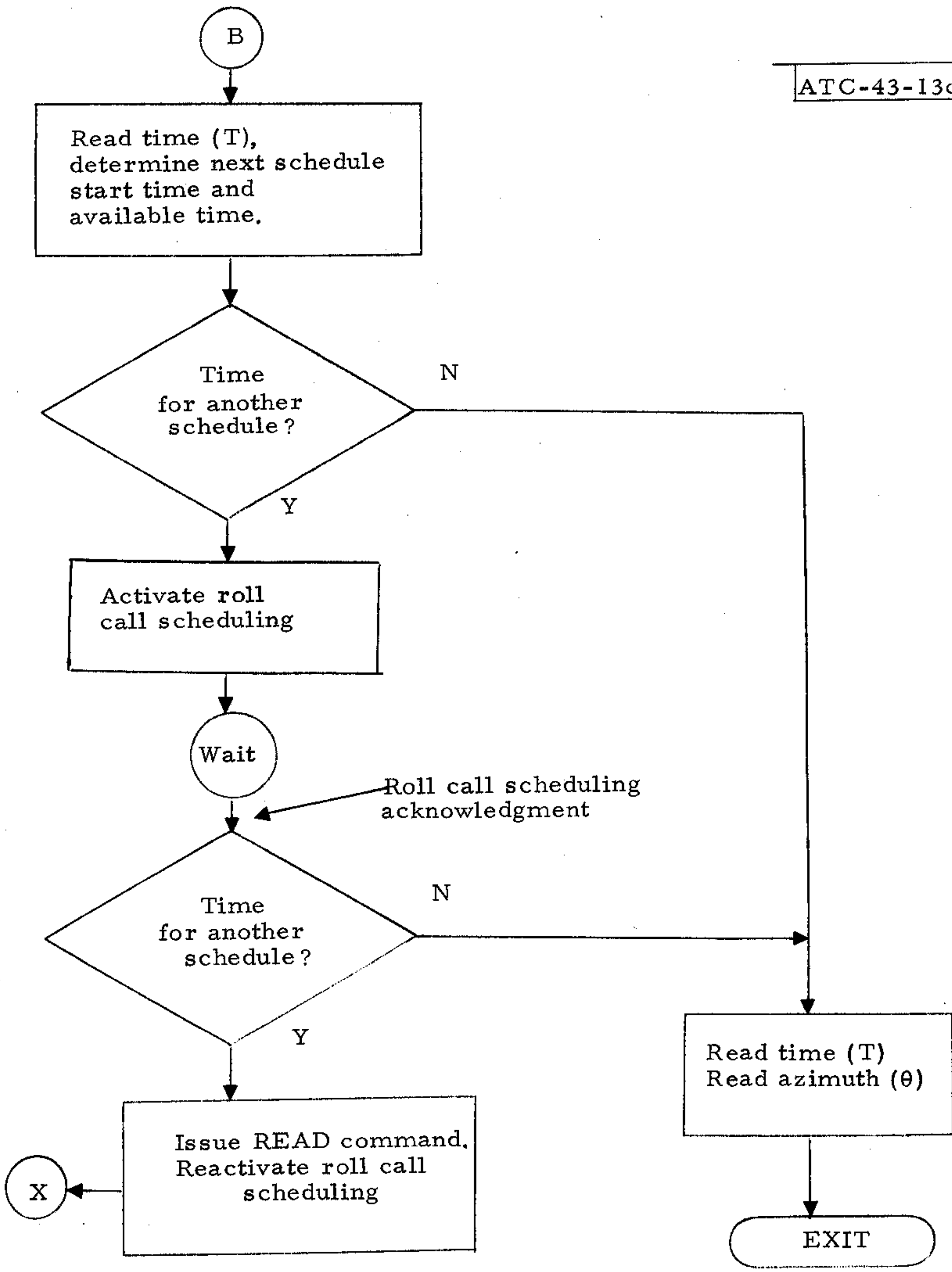


Figure 13c. Box C Expansion (Continued)

PART II

ROLL-CALL SCHEDULING

II-1. Overview

This Part of the report is a continuation of the exposition of DABS channel management begun in Part I, which was devoted to the channel control subfunction. Part I provides the context for the present discussion of roll-call scheduling, hence it is not necessary to include here a description of the part scheduling plays in the overall channel management picture. Roll-call scheduling is viewed as a subroutine, which is called upon to perform a specific task, namely the scheduling of DABS roll-call interrogations and replies so as to make efficient use of a limited interval of RF channel time.

Roll-called scheduling is "called" by channel control, which issues to it an enabling command that includes an actual time at which the RF channel may be used for the start of a schedule, and the interval of time available for the schedule. Roll-call scheduling, in turn, makes two responses to channel control: an acknowledgement and a completion notice. The acknowledgement tells channel control that roll-call scheduling either will produce a schedule (which it then proceeds to do), or that roll-call scheduling has determined that the available channel time is insufficient to accommodate a schedule. In the latter case, roll-call scheduling "returns" to channel control and the subroutine is finished. Before enabling roll-call scheduling, channel control has made its own check on the sufficiency of available time, but the test made is simple and it takes no account of the target ranges that will enter into scheduling and affect the schedule duration. Roll-call scheduling makes a more careful test, including target range information, and hence it can decide that no schedule should be produced. If a schedule is produced, the last act of roll-call scheduling is the generation of a completion notice, which returns control to channel control and informs the latter of the actual duration of the schedule.

The input data upon which roll-call scheduling operates is contained in the active target list, discussed in Section II-2. From this list roll-call scheduling determines which targets are candidates for the schedule and what kind of transaction (i.e., interrogation/reply pair) is pending for each candidate target. The output of roll-call scheduling is a set of interrogation and reply control commands, which are made available for transfer to the hardware (transmitter/modulation control unit and DABS reply processor) by channel control. These commands include the interrogation times and reply listening period start times which together constitute the "roll-call schedule," along with control bits which regulate the action of the hardware. In addition, each interrogation control command includes a fully formatted interrogation message. The structure of these commands and the message formatting task are discussed in Section II-3.

It will occasionally happen that the pending transactions on the active target list would require a schedule whose duration would exceed the available channel time if all were scheduled. In this case priority rules are applied so that the channel time is used for the most important business first. The exercise of these rules is included in the allocation and target selection tasks of roll-call scheduling, discussed in Section II-4. In general, surveillance is given priority over communications, when they cannot be combined, and single-segment communications outrank extended length message (ELM) transactions. Among the latter, uplink ELM's outrank downlink ELM's so that four general "allocation classes" result.

Uplink ELM's present a special case, in that a set of interrogations is transmitted to an aircraft, only the last of which elicits a reply. It proves convenient to transmit these non-final interrogations in a batch, ahead of the main portion of the schedule. If several uplink ELM's are to be scheduled, the batches of non-final interrogations are combined into one string of interrogations, none of which elicits an aircraft reply. This string is called the schedule precursor, and it is discussed in more detail in Section II-5.

Once it has been determined which transactions are to be scheduled, it remains to choose the times for the transmission of interrogations and the intervals to be reserved for replies. This is the basic "interrrogation/reply scheduling" task, and it must be accomplished in a way that makes efficient use of channel time while preventing any overlap of periods devoted either to interrogations or replies. The so-called "modified full-ring algorithm," which is used for this purpose, is introduced in its simplest form in Section II-6. The case discussed involves a roll-call schedule free of synchronized and downlink ELM transactions. The modifications necessitated by these two kinds of transaction are discussed in the following two Sections. Section II-9 is devoted to a special case in which only interrogations must be prevented from overlapping one another. The application which leads to this special case, and the simple modification necessary to accommodate it, are treated in that Section. The final Section consists of a summary of the roll-call scheduling procedure and an illustrative flowchart of an algorithm for its execution.

II-2. The Active Target List

The active target list is managed and updated by the actions of transaction update and target list update, hence, a detailed discussion of the structure of this list will be deferred to Part III, which deals with these subfunctions. However, the list is used as input by roll-call scheduling, and, therefore, a brief description of its structure, as seen by roll-call scheduling, will be given here.

The active target list consists of a header and a list of entries, with each entry corresponding to a different target. These targets will all be within the antenna beam at the time roll-call scheduling is activated, and some of them will have transactions still pending. This list may be organized as a linked list, and it will in any case be ordered in approximately decreasing order of target range. The sorting parameter that is actually used is called the "range delay," which equals the two-way propagation time to the target, augmented by the fixed transponder reply delay. The approximate ordering is exactly defined below in Section II-6.

The active target list header contains a pointer to the first target on the ordered list, or else a suitable indicator if the list is empty. The remaining list header fields contain certain channel time estimates and the range delay of the first target, all of which are used by the allocation task (described in Section II-4), and several fields relating to uplink ELM transactions, used in connection with allocation and precursor preparation (described in Section II-5).

In the list itself, each entry consists of a "target record" and a "transaction record". The target record contains information typical of the target itself, such as its range delay, and various control bits that indicate which tasks have been completed for this target so far in the present scan. The transaction record contains a coded description of the transaction waiting for execution, including a partially formatted interrogation message.

Target record fields utilized by roll-call scheduling are the following:

DABS address	(24 bits)
Range delay	(16 bits)
Range guard	(16 bits)
Range correction	(16 bits)
Target completion	(1 bit)

Two additional fields, generated for certain transactions by transaction update and appended to the target record, are

High power flag	(1 bit)
ALEC	(12 bits)

The generation of the data which reside in these fields will be described in Part III of this report. The significance of the fields will become apparent as their use is described below. Range delay, range guard and range correction are quantities used in the interrogation/reply scheduling process (Sections II-6 and II-7), while DABS address and ALEC (altitude echo) are used in message formatting (Section II-3). The target completion bit is used to exclude certain targets from consideration by roll-call scheduling for one reason or another. When this bit is set to "1" for a given target, roll-call scheduling must skip that target as though it were not on the active target list. The bit will be reset to "0" later by target list update or else the target will be removed from the active target list by that subfunction. The high power flag, when set to "1", is a request for high transmitter power in the forthcoming interrogation to this target. Roll-call scheduling must check to see if sufficient time has elapsed since the last use of high power (Section II-6) before including this request in the corresponding interrogation control command.

The various DABS uplink and downlink message types, specified in Reference 2, are combined in many ways to accomplish the system's surveillance and communications objectives. A particular pairing of interrogation and reply message types is called a transaction, of which eight types are recognized by channel management (defined below).

These types belong to two major classes of four types each: standard transactions and ELM transactions. Standard transactions always involve a single interrogation, paired with a single reply, while ELM transactions usually involve either multiple interrogations (with one reply) or multiple replies (with one interrogation). Basic surveillance tasks and some data link tasks are accomplished with standard transactions. ELM transactions are used only for ELM data link purposes.

The specific matching of transaction type to the needs of a given target is the province of transaction preparation and transaction update. Roll-call scheduling is concerned with transaction type only as it affects the channel time required for the execution of the transaction. All of the transaction information and data required by roll-call scheduling is contained in the transaction record. Those fields of standard transaction records used by roll-call scheduling are the following:

Priority	(1 bit)
Channel time	(16 bits)
Transaction type	(3 bits)
Format type (F)	(1 bit)
Data-block length (L)	(1 bit)
Interrogation type (IT)	(1 bit)
DABS lockout (DL)	(2 bits)
ATCRBS lockout (AL)	(1 bit)
Synchronization indicator (S)	(1 bit)
Altitude/Identity designator (AI)	(1 bit)
Reply length (RL)	(1 bit)
Air-to-ground data link message source (MSRC)	(4 bits)
Synchronous reply time (EPOCH)	(6 bits)
Clear PBUT (CP)	(1 bit)
Clear Comm-B (CB)	(1 bit)
Ground-to-air data link message (MA)	(56 bits)

Priority and channel time are used in the allocation computation, and their significance will be explained in Section II-4. The transaction type identifies this transaction as one of the eight types mentioned above. All the remaining fields are parts of the intended interrogation message and their use is discussed in Section II-3 below.

The fields of an ELM transaction record used by roll-call scheduling are:

Priority	(1 bit)
Channel time	(16 bits)
Transaction type	(3 bits)
Length	(4 bits)
Format type (F)	(1 bit)
Data-block length (L)	(1 bit)
Reply type for Comm-C interrogations (RTC)	(2 bits)
Segment number of ground-to-air ELM segment (SNC)	(4 bits)
Ground-to-air ELM Segment (MC)	(80 bits)

The first three fields have the same significance as the corresponding fields of standard transaction records. The field "length" indicates the number of message segments (interrogations for an uplink ELM, replies for a downlink ELM) in an ELM transaction. The remaining fields are parts of the intended interrogation, and they are used during the message formatting task (Section II-3).

As mentioned in Section I-1, there are two interrogation signal durations and two reply signal durations. These lengths are given, in microseconds in the following table, followed by the corresponding lengths in Range units (1 Ru = 62.5 nsec).

	Short	Long
Interrogation	18.5 (296)	32.5 (520)
Reply	64 (1024)	120 (1920)

TABLE 5 - Signal Durations

The four standard transactions correspond to the four possible combinations of the signal durations shown above, given a pairing of a single interrogation with a single reply.

ELM transactions of any kind always involve long interrogations and long replies. Two of the ELM transaction types, the so-called "clear Comm-C" and "clear Comm-D" transactions, pair a single interrogation with a single reply. The other two ELM transaction types, uplink ELM and downlink ELM transactions, usually involve either multiple interrogations or multiple replies (with the number of segments given in the length field), although single-segment uplink or downlink ELM transactions are allowed. From the scheduling point of view, an ELM transaction pairing a single interrogation with a single reply is indistinguishable from a standard transaction with long signal durations, and these cases are all treated the same way by roll-call scheduling. Only uplink or downlink ELM transactions of two or more segments are given special treatment in the formation of a roll-call schedule.

II-3. Message Formatting and Control Command Formulation

In order to include a transaction for a particular target in a schedule, roll-call scheduling must determine an interrogation time and a reply listening interval, and it must formulate interrogation and reply control commands, which are sent to the hardware (transmitter/modulation control unit and DABS reply processor) for execution. These commands include the computed interrogation and reply times, hardware control information, and in the case of the interrogation control command, a complete interrogation message.

As specified in Reference 1, an interrogation control command consists of 10 bytes for a short interrogation, and 17 bytes for a long one. In either case, byte 0 contains control information, bytes 1 and 2 contain the 16-bit interrogation time, and the remaining bytes contain the interrogation message.

Byte 0 consists of a two-bit command type field (set to "01" for a DABS interrogation), a power bit (P), a length bit (L) and four spares. The P-bit regulates transmitter power ("1" signifies the high power mode) and the L-bit identifies the length ("0" signifies a 56-bit message) of the interrogation message which follows. Message length is set by roll-call scheduling in accordance with the transaction type.

The P-bit has a more complex behavior which reflects the power-programming algorithm employed by the sensor. According to this algorithm, low power is always used for the first interrogation attempted for a given target in a particular scan. If the target provides an acceptable reply, high power will never be used in that scan. If no acceptable reply is received, reinterrogations will be permitted at high power, provided they occur in the same DABS period. If success is once achieved at high power, then any subsequent transactions in the scan (except uplink ELM transactions) will use high power. If the first interrogation (at low power) fails, and high power re-interrogations within the same DABS period also fail, then the first attempt in the next DABS period will again be at low power, with subsequent action as if the target were being interrogated for the first time in that DABS period.

The power-programming logic described here is managed by transaction update, which appends the so-called "high power flag" to the target record of a target requiring high power under these rules. The entire process is inhibited for close-in targets, and this feature is managed by the power programming control bit (in the target record). Transaction preparation sets this bit to signify "inhibit" (no high power) if target range is less than the value of an appropriate system parameter.

The presence of a high-power flag for a certain target does not, however, guarantee that the P-bit will be set to "1" by roll-call scheduling, since the duty cycle of the high-power transmitter must not be exceeded. To prevent this, roll-call scheduling must keep track of the time at which high power was last used, and allow P to be set to "1" only if the elapsed time exceeds a system parameter called HYTIME. The nominal value of HYTIME is 24000 Range units, or 1.5 msec. The computation required to allow the use of high power is considered to be a part of interrogation/reply scheduling, since it can be carried out only after the interrogation time for a transaction with high power requested has been determined.

Interrogation time refers to the time of transmission of the leading edge of the P_1 preamble pulse of a DABS interrogation (see Reference 2). This time is in range units, and the value contained in the control command is compared to the value of the 16 lowest-order bits of the real-time clock (see Reference 3). When coincidence occurs, the transmitter is fired.

The DABS interrogation message formats are specified in Reference 2. The contents of the various fields must be entered in the interrogation control command by roll-call scheduling. Most of these fields are obtained directly from the transaction record, whose fields in these cases carry the same name and abbreviation as those of Reference 2. For standard transactions, the fields F, L, IT, DL, AL, S, CP and CB will always be present in the transaction record. If S = "0" (unsynchronized interrogation), the fields AI, RL and MSRC (a total of six bits) will be found in the transaction record. If S = "1" (synchronized transaction), these six bits will be replaced by a dummy field, EPOCH, in the transaction record. The value of EPOCH is computed by roll-call scheduling itself in the course of scheduling the synchronized interrogation (see Section II-7). The SD (special data) field is also generated by roll-call scheduling, with the help of transaction update. If S = "0", the SD field is set equal to "1111" followed by twelve zeroes. If S = "1", the first four bits are set to "0000" and the last twelve bits are equated to ALEC (altitude echo). In the latter case, ALEC will have been generated by transaction update and appended to the appropriate target record. If the interrogation is long, the 56-bit MA field will be found in the transaction record, otherwise this field is absent. The final field of any interrogation, the DABS address, is obtained from the target record. The generation and overlay of parity bits is carried out in hardware by the transmitter/modulation control unit.

For an ELM interrogation, which is always long, all the fields except the address are obtained from the transaction record.

One function of the transaction record is to facilitate interrogation message formatting. For the most part, the various fields are merely assembled by roll-call scheduling in order to prepare an interrogation control command. The only computation required is the scheduling task itself, which produces the interrogation time and (when needed) the EPOCH field, and the check described above whenever high transmitter power is requested by transaction update.

Given an interrogation time, the time of arrival of the elicited reply is approximately known, since channel management is provided with a predicted range. Since predicted range is only an estimate, based upon track history, a listening period must be provided which is somewhat longer than the duration of the expected reply. The required information is provided by surveillance processing during its track update operation.

Surveillance processing, after smoothing a track with a new measurement, predicts range one scan ahead. Let the prediction be called R_p and let ΔR be a measure of uncertainty in R_p, which is also computed by surveillance processing. The computation of ΔR takes account of track firmness and the possibility of aircraft maneuver. A system parameter is included in the computation of ΔR, and it is intended that this parameter be adjusted so that predicted range will lie in the interval $(R_p - \Delta R, R_p + \Delta R)$ with high probability. This adjustment will, of course, depend upon experience with the tracking and scheduling algorithms.

Surveillance processing then computes the quantities

$$D = (2/c)\,(R_p - \Delta R) + \Delta_T \quad , \text{ and}$$

$$G = (4/c)\,\Delta R,$$

and places them in the surveillance file. The term Δ_T is the transponder nominal reply delay, measured from the time of reception of the beginning of the interrogation to the time of transmission of the beginning of the reply. This delay is specified to be 132 μsec (2112 Range units). The quantity D is called "earliest likely range" (by surveillance processing), or "range delay" (by channel management), and it represents the expected time interval between transmission of the start of the interrogation by the sensor and the reception of the start of the reply, given that the aircraft range is the minimum expected, $R_p - \Delta R$. Roll-call scheduling always schedules a reply listening interval to begin a time D after the interrogation time. The quantity G, called "range guard", represents the amount by which the reply listening interval must exceed the expected reply duration in order to accommodate actual ranges from the minimum expected to maximum expected range, $R_p + \Delta R$.

If an interrogation is transmitted at time T , then the reply listening interval will start at time

$$L = T + D$$

and last until time $L + G + R$, where R is the expected reply duration. The reply beginning will be expected to arrive between L and $L + G$, and this information must be given to the DABS reply processor so that preamble detection can be enabled during this interval. The reply processor also requires knowledge of the reply length (short or long) and the expected DABS address for reply verification and error detection and correction.

II-4. Allocation and Target Selection

Whenever channel control decides that a roll-call schedule should be produced, the channel time available for that schedule is known and provided to roll-call scheduling. The active target list, to which roll-call scheduling turns in order to generate the schedule, will have been updated or processed since the last schedule, so that it now looks like a new list of transactions (one per target) awaiting execution. There is no assurance that a schedule including all these transactions would fit in the available time on channel. Before it can embark on interrogations/reply scheduling, roll-call scheduling must assess the demands on channel time presented by the active target list, in view of the time actually available, and then decide how that time should be used. If there is time to execute a schedule which includes all the transactions on the list, then there is no problem. Otherwise, however, it must be decided which transactions will be included in the forthcoming schedule. This process is called allocation, and it amounts to the application of a set of simple priority rules.

All transactions are classified into one of four "allocation classes", depending on the nature of the surveillance and communications tasks the transaction is intended to accomplish. Standard transactions are designated either "priority" or "normal" and allocation class one consists solely of priority standard transactions. Allocation class two includes all normal standard transactions and any ELM transactions which do not involve multiple segments, such as the ELM clear Comm-C and clear Comm-D commands.

The reply control command contains six bytes, beginning with a control byte. The control byte has a two-bit command type identifier (set to "11"), a reply length bit ("1" indicates a long reply), and a five-bit reply window field. The reply window is equal to the range guard, G, in a special format whose LSB is 4 μsec (G is rounded up, if necessary). This window is used to control the interval over which preamble detection is enabled. Bytes 1 and 2 represent the earliest reply arrival time, denoted by L above, and bytes 3, 4 and 5 contain the DABS address.

All the information required to formulate a reply control command is available in the target record, except reply arrival time, which is computed by roll-call scheduling itself. Range delay and range guard are obtained from the surveillance file (along with DABS address and other data) by transaction preparation. The remaining target record field, range correction, is generated by transaction update after the first successful reply. It equals the time interval between the scheduled earliest reply arrival time and the time of arrival of the actual reply. Thus, range correction provides a range measurement, since range can be found from range delay (D) and range correction (K) as follows:

$$\text{Range} = (c/2)(D + K - \Delta_T) \ .$$

Range correction is used in this way by surveillance processing. It is also used in a special way in the scheduling of synchronized interrogations (see Section II-7).

Multiple-segment uplink ELM transactions comprise allocation class three and multiple-segment downlink ELM transactions make up allocation class four.

Standard transactions can be designated priority for many different reasons. The first transaction of a scan (called the "surveillance transaction") is always a priority transaction, as is any synchronized transaction. Transactions in which the interrogation requests ATCRBS identity have the priority designation, since these will usually result from emergency situations (pilot request). If the sensor experiences difficulty obtaining a satisfactory azimuth measurement, additional transactions may be added, by transaction update, in an attempt to measure azimuth before the target leaves the beam, and these will be priority transactions. Finally, any standard transaction is designated priority if it carries a Comm-A message or a Comm-B message request which has been given "high-priority" status by data link processing.

The principle of allocation is simply to give each allocation class preference over all lower classes, so that a schedule will exhaust all class one transactions before including any class two transactions, and so on. In order to do this without iteratively calculating a whole series of tentative schedules, roll-call scheduling makes use of estimates of the channel time required for each transaction (found in the transaction records) and totals of the estimated channel times demanded by the various allocation classes (found in the active target list header).

The exact amount of channel time required by a given transaction is the difference in duration of a schedule which includes the transaction and of another schedule which includes all other transactions but the given one. This exact channel time depends upon the target ranges and transaction types of all the targets included on the schedule, and hence, it cannot be predicted on the basis of data on the given transaction alone. However, experience with the interrogation/reply scheduling algorithm used by roll-call scheduling provides a simple rule which permits a conservative estimate of channel time required for a given transaction in any schedule, and this rule is used to generate channel time estimates at the time each transaction record is prepared. The basis for these time estimates will be given here, and the discussion will then return to their use in the allocation task.

The duration of a schedule which contains N transactions (of any kind) will depend somewhat on the actual distribution of the ranges of the N targets. However, the duration never exceeds an expression composed of two terms, the "channel time sum" and the "schedule overhead". The former term is a sum of "channel time estimates", one for each transaction, while the overhead term depends only on the range distribution. The channel time estimate is a time interval which depends only on the transaction type, and it represents the unavoidable increase in schedule duration resulting from inclusion of the transaction, augmented slightly to represent normal scheduling efficiency. The schedule overhead time is associated with propagation delays not fully utilized by other transactions, and it is approximated by the sum of the range delay for the longest-range target on the list and a constant term. If a schedule contained only one (standard) transaction, then the channel time estimate would be approximately equal to the actual time used by the interrogation and the reply, while the overhead would be almost equal to the propagation delay, or range delay, for this target. To permit later "fine tuning" of the system, the channel time estimates used for each transaction type are system parameters, and the constant term used in the computation of overhead time is also an adjustable system parameter. The suitability of this method for schedule duration estimation will become more clear after the discussion of the scheduling algorithm in Section II-6.

We introduce the allocation algorithm by discussing the simple case in which only allocation classes one and two are represented among the transactions on the active target list. Each of the transaction records contains a channel time estimate, one value for transactions with short replies and a different value for those with long replies. It is characteristic of the scheduling algorithm that the channel time estimate is independent of the interrogation length. The sum of the channel time estimates of all the priority transactions (class one) will be present in the active target list header, along with the corresponding sum for the class two transactions. Let the values of these sums be S_1 and S_2, respectively.

Roll-call scheduling, when enabled, is given an available time, T_a, for the schedule. It's first allocation task is to compute a schedule overhead time estimate, T_o, by the formula

$$T_o = D_1 + C \qquad ,$$

where D_1 is the first-target range delay (found in the active target list header), and C is a system constant. Next, the "remaining time", T_r, is computed:

$$T_r = T_a - T_o \qquad .$$

According to our rule for maximum schedule duration, a schedule containing all targets on the list should be no longer than $S_1 + S_2 + T_o$. Thus, if T_r exceeds $S_1 + S_2$, all transactions may be scheduled. If

$$S_1 \leq T_r < S_1 + S_2 \quad ,$$

there will be time for all class one transactions and some class two transactions. If

$$T_r < S_1 \quad ,$$

some class one transactions can be executed, but none of class two.

The procedure for allocation consists of the following steps, in which an "allocation level" is fixed:

1) Compute T_r and test the value against zero.
2) If T_r is zero or negative, the allocation level is set to "zero", and no schedule is produced. In this case (or if the active target list is empty), roll-call scheduling reports to channel control (the "acknowledgment") that no schedule will be produced.
3) If T_r is positive, it is then compared to S_1, and roll-call scheduling notifies channel control that a schedule will be produced.
4) If T_r is less than or equal to S_1, the allocation level is set to "one", and only class one transactions will be scheduled. All class one transactions are said to be "qualified for scheduling", and all others will be "unqualified for scheduling".
5) If T_r exceeds S_1, it is compared to the sum $S_1 + S_2$. The method is to compute

$$T_r' = T_r - S_1$$

and compare T_r' to S_2.

6) If T_r' does not exceed S_2 , the allocation level is set to "two", in which case all class one transactions are qualified and will be scheduled. Class two transactions are selected during scheduling.

7) If T_r' exceeds S_2 , all transactions of classes one and two are scheduled. In the case under discussion, there are no other transactions, and the preliminary allocation computation is complete.

In allocation level one, all class one targets have been qualified (i.e., considered eligible) for scheduling, although it is known that insufficient time is available to schedule them all. The decision to terminate the schedule is made during the schedule computation process, by checking the actual duration of the schedule against available time, as the schedule is generated, target by target.

In level two the situation is more complex, since there is insufficient time for all class two transactions and the scheduling of these transactions must be stopped at some point to allow time for subsequent class one transactions on the active target list. Thus, class two targets are selected on the basis of channel time estimates, beginning with the quantity T_r' , which is an estimate of the total time available for them. As each class two target is encountered during the scheduling operation (which makes a single pass through the active target list), the channel time estimate of the corresponding transaction is subtracted from T_r' . Class two targets are selected so long as this subtraction results in a positive quantity. In addition, as a precaution, the actual schedule duration is checked against available time as each transaction is added to the schedule. Note that, whenever the targets of a given class cannot all be scheduled, the longer-range ones are given preference by this procedure, which is adopted for its simplicity. The actual selection of class two targets is deferred until the schedule is computed in order to save an extra pass through the active target list.

Before going on to allocation levels three and four, we discuss a small complication resulting from the fact that DABS periods are characterized by one of four "schedule modes" (see Section I-5). The schedule mode sets rules for the inclusion or exclusion of synchronized transactions in the schedules of the relevant DABS period, regardless of considerations of available time. There are three basic modes:

1) Mixed mode - both synchronized and unsynchronized transactions are permitted in any schedule generated during the DABS period,
2) Unsynchronized mode - synchronized transactions are excluded, and
3) Synchronized mode - only synchronized transactions are included.

A fourth mode, really a variation of the third mode, includes only synchronized transactions and uses a modified scheduling algorithm which is described in Section II-9.

Since synchronized transactions are always class one transactions, the chief effect of the restrictions presented by schedule modes 2) and 3) is to change the value of the priority channel time sum. In schedule mode 2), the sum of the channel time estimates of the synchronized transactions (present in the active target list header) is subtracted from the priority time sum and the result used for the allocation computation as above. In schedule

3), we simply declare allocation level one, since only synchronized transactions are permitted.

In the general case, when uplink and downlink ELM transactions are present, the allocation computation does not stop with Step 7), described above, but continues in an analogous fashion to determine which allocation class (if any) will be incompletely served. The active target list header will contain enough information regarding uplink ELM transactions so that an estimate, S_3, of the channel time required to service all pending class three transactions can be made. Details of the list header structure, relevant to uplink ELM transactions, are given in Section II-5. At the end of step 5), roll-call scheduling has calculated the time T_r', which is an estimate of the time available for transactions of classes two, three and four. This time is compared to S_2, and in general step 7) takes the following form:

7) If T_r' exceeds S_2, the quantity

$$T_r'' = T_r' - S_2$$

is computed and compared to S_3.

8) If T_r'' does not exceed S_3, the allocation level is set to three, in which case transactions of classes one and two will all be qualified, and those of class four unqualified for scheduling. Class three transactions (multiple-segment uplink ELM transactions) will be selected during precursor scheduling, as described in Section II-5.

9) If T_r'' exceeds S_3, the process stops with the declaration of allocation level four. Transactions of classes one, two and three will all be qualified and scheduled, while multiple-segment downlink ELM transactions (class four) will be individually selected during scheduling (see Section II-8).

The whole point of the allocation computation is to determine which allocation class will be incompletely served. Those of higher classes are automatically qualified, and those of lower classes are unqualified for scheduling. The selection of transactions from the class to be incompletely served is made concurrently with scheduling. In all cases except allocation level one, this selection is made on the basis of channel time estimates. In allocation level three, the selection occurs during precursor scheduling, in all other cases during the computation of the main schedule itself. Although no specific "qualification bit" has been specified, some such indicator must be used for the benefit of transaction update, which must be able to determine which of the pending transactions were actually scheduled. The target record would be the logical place for such an indicator.

II-5. The Schedule Precursor

The precursor is a portion of a DABS roll-call schedule which is used only to service uplink ELM transactions (UELM's). The message formats and waveform characteristics of UELM interrogations and replies may be found in Reference 2. From a scheduling point of view, we need only know the following characteristics.

1) UELM interrogations and replies are always long.
2) A UELM transaction will contain no more than sixteen segments, each segment corresponding to an interrogation.
3) Internal message coding in the segments instructs the transponder to reply or not to reply, and the reply contains information

which identifies the successfully received segments of the UELM transaction.

4) Interrogations may be transmitted with any spacing, provided no two are closer together than 50 μsec (800 Range units), leading edge to leading edge. This minimum spacing is imposed to assure resuppression of ATCRBS transponders with each DABS interrogation. The interrogations which represent the segments of a given UELM may be spread over several schedules or even several scans, and other interrogations (but not UELM's) addressed to the aircraft may be interspersed between them.

5) It is intended that the first attempt to deliver a UELM contain as many segments as time permits, and that the final interrogation of the group be of the type which elicits a reply. The sensor can then evaluate the degree of success attained, by interpreting the information coded in the reply, and schedule another UELM transaction to send untransmitted or unreceived message segments. The re-interrogation may occur in the same or a subsequent scan.

Most of the transactions in a schedule will be standard transactions, pairing a single interrogation with a single reply. The basic scheduling algorithm (Section II-6) is designed to pack interrogation/reply pairs of this kind efficiently into the available time. It would be awkward and inefficient to include a complete UELM transaction in such a schedule, even if the uplink segments are transmitted as a burst, one interrogation every 50 μsec. The reason for this will be given in Section II-8, in connection with a similar problem in the case of downlink ELM transactions.

Inefficiency of channel time is avoided by the use of a precursor, taking advantage of the flexibility of interrogation spacing discussed above. Instead of including the non-reply-eliciting interrogations of a UELM in the main schedule, these are transmitted as a burst before the main schedule begins. If the total schedule is to contain several UELM transactions, the bursts just described are combined to form the precursor, which is one long burst of non-reply-eliciting interrogations, spaced every 50 μsec. These interrogations are individually 32.5 μsec in duration, hence each one, including the last one, is followed by a 17.5 μsec gap.

Any UELM interrogation which elicits a reply may be called a "collector", since its function (besides delivering a message segment) is to collect the technical acknowledgment data the reply will contain. The collector and its reply are then included in the main schedule. where they are indistinguishable from a standard transaction with a long interrogation and reply.

Once it is decided which interrogations will be included, precursor scheduling becomes a trivial task. The first interrogation is scheduled to begin at the schedule start time received by roll-call scheduling from channel control. Each successive interrogation time is obtained by adding 800 Range units to that of its predecessor. The main schedule start time is equated a time 800 Range units following the last precursor interrogation. All precursor interrogations and collectors are transmitted at low transmitter power, and the length bit will indicate "long". The fact that this computation is simple provides a secondary advantage to the precursor concept. The scheduling operation can quickly provide a set of interrogation times, when called upon to produce a schedule containing UELM's, the transmission of which will occupy the RF channel for a while, giving the scheduler time to develop the early part of the main schedule to follow.

It remains to discuss the allocation of channel time to UELM transactions. The basic approach is simply stated:

1) A collector is included in each UELM transaction in any schedule, and

2) As long as channel time is available, each pending UELM will be given time to transmit all message segments still undelivered.

The intent of this approach is to dispose of each UELM completely as soon as possible, rather than to use a limited time to service a number of UELM's partially.

The information necessary to carry out target selection in allocation level three is contained in the active target list header. The location of each UELM transaction record which represents the "current transaction" of a target on the active target list is given by a pointer in the list header. The current transaction for any target is the one which has been designated for execution in the next available schedule. This pointer allows the data in this transaction record to be used in formulation of hardware control commands during precursor scheduling, without the need for a pass through the entire target list. Also, for each UELM transaction, the active target list header indicates the number of message segments waiting to be delivered (this is called the "length" of the UELM) and a channel time estimate for delivery of the entire UELM.

Precursor channel time can be predicted exactly; it is simply 800 (N-1) Range units, for a message length N. The collector, embedded in the main schedule, will require a time equal to that of a standard transaction with a long reply, hence the usual estimate for this case is added to the precursor time to provide the desired estimate of total time for the UELM transaction.

Suppose there are M UELM transactions among the current transactions on the active target list, and let the channel time estimates for these transactions be $T_1, T_2, \ldots, T_M$. Then the total time estimated to be needed to accommodate all the pending class three transactions is

$$S_3 = T_1 + T_2 + \ldots + T_M \quad .$$

In allocation level three, the remaining available time, T_r'', does not exceed S_3. To find how many of the UELM's can be fully serviced, we need to find the largest integer, m, such that

$$T_1 + T_2 + \ldots + T_m < T_r''$$

This can be accomplished by subtracting T_1 from T_r'', T_2 from the result, and so on, until a positive result no longer occurs. Suppose we find, in this way, that

$$T_1 + T_2 + \ldots + T_m < T_r'' \leq T_1 + T_2 + \ldots + T_m + T_{m+1} \quad .$$

Then the first m UELM's will be "qualified" for scheduling in their entirety. The remainder,

$$T_r''' = T_r' - (T_1 + T_2 + \ldots + T_m)$$

is computed to find out how much time (if any) will remain to give partial service to the next UELM (if any). The standard estimate for the collector and its reply is subtracted from T_r''', and the (M + 1) - st UELM will be attempted only if there is time for the collector. After the collector time is removed from T_r''', the result is divided by 800 to determine how many interrogations for this UELM can be included in the precursor.

After this lengthy but straightforward process is complete, the precursor can be scheduled, since the qualified interrogations will have been so designated. Also, the collector interrogations qualified for inclusion in the main schedule will have been designated. Of course, in allocation levels one and two, no precursor will be formed (and no collectors scheduled), while in level four, all UELM transactions will be fully qualified.

II-6. Interrogation/Reply Scheduling

Interrogation/reply scheduling is the central activity of roll-call scheduling. Allocation is a necessary preliminary step and message formatting is part of the formulation of interrogation/reply (I/R) scheduling output, namely interrogation and reply control commands. An effective I/R scheduling algorithm is essential to efficient use of RF channel time. I /R scheduling is also a highly time-critical activity, since the scheduler must begin to produce useful output very soon after its activation, and the computation of the schedule must require less time than its execution, in order to allow time for the operation of transaction update. Of the many algorithms which have been considered for I/R scheduling, the one described here has been selected because it has the following desirable features:

1) The algorithm produces efficient schedules, regardless of the range distribution of the targets,

2) The computation is very simple and is accomplished in a single pass through the active target list,

3) The basic algorithm has the flexibility to accommodate the several distinct transaction types used in DABS, as well as to meet the special requirements of synchronized transactions, and

4) The targets are added to the schedule in the same order in which they are to be interrogated, so that schedule execution can begin before schedule computation is complete.

The I/R scheduling algorithm will be introduced in this Section in the relatively simple case of a scheduler which contains no synchronized or downlink ELM transactions. Uplink ELM transactions are included, since only the collector interrogation and its reply enter into the main schedule. The input, of course, is the active target list, modified by the action of allocation processing, which has qualified some targets and designated others as unqualified. Unqualified targets and targets marked "complete" by transaction update are ignored by the scheduler. The allocation level is also an input, together with the residual time allotted to that allocation class which is to be incompletely served. Although carried out concurrently with I/R scheduling, the final qualification of individual targets (e.g., class two targets in allocation level two) will not be discussed further here (see Section II-4). Finally, the main schedule start time is input, along with the remaining time available for this schedule, after the preparation of the precursor. These data are used to determine the latest permitted schedule end time.

The output of the I/R scheduling algorithm is a set of interrogation and reply times which constitute the actual schedule. The inclusion of these times in interrogation and reply control commands was discussed in Section II-3, along with uplink message formatting. The final step in power programming is also carried out concurrently with I/R scheduling, as previously mentioned. Certain targets on the active target list will have high power flags set by transaction update. These flags are treated as requests for high transmitter power, which will be granted only if sufficient time has elapsed since its last use. Roll-call scheduling will keep track of the time of last use of high power in the current DABS period, and after an interrogation flagged for high power is assigned an interrogation time, a check can be made on elapsed time. The first request for high power in any DABS period is automatically granted; after that the elapsed time must exceed the value of an appropriate internal system parameter. This activity is carried out along with I/R scheduling, and needs no further discussion.

To understand the scheduling algorithm, consider first a single transaction, characterized by a range delay, D, range guard, G, and a given transaction type. Suppose that the reply duration (short or long), as determined by the transaction type, is R. If the interrogation is assigned a start time, T, then the reply listening period must begin at time

$$L = T + D \quad .$$

It will be recalled that the range delay represents the out-and-back propagation time corresponding to the minimum estimate of target range, augmented by the fixed nominal transponder reply delay. The reply control command will specify that preamble detection for this target will be enabled between time L and time L + G. The listening period will last until L + G + R, to allow for the case of maximum estimated target range. If the interrogation duration for this transaction is I, then the interval from T to T + I will be reserved for the interrogation. These quantities are illustrated in Figure 14 where the interrogation interval is shown above the time line, and the reply listening interval is pictured below.

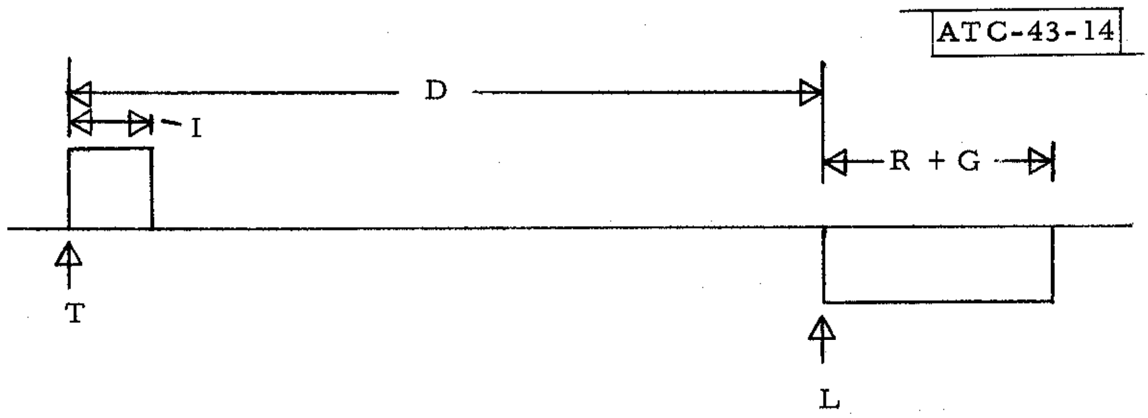


Figure 14 - Scheduling One Transaction

Suppose that the transaction illustrated above corresponds to the first target on the active target list, and that the second target to be scheduled has a range delay, D , which does not exceed D. The interrogation time, T, of the first transaction is set to the desired schedule start time, which then fixes the absolute value of L. The second target is scheduled by assigning the start time, L', for its reply listening period to follow immediately after its predecessor:

$$L' = L + G + R \quad .$$

Working backwards, the second interrogation time is computed to be

$$T' = L' - D' \quad .$$

Even the short reply is longer than the long interrogation, and the first reply listening interval is prolonged by the range guard, hence the range ordering ($D \geq D'$) guarantees that T' falls after the end of the first interrogation interval, as shown in Figure 15:

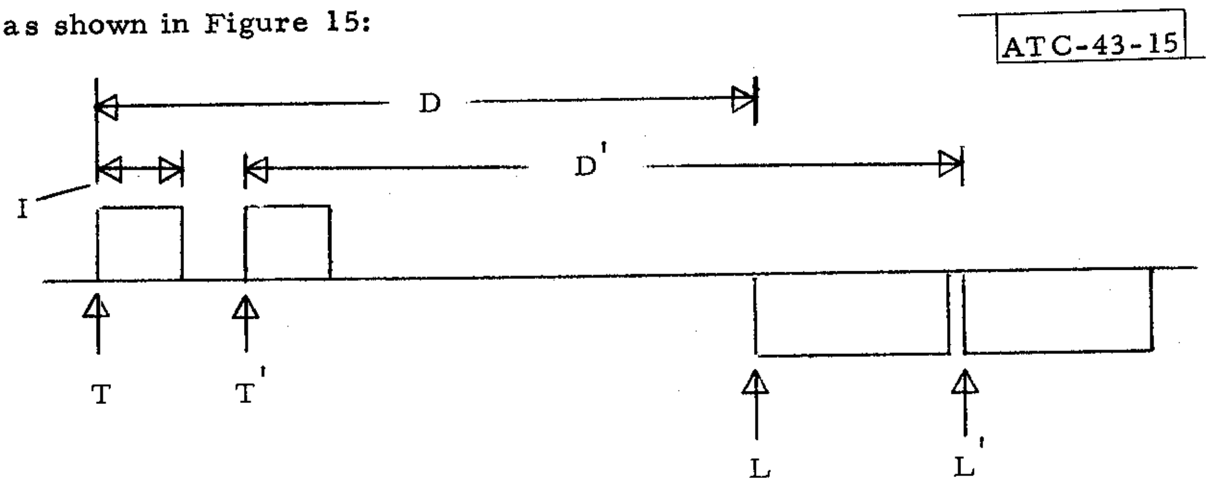


Figure 15 - Scheduling two transactions

Analytically,

$$\begin{aligned} T' - T &= (L' - D') - (L - D) \\ &= R + G + (D - D') \\ &\geq R \quad , \end{aligned}$$

and R always exceeds I. The durations of the second interrogation and reply are immaterial to this argument. Moreover, the interrogation spacing, $T' - T$, always exceeds 64 μsec (the short reply duration), which leaves ample time for resuppression of ATCRBS transponders. Because of the requirement to resuppress these transponders, no scheduling algorithm can be allowed to schedule interrogations with a spacing ($T' - T$) less than 50 μsec.

The process described above can evidently be continued, scheduling reply listening intervals back-to-back and computing the corresponding interrogation times, provided the active target list is ordered in decreasing order of target range delay. The gaps between interrogations will vary, being large when the two targets in question differ greatly in range delay, and also whenever the earlier transaction contains a long reply. As each reply listening interval is computed, its ending is checked against the latest permitted schedule end time to prevent overrun of the available time by the schedule. Overrun is possible, since the allocation calculations were based on estimates of required channel time.

Eventually, of course, the interrogations will encroach on the time already committed to listening for replies, and the portion of schedule produced, called a "cycle", is terminated. The I/R scheduling algorithm will assign each new listening period start time tentatively, and then compute both interrogation start and interrogation end times for the transaction at hand. The interrogation end time is tested against the stored value of the beginning of the first reply listening period start time of the cycle. If the interrogation falls in the clear, the transaction is included in the cycle, otherwise a new cycle is begun, starting with the transaction which did not fit. A new cycle begins at the end of the listening period for the last reply of the preceding cycle, and scheduling begins with the assignment of an interrogation time, just as was done for the first target on the active target list. A typical cycle is illustrated in Figure 16. Note the occurrence of a variety of transaction types.

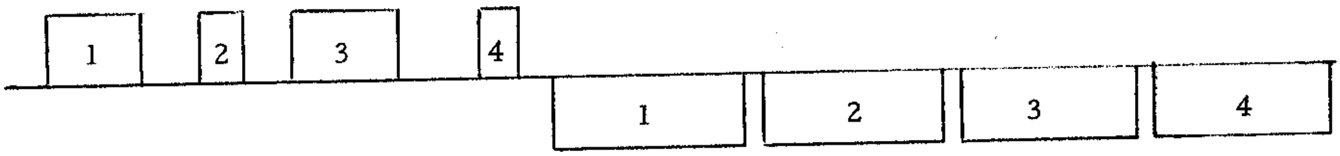


Figure 16 - A schedule cycle

As the scheduling process continues, the cycles become shorter, including fewer targets since target range delay decreases as the scheduler progresses down the list. The schedule ends when the list of qualified targets is exhausted. Often, the last cycle will contain only one or two targets. The transponder reply delay is specified to have a nominal value of 132 μsec (leading edge of interrogation to leading edge of reply), with a departure from nominal not exceeding ± 250 nsec (Reference 2). Since the long interrogation is only 32.5 μsec in duration, there will be a gap of 100 μsec or more between any interrogation and its reply, even for a target at minimum range.

It has been assumed, in the discussion so far, that the actual reply delay of a target will never fall outside the range (D, D+G), and that the active target list is rigorously ordered on decreasing range delay. The predicted (minimum) range delay and range guard are tracker outputs, and they will occasionally be sufficiently in error to cause a reply listening period to be scheduled too early or too late. Since preamble detection in the DABS reply processor is only enabled for an interval as long as the range guard, an early or late reply will be lost. If all the replies of a scan are lost in this way, the track will be coasted and the range guard value for the next scan will be increased. The algorithms for computing range delay and range guard contain internal system parameters, which will eventually be set to values which will effect a balance between wasting channel time with excessively large listening windows and missing replies for the opposite reason.

A certain departure from perfect range ordering can be tolerated if suitable account of the degree of departure from perfect order is taken in establishing the listening window. It was shown previously that the interval between one interrogation and the next ($T'-T$) exceeds the first reply duration (R) by the amount

$$G + D - D' .$$

The range guard is never negative and, in the example discussed above, $D-D'$ was positive (or zero) because of the range ordering. But if the range guard is increased by a constant amount, d, from the value required to accommodate tracking errors, then decreasing range ordering can be violated, so long as

$$D' \leq D + d \quad ,$$

and the interrogation spacing will still be no smaller than R. It is desirable to allow a small departure from perfect range ordering in order to speed up the computational task of ordering the active target list (which occurs frequently), even though a small amount of channel time is wasted in the prolongation of listening periods.

In the DABS specification, the "ordering discrepancy", d, is an adjustable system parameter, nominally equal to a few microseconds. The ordering required of the active target list is stated (in Reference 1) as follows: Let D_1 and D_2 be the range delays of two targets, T_1 and T_2, on the active target list. If

$$D_1 < D_2 - d \quad ,$$

then T_1 must follow T_2 on the active target list. If

$$D_2 < D_1 - d \quad ,$$

then T_2 must follow T_1. If

$$|D_1 - D_2| \leq d,$$

then the order is immaterial. The largest mismatch this specification permits is d, as in the example above. The extra term, d, must be included in the computation of every range guard. One way to achieve an approximate ordering of this kind is to mask some low order bits of the range delay word, and order strictly on the high order bits, using a fast algorithm such as the bubble sort. The effect of the masking is that the bubble-sort algorithm will find the list in satisfactory order after fewer passes than would otherwise be required.

It should be noted that a considerable departure from perfect ordering could be tolerated without any compensation in the form of an increased range guard because the short reply (64 μsec) exceeds the minimum interrogation spacing (50 μsec) required for resuppression. However, in schedules which include synchronized transactions this freedom cannot be utilized, for a reason discussed in Section II-7. In general, schedules are mixed in character, including both synchronized and unsynchronized transactions, and it seems simplest to use a single rule in all cases which guarantees the extra interrogation spacing when it is needed.

II-7. Scheduling Synchronized Transactions

The capability of executing synchronized transactions is built into the DABS sensor in order to support the possible implementation of an airborne CAS (collision avoidance system) which makes use of DABS replies. The service provided by DABS is called "synchronous service," and the relevant features which affect the DABS design are fully specified (in Reference 1). The overall CAS scheme which uses DABS synchro service is a provisional system concept at this time, and only its basic features will be described here, in order to provide the context and motivation for the special properties of synchronized transactions and their scheduling requirements.

The basic idea is for aircraft to perform one-way ranging on other aircraft in their vicinity by timing the arrival of synchronized DABS replies from these neighbors. The listening aircraft, which receives the benefit of CAS service, requires special equipment, including a clock, a receiver for the 1090 MHz DABS replies, and a CAS processor. The DABS interrogator cooperates by timing its synchronized interrogations in a way which permits the one-way ranging, but the aircraft whose presence is detected (and range measured) by the listener need have no special equipment other than a DABS transponder. Thus, all DABS-equipped aircraft would announce their presence by means of DABS replies, but only CAS-equipped aircraft could detect these aircraft and range on them. The synchronized DABS replies will always contain DABS ID and altitude, hence the listener can track each proximate aircraft in range and altitude, thereby deriving the information necessary to assess collision threat and determine an evasion maneuver.

Each synchronized interrogation is timed (by the DABS interrogator) so that the reply it elicits will be transmitted by the addressed aircraft at a predetermined time. A coded value of this emission time is included in the synchronized reply, and by comparing this time code with the reading of its on-board clock, a listening aircraft determines range from the one-way transmission. The clocks of CAS-equipped aircraft are kept in synchronism by the ground network of DABS sensors, using synchronized interrogations in the following way.

Each aircraft receives a synchronized interrogation from a DABS sensor once each scan, and it responds with a synchronized reply. Each synchronized interrogation contains a six-bit field, called EPOCH, which represents "system time" at the predicted moment when the aircraft will respond to this interrogation. The aircraft copies the EPOCH field into its synchronized reply to identify the time of its transmission. If the aircraft is CAS-equipped, the value of EPOCH is used, at the moment of reply transmission, to synchronize the on-board clock so that all clocks can maintain system time. The airborne clock is synchronized by forcing it to display the value of EPOCH at the reply time. The so-called system time is derived from the DABS clock in the sensor, and the clocks of all sensors are synchronized on the ground. The least significant bit of the EPOCH field, called the "synchro subepoch", has a nominal value of 16 μsec, hence, the EPOCH field corresponds to a six-bit portion of the DABS sensor clock. The subepoch has been made a system parameter, capable of assuming the additional values 8 μsec and 4 μsec. Use of these values would shift the position of the EPOCH field, relative to the sensor clock, to the right by one or two bits. Reference 3 contains a description of the synchronizing procedures for the sensor clocks. The EPOCH value and the airborne clock both represent time in units of the synchro subepoch with a dynamic range of six bits. The resultant range ambiguity is too large to be of consequence in the CAS application.

The interrogations must be accurately timed to assure that the synchronized reply is transmitted (leading edge) at the time coded in the EPOCH field, with a precision much higher than the subepoch quantization. To accomplisn this the sensor must know target range very accurately, and hence, a synchronized interrogation will only be made after an unsynchronized transaction which yields a range measurement, both transactions occurring in the same passage of the antenna beam past the target. The new range measurement is used in timing the synchronized interrogation, according to the following procedure.

The scheduling of a synchronized transaction begins by first establishing a reply listening interval and interrogation time, just as if the transaction were unsynchronized, and then modifying these times in a special way. The range delay and range guard used in the first step are unchanged from the values used in the most recent unsynchronized transaction. This latter transaction will have produced a measured "range correction", the interval from the start of the listening period to the actual arrival of the leading edge of the reply. Since only a few milliseconds, at most, will have elapsed since this range measurement, it is assumed that the synchronized reply would arrive after the start of the synchronized reply listening interval by the same "range correction" interval. Knowing this, and the actual range of the target (inferred from the range correction), the scheduler can determine the time of emission of the synchronized reply by the aircraft, given the interrogation time established in the first step. In general, this emission time will not be "on a subepoch", and to make the emission time an integral multiple of the subepoch value, Σ, a delay must be introduced in the interrogation time and reply listening interval. This is called the "subepoch delay", and after its computation the scheduler must test to be sure that the delayed interrogation interval will still fit within the current cycle (unless the synchronized transaction, as tentatively scheduled, happens to begin a cycle). If the subepoch delay would prevent the transaction from fitting in the cycle, then a new tentative interrogation time is established, beginning a new cycle, and the subepoch delay computation is repeated.

The subepoch delay calculation, briefly described above, is carried out as follows. Let L represent the listening interval start time tentatively established for the synchronized reply. If the range correction is K, and the original range delay is D, then the reply will actually arrive at L + K, and the interval from interrogation start to reply arrival will be D + K. Thus, the one-way propagation time is

$$\tfrac{1}{2}\ (D + K - \Delta_T)\ .$$

The transponder reply delay, Δ_T, was introduced in Section II-3, along with a similar analysis of the measurement of target range. Finally, we compute the reply emission time, E, which results if the transaction is scheduled with a listening interval starting at L:

$$E = L + K - \tfrac{1}{2}(D + K - \Delta_T)$$

$$= L - \tfrac{1}{2}(D - K - \Delta_T) \quad .$$

In order that the synchronized reply be emitted at an identifiable time, the interrogation must, in general, be delayed to cause the emission time to be an integral multiple of Σ. When that is the case, the emission time will be represented by a sensor clock value which has all zeroes to the right of the EPOCH field, and the value of the EPOCH field for this time is the desired identifier which is incorporated into the synchronized interrogation.

Suppose that the emission time, E, computed above, can be represented in the form

$$E = k\Sigma + X \quad ,$$

where k is an integer and X is less than Σ. Then a delay,

$$d = \Sigma - X \quad ,$$

added to the interrogation time, will change E to the value

$$(k + 1)\ \Sigma \quad .$$

This value, d, is the desired subepoch delay, and the EPOCH code will be the six lowest-order bits of the binary representation of (k + 1). The interrogation and reply listening interval start times, computed in the first step, are increased by d to satisfy the synchronizing requirement for this transaction.

The scheduling algorithm described in Section II-6 assures that no two replies will overlap, i.e., garble each other, as they are received at the sensor. It is necessary also to preclude the possibility of the garble of a synchronized reply as it is received in a listening CAS-equipped aircraft. If aircraft B follows aircraft A in the interrogation sequence of a given roll-call schedule, then there will be no garble of their replies, at any receiving point, if the reply generated by A has completely passed by aircraft B before B's reply begins. If A and B lie on a radial line through the sensor, then the desired effect will automatically be produced by the scheduling algorithm itself. In general, a vector drawn from A to B will have a component perpendicular to the direction from the sensor to either aircraft, and this requires an additional delay in the interrogation to B. A single adjustable system parameter is used for this delay to cover all cases allowed by the geometry and antenna pattern in use. An analytical discussion of the isolation problem is given in Appendix A.

Isolation delay is introduced in the scheduling of every synchronized transaction. Account is taken of the delay already introduced for reply emission timing, and additional delay is introduced in multiples of the subepoch value, Σ, to preserve the synchronous character of the transaction. The number of subepoch intervals added to the initial subepoch delay to achieve isolation is added to the EPOCH field already computed.

If a synchronized transaction is followed by another synchronized transaction, then the isolation delay of the second protects the two replies from garbling each other. However, when a synchronized transaction is followed by an unsynchronized one, then an isolation delay must be added to the unsynchronized interrogation to protect the synchronized reply. In this case, the isolation delay parameter is added directly to the interrogation and reply times computed for the unsynchronized transaction. Whenever isolation delay is added to any transaction which was tentatively scheduled in an existing cycle, another check must be made to be sure the transaction still fits with the delayed times.

II-8. Scheduling Downlink ELM Transactions

The procedure for scheduling a downlink ELM (DELM) transaction is only slightly different from that for a standard transaction. In allocation level four, all DELM transactions are qualified during the allocation calculation, and the time remaining for DELM transactions is used during the I/R scheduling step to qualify transactions individually. The situation here is analogous to the handling of class two transactions in allocation level two. First, the scheduling of a DELM is described, and then the qualification procedure can be explained.

Suppose that a schedule cycle is being developed and that the next transaction fetched from the active target list is a DELM. Suppose further that the DELM transaction will fit in the cycle, taking account of isolation delay, if required by the preceding transaction. The DELM interrogation will elicit a string of replies, one every 136 μsec. The number of replies is controlled by bits coded in the interrogation and set by roll-call scheduling itself. The reply string is treated as a single, extra long reply by the scheduler, and is scheduled in the cycle in the normal way. The interrogation time is found by subtracting the range delay from the start time of the DELM reply listening period, just as for a standard transaction. If a DELM begins a cycle, it is again scheduled like an ordinary transaction, taking account of the unusually long reply listening interval.

The only unique feature of DELM scheduling is that the current cycle is always terminated with the DELM transaction, even if the DELM begins the cycle. In most cases, a transaction following a DELM would be so delayed by the long DELM reply listening interval that it would not fit within the cycle. To save testing for a case which rarely occurs, the scheduler automatically goes on to start a new cycle. This causes the DELM to be responsible for a little wasted time, from the end of the DELM interrogation to the start of the first reply listening period of the cycle.

We can now make an accurate assessment of the channel time required by the inclusion of a DELM in a schedule. Let the DELM interrogation time be T and the start of the first reply listening period in the cycle (called the "cycle test time") be L_o. If the DELM begins the cycle, L_o will be the start time of the DELM reply listening period. The DELM will thus require a channel time equal to $(L_o - T)$ plus the time required for replies. For N replies, the reply time is

$$(136N - 16)\ \mu\text{sec},$$

since the replies are 120 μsec long with a 16-μsec gap between them.

Before qualifying a DELM which has been fetched for scheduling, the scheduler will test to see if the remaining (class four) channel time exceeds $(L_o - T)$, after establishing the tentative interrogation time, T. If the transaction fails this test, this and all subsequent DELM transactions are unqualified for scheduling. If the transaction passes the test, the remaining channel time is reduced by the value of (L_o-T) and the result used to determine the number of reply segments which can be scheduled. The formula for reply time, given above, is used, and if there is time for fewer segments than the number requested in the transaction record, then all subsequent DELM transactions are unqualified.

The scheduler will also modify the MC field of the interrogation to conform to the number of replies desired. The first sixteen bits of this field, in a DELM interrogation, correspond to reply segments. If an earlier reply informed the ground that an aircraft wished to transfer a DELM containing M ($M \leq 16$) segments, then the transaction record for this DELM will first be prepared with an MC field beginning with M "1"'s followed by (16-M) "0"'s. If there is time for all M replies, the interrogation will be transmitted with the MC field copied from the transaction record. If there is time for only $K < M$ replies, the transaction record is unchanged, but the transmitted MC field will have K "1"'s followed by (16-K) "0"'s.

After the replies are received and assessed by transaction update, the "1" corresponding to a successfully received segment will be changed (by transaction update) to a "0" in the transaction record. Thus, a DELM appearing for a second or third time on the active target list for scheduling, can have any pattern of "1"'s and "0"'s in the leading 16-bit portion of the MC field. The number of "1"'s equals the number of segments not yet transferred. If there is insufficient time for all of them in a particular schedule, some of the "1"'s will be changed to "0"'s in the actual interrogation message by roll-call scheduling. Again, if there is time for K segments, the first K "1"'s will be copied into the interrogation MC field, the remaining bits will be set to "0".

II-9. Interrogation Scheduling without Reply Processing

In Section I-5 the various DABS schedule modes were described, including a mode in which only synchronized transactions are to be scheduled. It is expected that this mode will only be used in a situation in which several sensors are providing surveillance of a metroplex area in a tightly coordinated way. One feature of this coordination will be a time-sharing, among the sensors, of the channel time devoted to synchronized transactions. In this scheme, ATCRBS periods and DABS unsynchronized periods occur at the same time at each sensor, but a time period devoted to synchronized transactions is split into subperiods, one for each sensor. The result of this subdivision and time sharing is a loss of channel time available to any one sensor. One way of recovering some channel time is to ignore the replies during the synchronized DABS period, scheduling interrogations on into the time when replies are due to arrive. The scheduler computes interrogation times by placing reply "listening periods" in sequence, including isolation delay, so that there will be no garble at an airborne receiver. In practice, the scheduler for this mode can be implemented from the normal scheduling algorithm simply by ignoring the test usually made to see if a transaction will fit within the cycle. Equivalently, a "cycle test time" can be established which is far enough into the future so that every transaction will be included.

It can be shown that none of the several sensors participating in the time sharing scheme will cause garbling of the replies elicited by any other sensor, provided the last reply scheduled by one sensor arrives at that sensor (even though the reply is to be ignored) before the next sensor on the channel transmits its first interrogation. This is guaranteed by retaining the standard test, made by the scheduler, to be certain that each reply will be received before the end of the DABS period.

II-10. A Roll-Call Scheduling Algorithm

The roll-call scheduling task breaks up naturally into allocation and I/R scheduling, and the latter decomposes into precursor scheduling and preparation of the main schedule. The relationship of these three subtasks to the roll-call scheduling input and outputs is shown in the overall flowchart, Figures 17a and 17b. The subtasks are expanded in the subsequent figures.

When activated, roll-call scheduling is provided with a schedule start time, SS, and an available channel time, AT. From these quantities the schedule limit, SL, is computed for later use in making sure that the schedule does not overrun the available time. The allocation calculation makes use of the available time (AT) and the active target list to determine the allocation level, AL. An additional output of the allocation task is the time remaining, here called RT, for the allocation class which may be incompletely served.

After performing the allocation calculation, an acknowledgment is made to channel control, indicating whether or not a schedule will be produced. No schedule is produced if the allocation level has been set to zero or if the active target list contains no pending transactions.

Next, a test is made to see if a precursor is required. By definition, a precursor will always be scheduled (or at least attempted) in allocation level three, but in level four it is necessary to check for the presence of class three (multiple-segment uplink ELM) transactions.

Through some error, it is possible that the schedule limit will be reached during precursor scheduling, hence, a test for this occurrence is made and a suitable exit is provided. If there is time for a main schedule, the schedule start time (SS) is changed to the value of the precursor end time, and the main schedule is prepared.

The roll-call scheduling exit consists of a completion notice to channel control, which includes the actual end time of the computed schedule. Interrogation and reply control commands are the major output of roll-call scheduling, and they are buffered for transfer by channel control, as explained in Section I-3.

The allocation task is expanded in the flowchart of Figures 18a and 18b. The available time (AT) is used, together with the channel time sums contained in the active target list header. These sums are the following:

S1	Priority channel time
S11	Synchro channel time
S2	Normal channel time
S3	Total uplink ELM channel time

The synchro channel time is used to correct the priority channel time when the schedule mode is "unsynchronized", as explained in Section II-4. The calculation of remaining time, RT, follows the equations given in that Section.

The output of the allocation computation is simply the allocation level and the modified remaining time, or else a flag which indicates that the active target list is empty (i.e., no pending transactions). The individual qualification of uplink ELM transactions is incorporated into the precursor scheduling, which represents a minor departure from the literal specification of Reference 1.

The allocation algorithm represents an iterative subtraction of channel time sums from the remaining time to find out which allocation class, if any, is to be incompletely served. If there are no pending transactions of a given class, the corresponding channel time sum will be zero, and the algorithm will, in effect, advance to the next test. In the test for an allocation level of three (or higher), RT is compared to S2 + CTSL + 800, rather than to S2, to prevent the declaration of level three when there is in fact insufficient time for one precursor interrogation and the collector transaction of the first UELM.

Precursor scheduling is expanded in Figures 19a and 19b. The task is begun with assigned values of SS and SL, the schedule start and schedule limit times, and a value of RT, the remaining time. If the allocation level has been set to four, then all class three transactions will be scheduled and RT will be used in the preparation of the main schedule, in connection with the qualification of DELM transactions. In this case, RT is unchanged during precursor preparation. In allocation level three, RT is used to qualify UELM transactions and individual UELM interrogations. We are assured that there will be time for at least one precursor interrogation and the corresponding collector transaction by the nature of the allocation algorithm.

Initialization consists of accessing the first class three transaction from the active target list. The "schedule end time" variable, SET, is initialized to the schedule start time, since each new interrogation time is equated to the current value of SET.

Testing for adequate channel time is bypassed in allocation level four since it has already been verified that there will be time for the complete service of all UELM transactions in that case. In level three, each transaction is checked for time availability, beginning with a test to see if the entire transaction can be accommodated (based on the channel time estimate, CTE, for the full transaction). If the full transaction will not fit, based on the estimated remaining time, a check is first made to see if the collector will fit. The collector will require a time CTSL, the system parameter representing the channel time estimate for a standard transaction with long interrogation and long reply.

If the collector will not fit, the precursor is already finished; otherwise the number of precursor interrogations which will qualify is computed. This number will never equal N-1, the number of precursor interrogations the full transaction would require, since there was insufficient time for a full transaction. Each precursor interrogation requires 800 Range units (50 μsec) of channel time. If only the collector will fit the precursor is already finished. If one or more interrogations will fit, their number is determined and a flag set to terminate precursor processing when these interrogations are scheduled.

If the transaction was fully qualified for scheduling, then the number of precursor interrogations is equated to N-1, i.e., one less than the number of message segments for the transaction. The last segment, of course, is included in the collector. Whenever a class-three transaction is included, fully or not, the collector transaction is separately "qualified," so that it will be included in the main schedule by I/R scheduling.

Once the qualification of an individual UELM is established, the interrogations are scheduled, one every 800 Range units, and the interrogation control commands are formulated. The variable SET is revised with each scheduled interrogation. The process repeats until all qualifying segments of each qualified UELM are scheduled.

It should never happen that a precursor will reach the schedule limit, since no precursor will be attempted unless it is estimated that there will be a following main schedule, containing at least one collector, along with any class one and class two transactions. Nevertheless, a test is made before scheduling each precursor interrogation to verify that actual channel time is available. (The test allows for the full 50 μsec, so that all suppressed transponders will be recovered before any other channel activity is attempted.)

Preparation of the main schedule is the major portion of the roll-call scheduling task. This task will be activated only if the active target list is not empty, and in passing through the list, only targets whose "target completion" bit is not set to "1" are considered for scheduling. The significance and regulation of the target completion bit will be explained in Part III, in connection with the target list update function.

For definiteness, it is assumed that a linked list structure is used for the active target list. Thus, threading the list consists of the steps: 1) advance a pointer, 2) test if list end has been reached (i.e., previous target was the last on the list), and 3) fetch the target record and current transaction record indicated by the pointer. The same procedure was used in Figures 19a and 19b to thread through the class three transactions.

Each target fetched is tested for qualification for scheduling. DELM transactions will all be qualified in allocation level four, but the I/R scheduling algorithm will determine which DELM transactions, if any, will actually be scheduled. Usually, the last DELM to qualify will do so with a reduced number of message segments. This process is called "partial qualification" in Reference 1.

The main scheduling task is entered with values of schedule start and limit times, and a remaining time, used in allocation level four for DELM qualification. Schedule end time, SET, is initialized to schedule start time, SS, since each target is scheduled at a time determined by the end of the schedule so far computed. The loop for fetching the first qualified target on the active target list (ATL) follows the logic stated above.

The next step, which follows the entry point "B", is the assignment of an interrogation time, T, and a reply listening period start time, L, for the first target of a cycle. Deferring discussion of the DELM case, a cycle test time, CTT, is established, which is used to check whether the interrogations for subsequent targets will fit within the cycle.

If the first target of a cycle requires a synchronized transaction, the subepoch delay and EPOCH value are computed according to the equations given in Section II-7. The equation in the flowchart gives the reply emission time, E, which would result if no delay were applied (TRD is the transponder reply delay of 132 μsec). In the usual case, in which replies are processed by the sensor, the cycle test time is modified (along with T and L, of course) to take account of the subepoch delay. If replies are not being processed, all transactions will be synchronized and the cycle test time is equated to a time which can never be reached by interrogations (SL is a convenient choice) in order to inhibit the ending of a cycle.

At this point (entry point "E"), we are ready to schedule the transaction. However, a test is first made to be sure that the schedule will not overrun the schedule limit, SL. If not, the schedule end time is updated and interrogation and reply control commands are prepared and "released" for execution. In Section I-3 an option was described in which roll-call scheduling collected these commands until a cycle was computed (or the precursor) and then released then as a group by means of a response to channel control. The exact mechanism is sensitive to the actual implementation of I/O between hardware and software in the sensor, hence, in these charts we do not elaborate further on the "release" of these commands, other than to indicate the point in the flowchart at which their release is first possible. If replies are not being processed, the reply command step is simply bypassed.

Having scheduled the first target of a cycle, a new qualified target is fetched and tentatively scheduled by placing its reply listening interval directly after that of its predecessor. The corresponding interrogation end time, S, is found and checked against the cycle test time. If the new transaction will not fit, a return to "B" is made, beginning a new cycle with this target. If the transaction is not a DELM (see below) then it is tested to see if it is a synchronized transaction (Synch?), or if it follows a synchronized transaction (Post-synch?). It is understood that a flag will be set, or an equivalent technique used, to record the fact that the previous transaction was synchronized.

Post-synchronized transactions require the standard isolation delay, ISO. Synchronized transactions require the subepoch delay, computed as before, together with a possible further delay for isolation, as described in Section II-7. The result of either calculation is a delay which must be added to T and L before scheduling. First, the delayed interrogation is checked again for fit in the cycle, and if the test succeeds the target is scheduled and the process is iterated. If the delay pushes the interrogation end time past the schedule limit, then the transaction will start a new cycle by way of entry point "B".

The special treatment of DELM transactions, which includes qualification as well as scheduling, is shown in Figures 20e and 20f. The flowchart is entered at "I" if the DELM appeared to fit in an existing cycle, and at "C" if the DELM is to begin a new cycle. The standard post-synch test is made after entry point "I", and if the resulting delay precludes the inclusion of the DELM in the current cycle, new times for interrogation and reply are computed, and the flow merged with that of entry point "C".

In either case, we then compute the amount of channel time, X, which the interrogation and a single reply will require, anticipating the termination of the cycle with the scheduling of the DELM. The remaining time is reduced by this amount and the result tested for sign, to see if there is time for this

DELM with at least one reply. The corresponding schedule end time test is also made. If either test fails, a suitable flag is set to inhibit further DELM scheduling, and returns are made to points where a new target is fetched, either to continue the cycle or start a new one.

Once it is established that the DELM can be scheduled with at least one reply, further replies are scheduled until either a) all segments are scheduled ($M = N$), b) the estimated remaining channel time runs out ($RT \leq 0$), or c) the schedule limit is reached. Reply control commands are prepared as each reply is scheduled, and the final step is the preparation of the interrogation control command. This latter command indicates which message segments are to be sent in reply, as described in Section II-8.

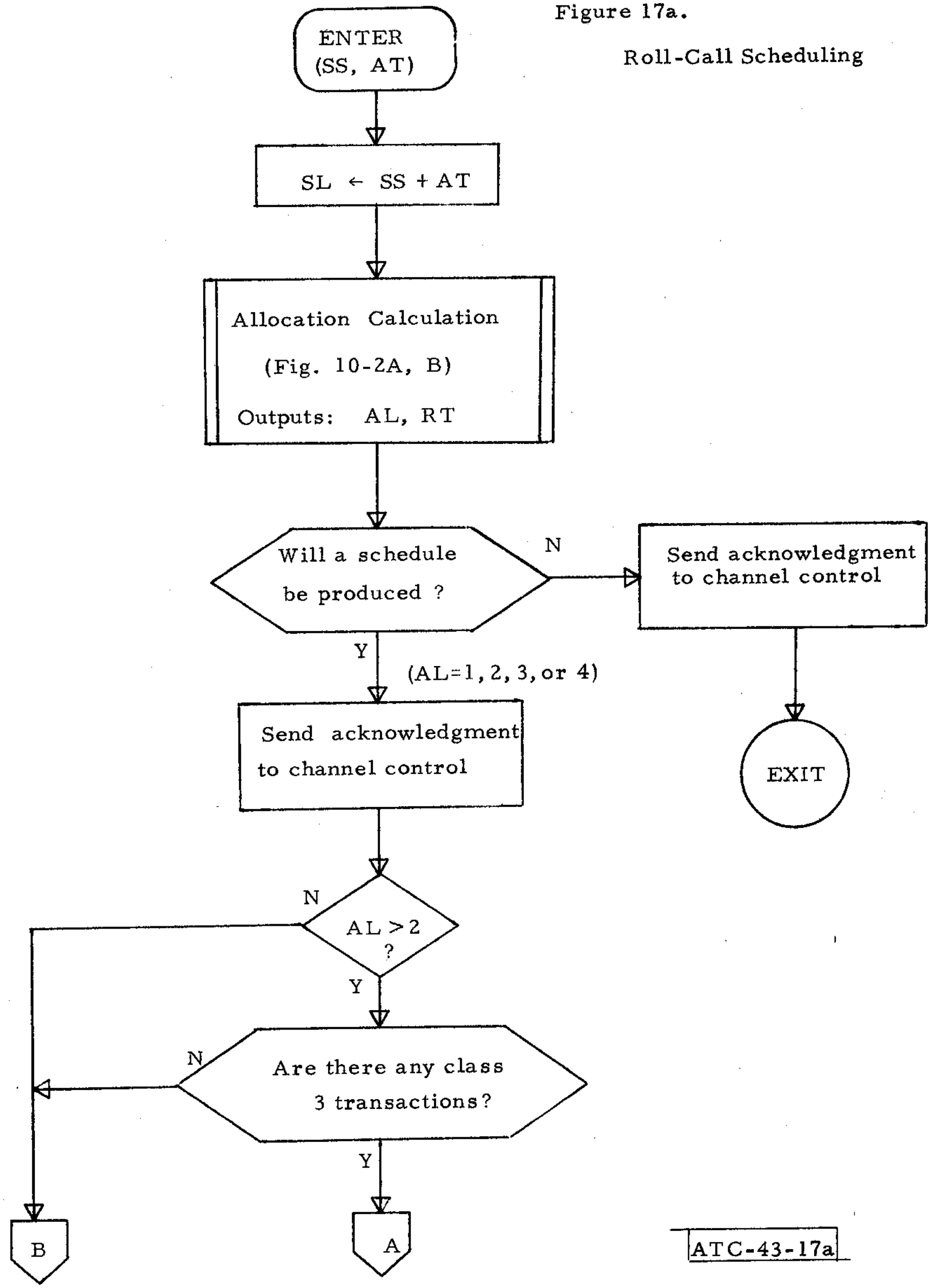
The variable names used in the flowcharts have the following meanings:

SS	Schedule start time
AT	Available time
SL	Schedule limit
AL	Allocation level
RT	Remaining time
SET	Schedule end time
S1	Priority channel time sum
S11	Synchronized channel time sum
S2	Normal channel time sum
S3	Uplink ELM channel time sum

CTSL	Channel time estimate for a standard transaction with a long reply
CTE	Channel time estimate (for a given transaction)
M	Internal segment counter
N	Number of segments in a given uplink ELM transaction
ATL	Active target list
T	Tentative interrogation start time
L	Tentative reply start time
CTT	Cycle test time
TRD	Transponder reply delay
X	Dummy variable
S	Tentative interrogation end time
ISO	Isolation delay parameter
DELAY	Dummy variable
RRP	Reply repetition period parameter (for downlink ELM replies)

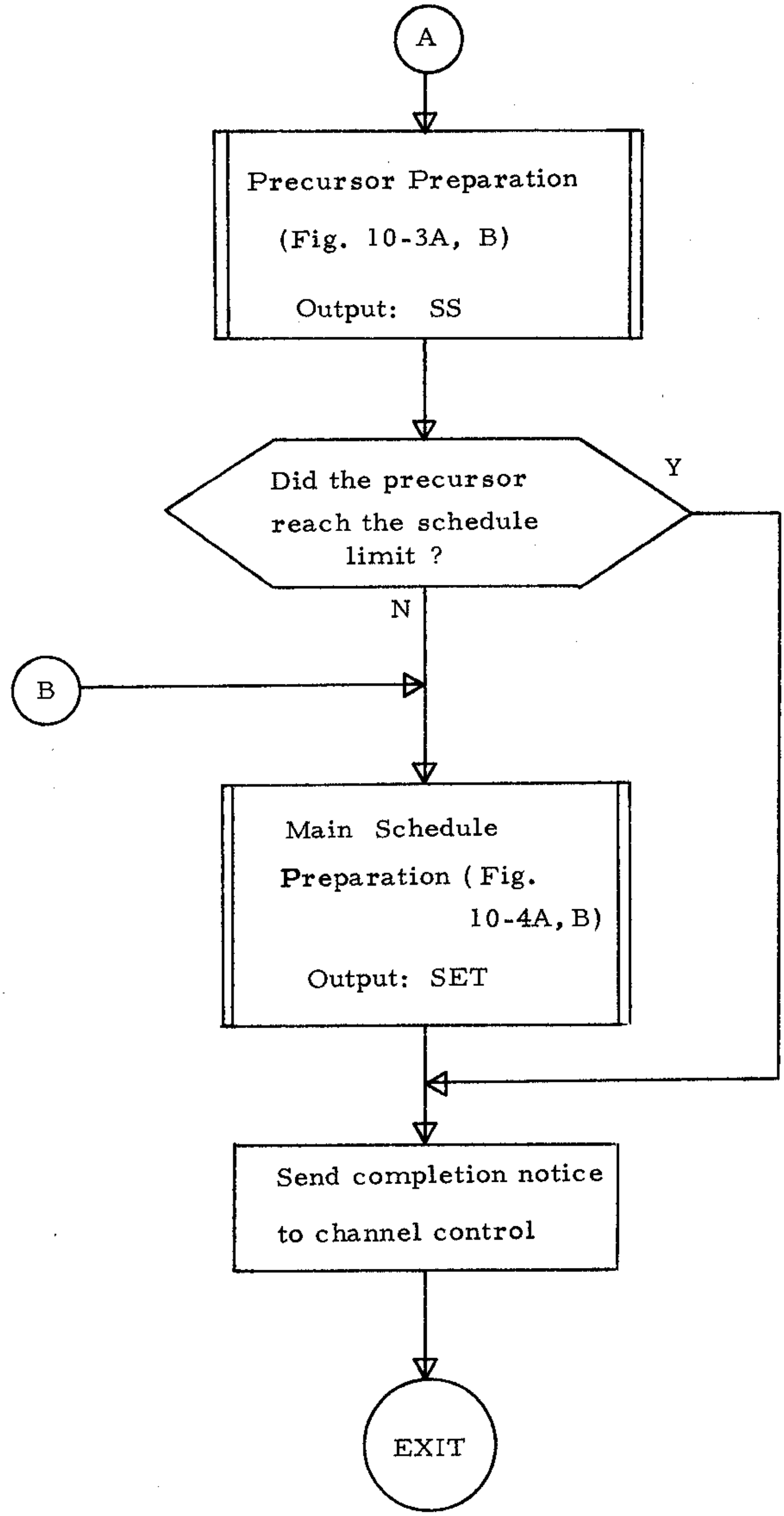
Figure 17a.

Roll-Call Scheduling



ATC-43-17a

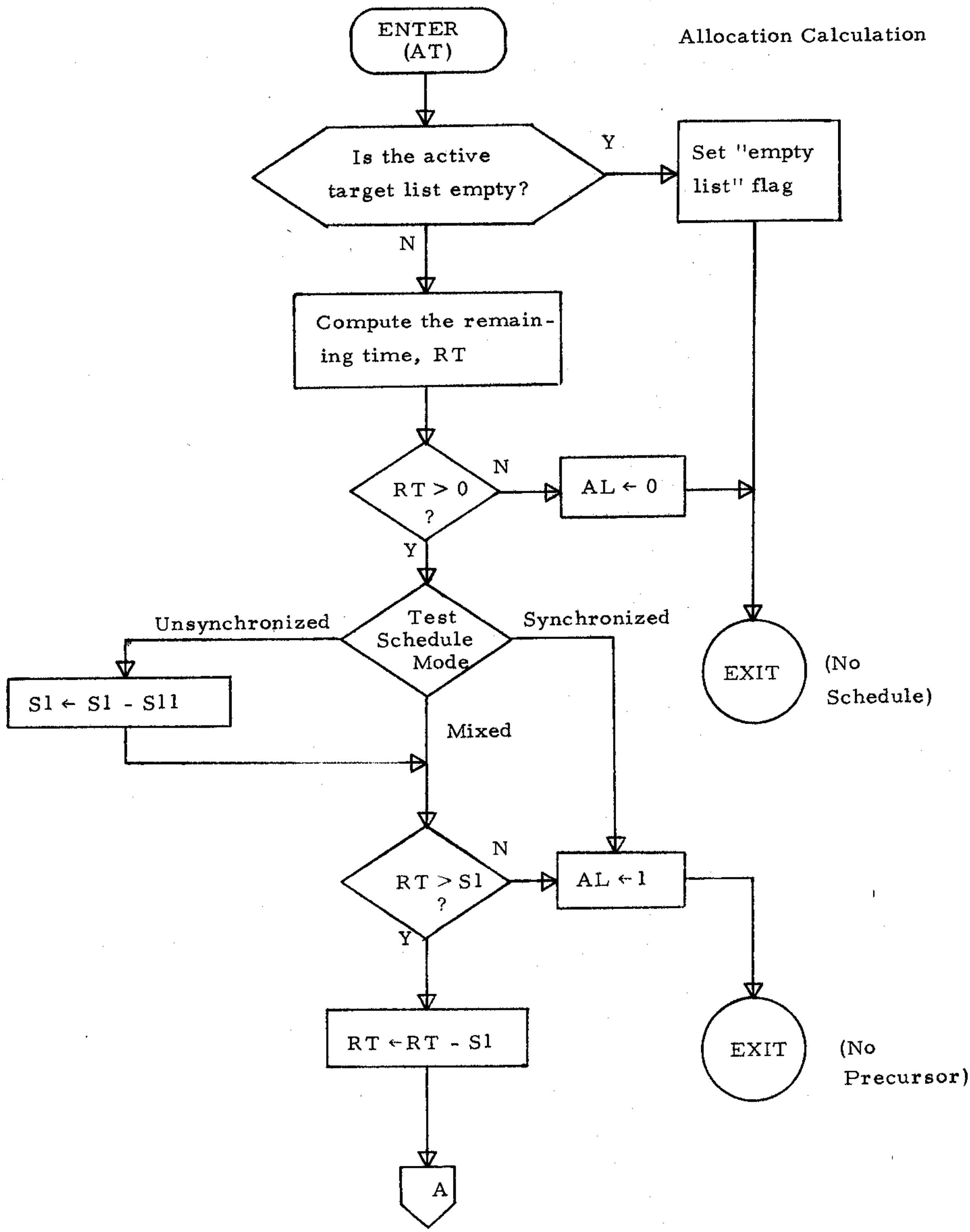
Figure 17b.
(Roll-Call Scheduling)



ATC-43-17b

Figure 18a.

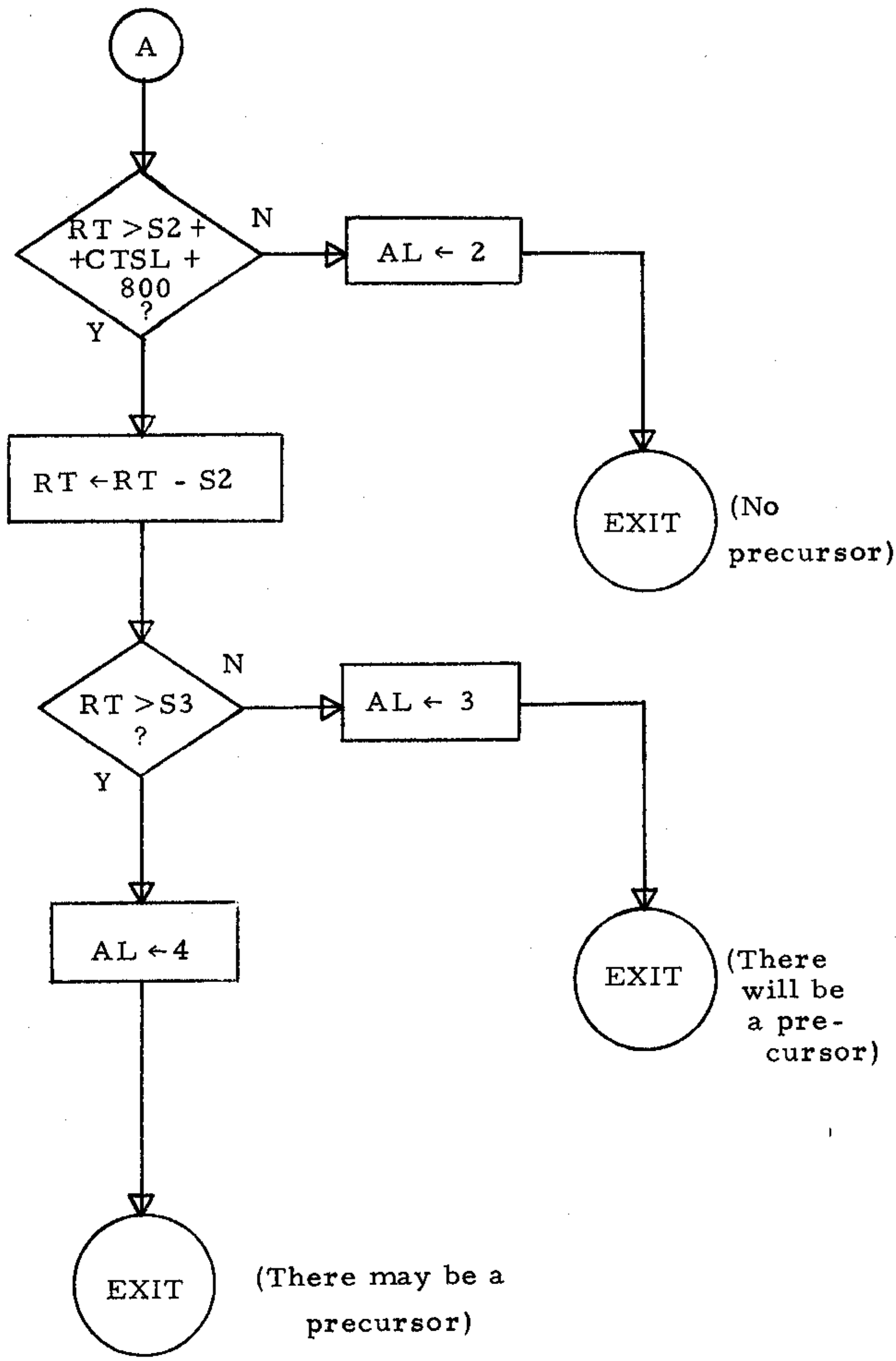
Allocation Calculation



ATC-43-18a

Figure 18b.

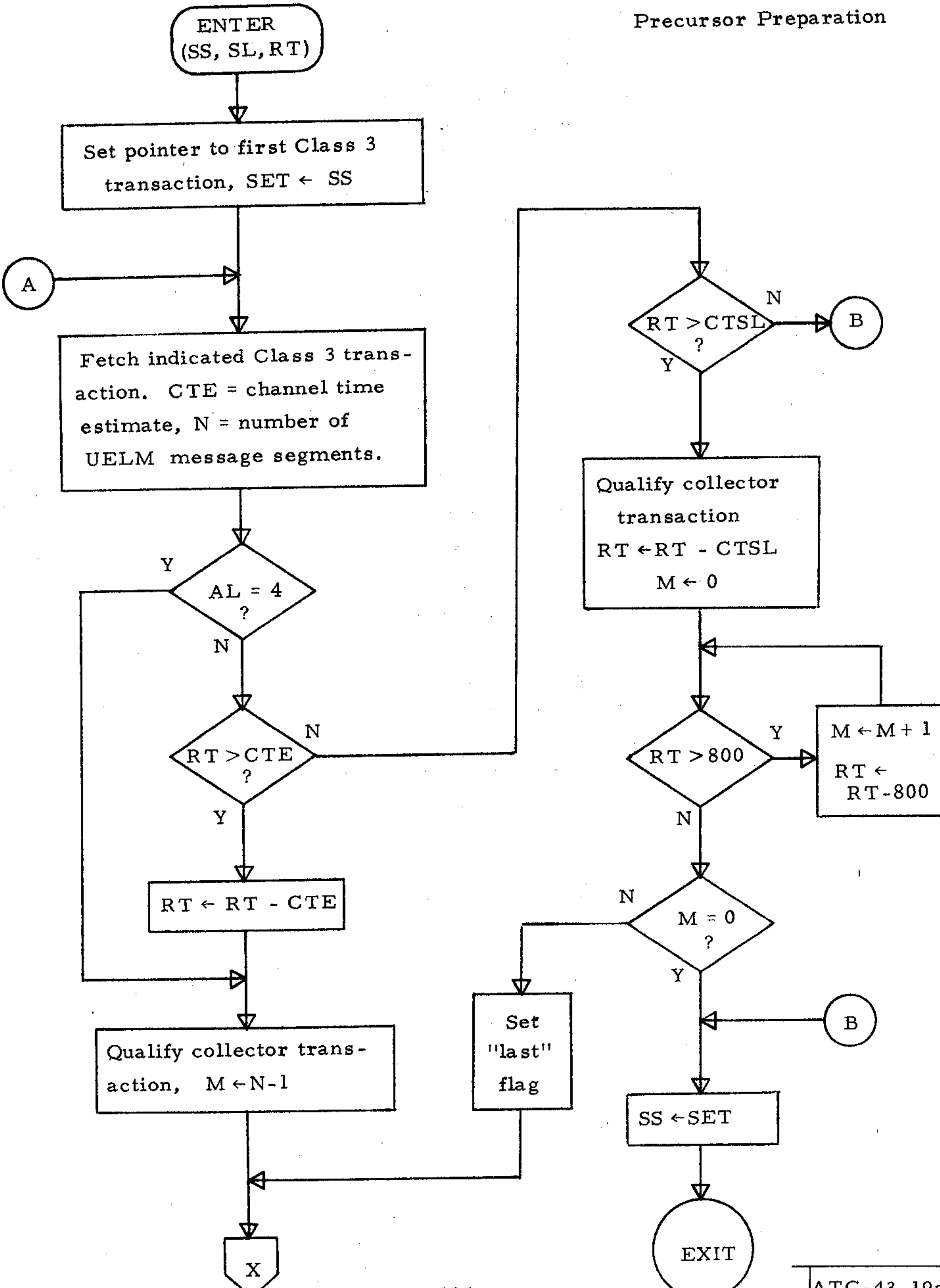
(Allocation Calculation)



ATC-43-18b

Figure 19a.

Precursor Preparation



ATC-43-19a

Figure 19b.

(Precursor Preparation)

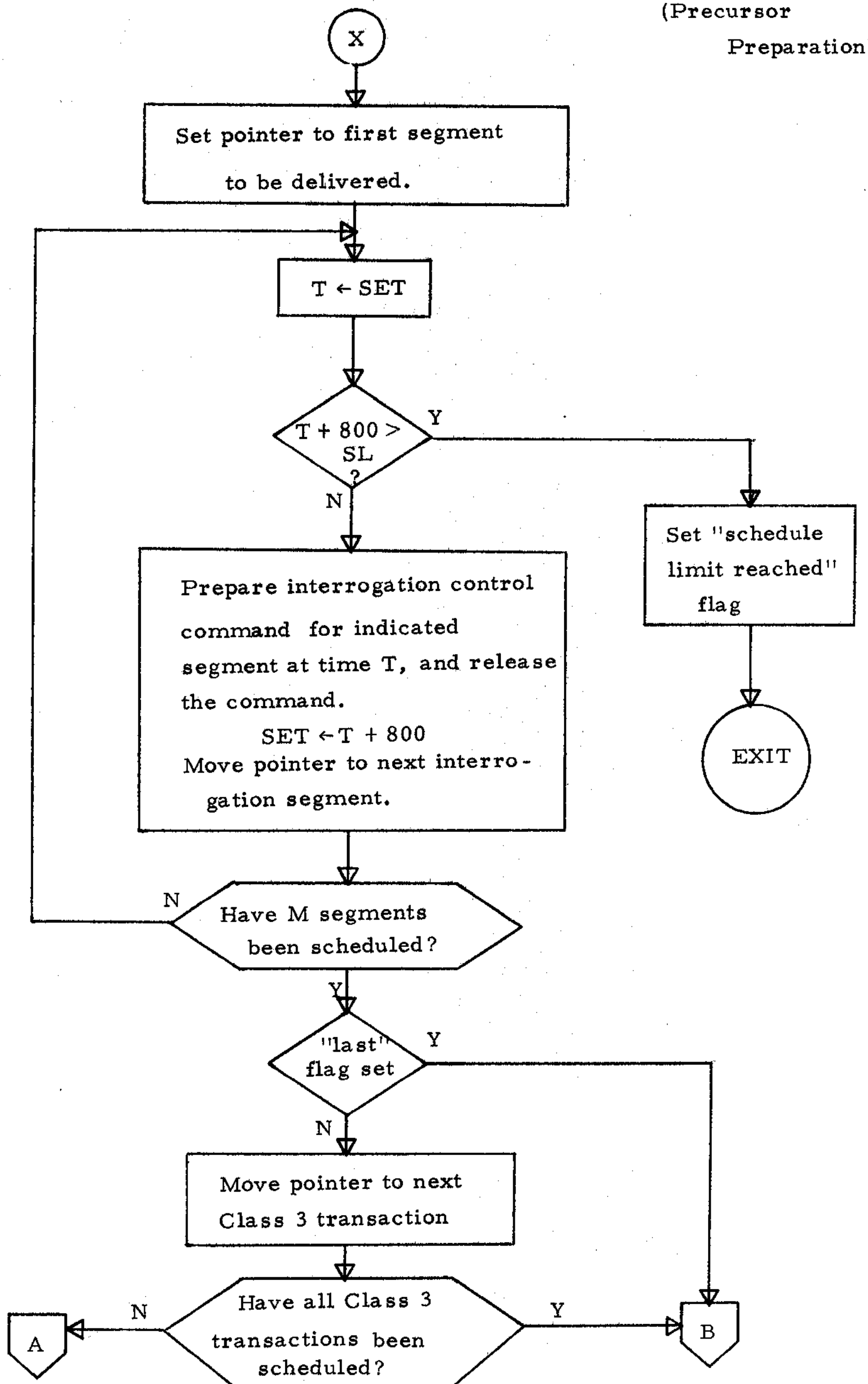
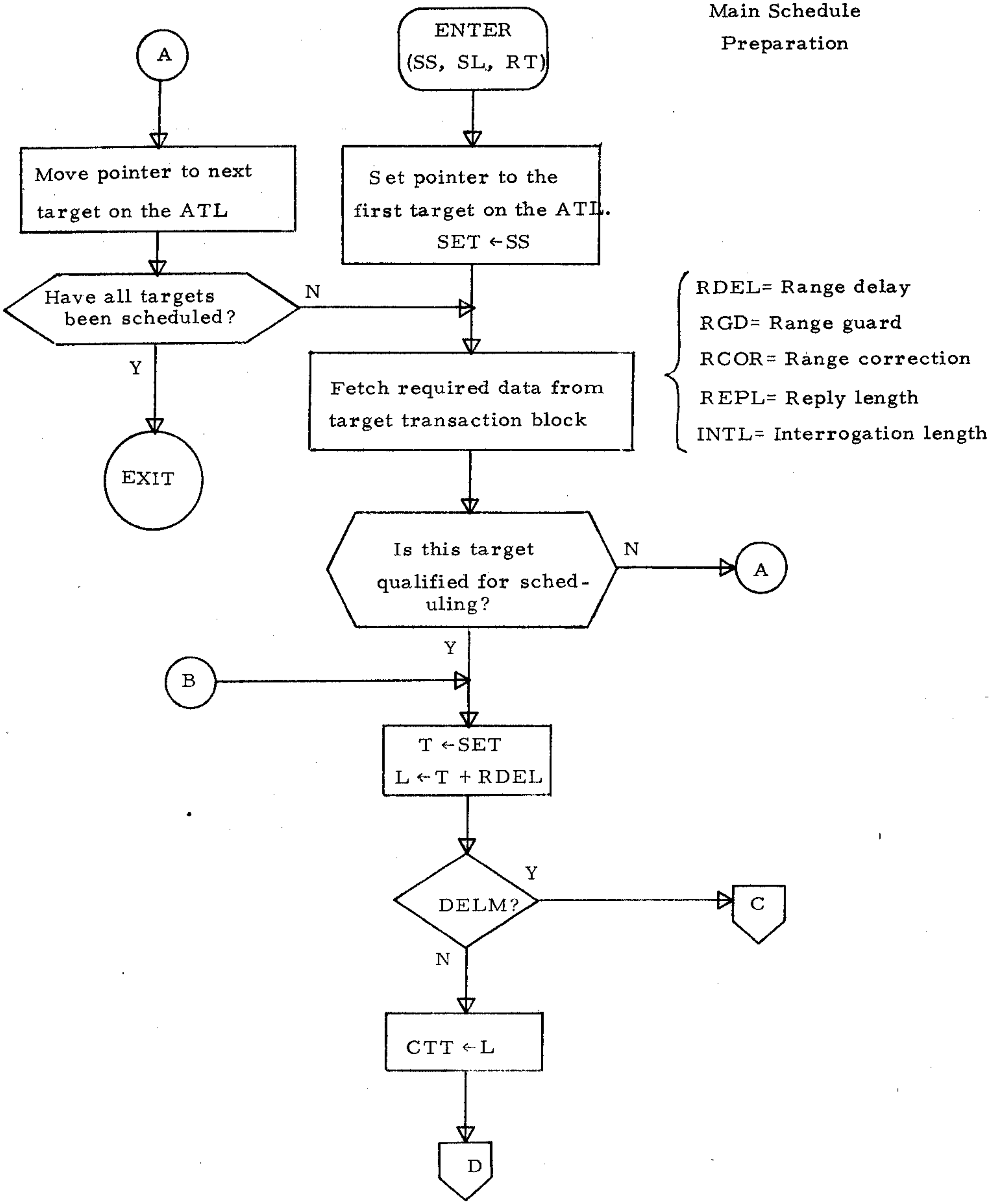


Figure 20a.

Main Schedule Preparation



ATC-43-20a

Figure 20b.
(Main Schedule Preparation)

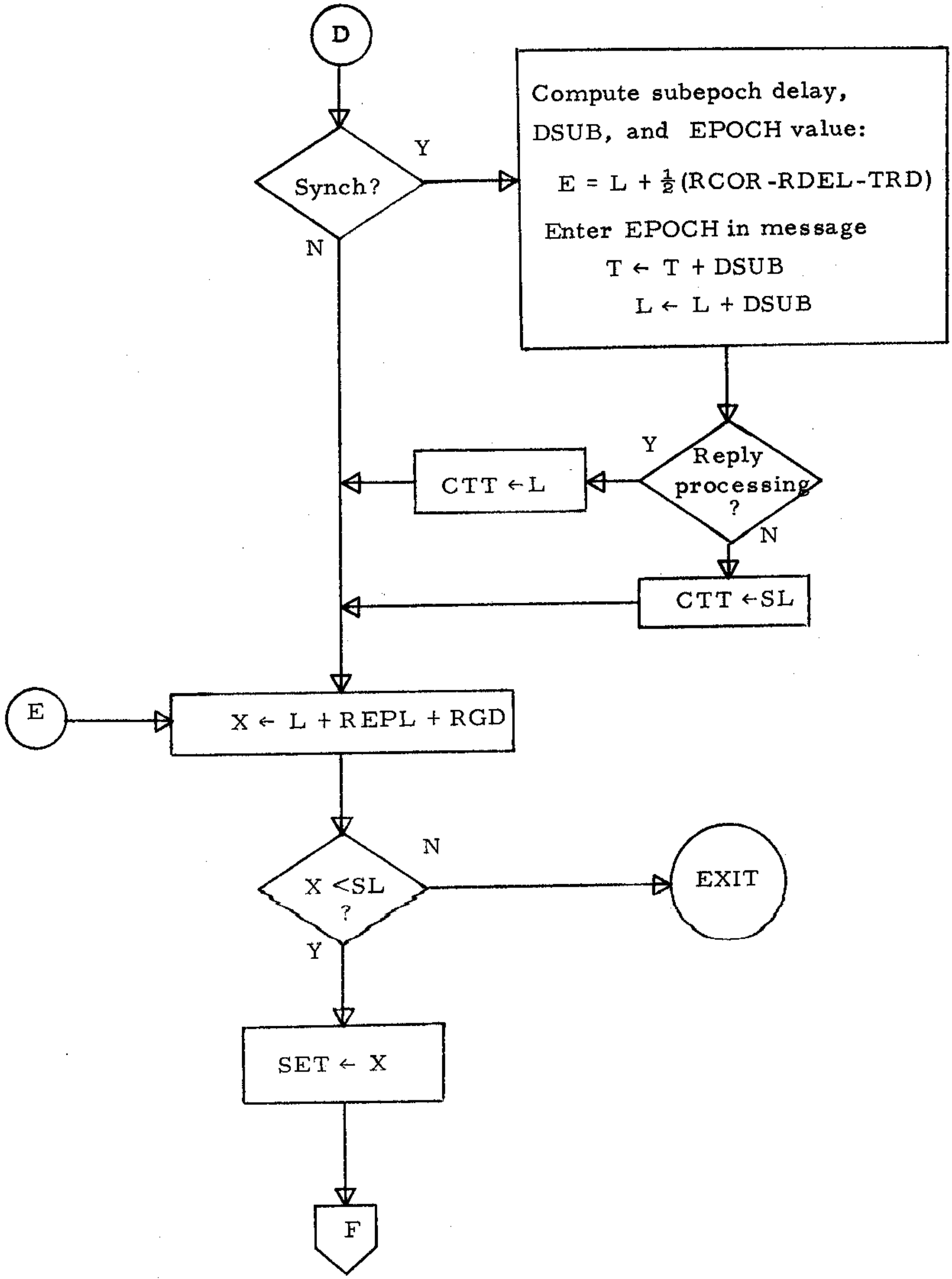


Figure 20c.

(Main Schedule Preparation)

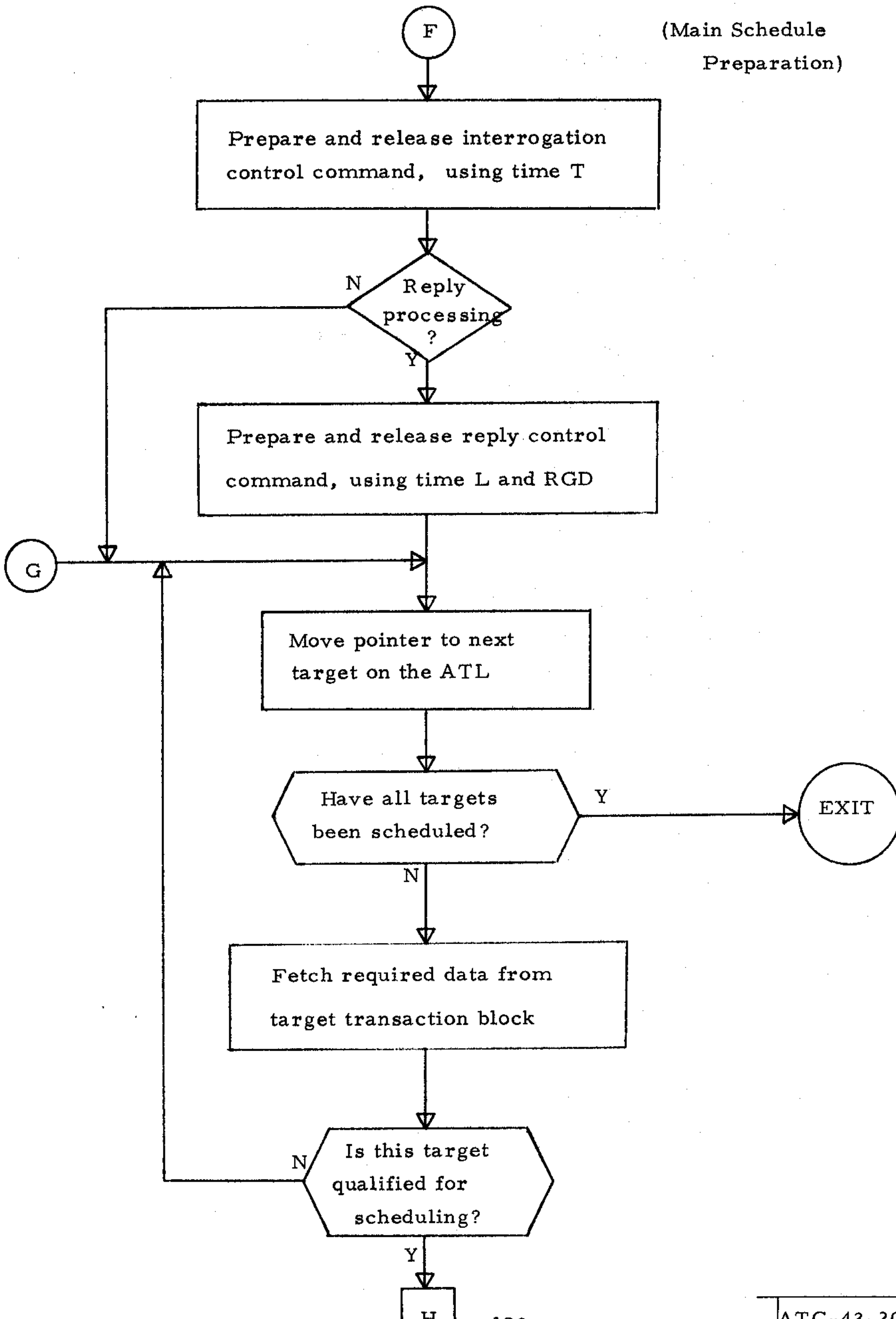
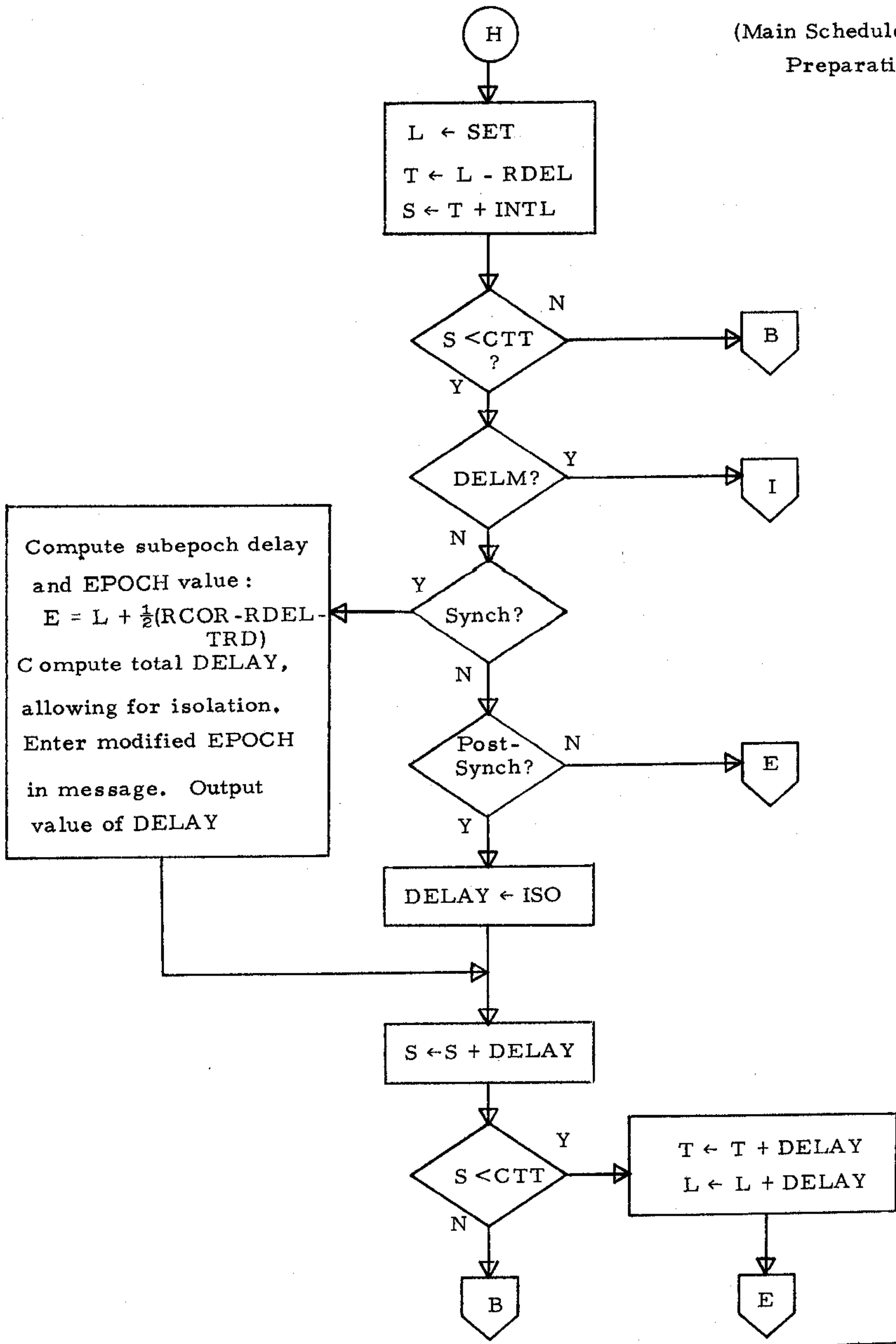


Figure 20d.

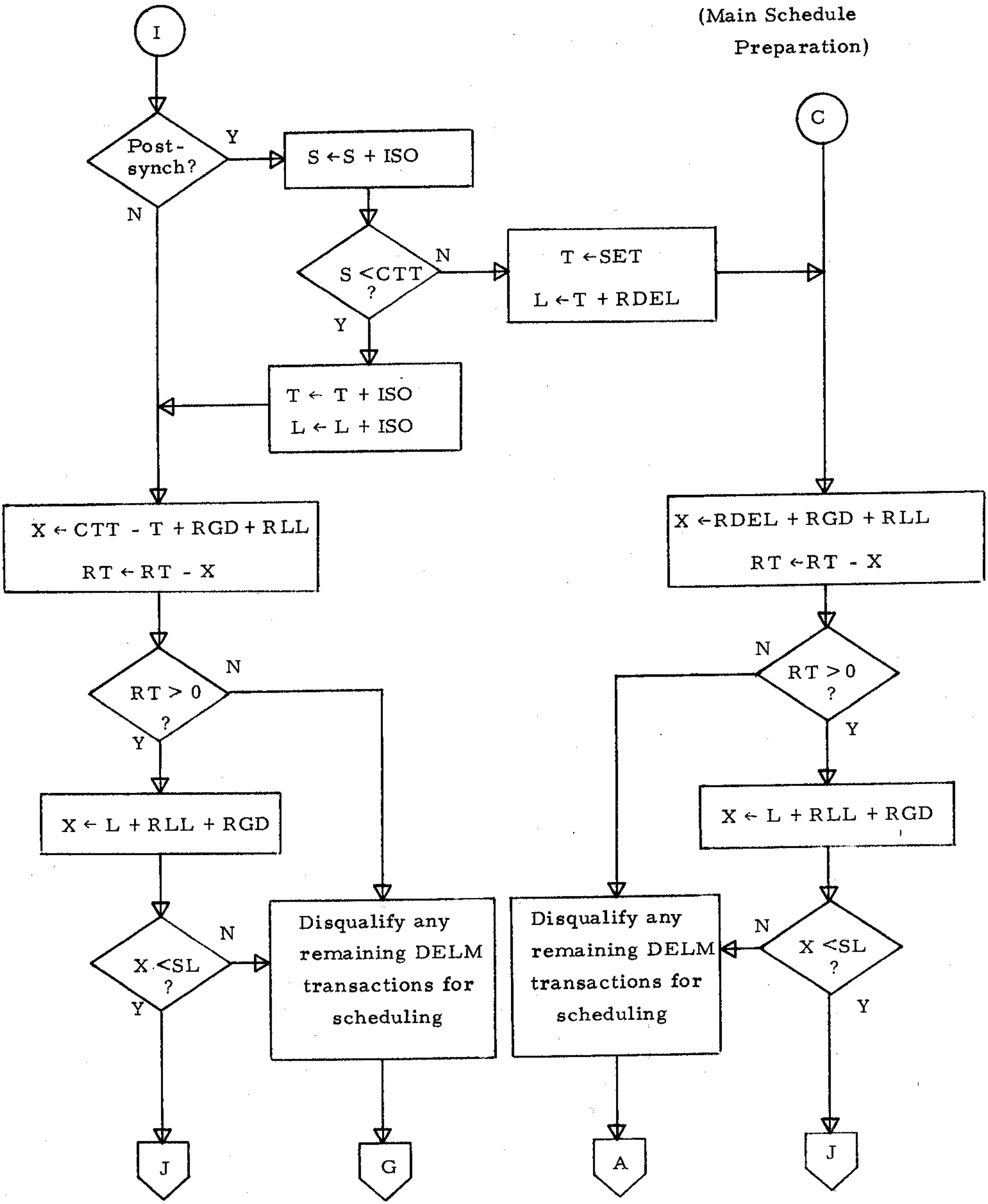
(Main Schedule Preparation)



ATC-43-20d

Figure 20e.

(Main Schedule Preparation)

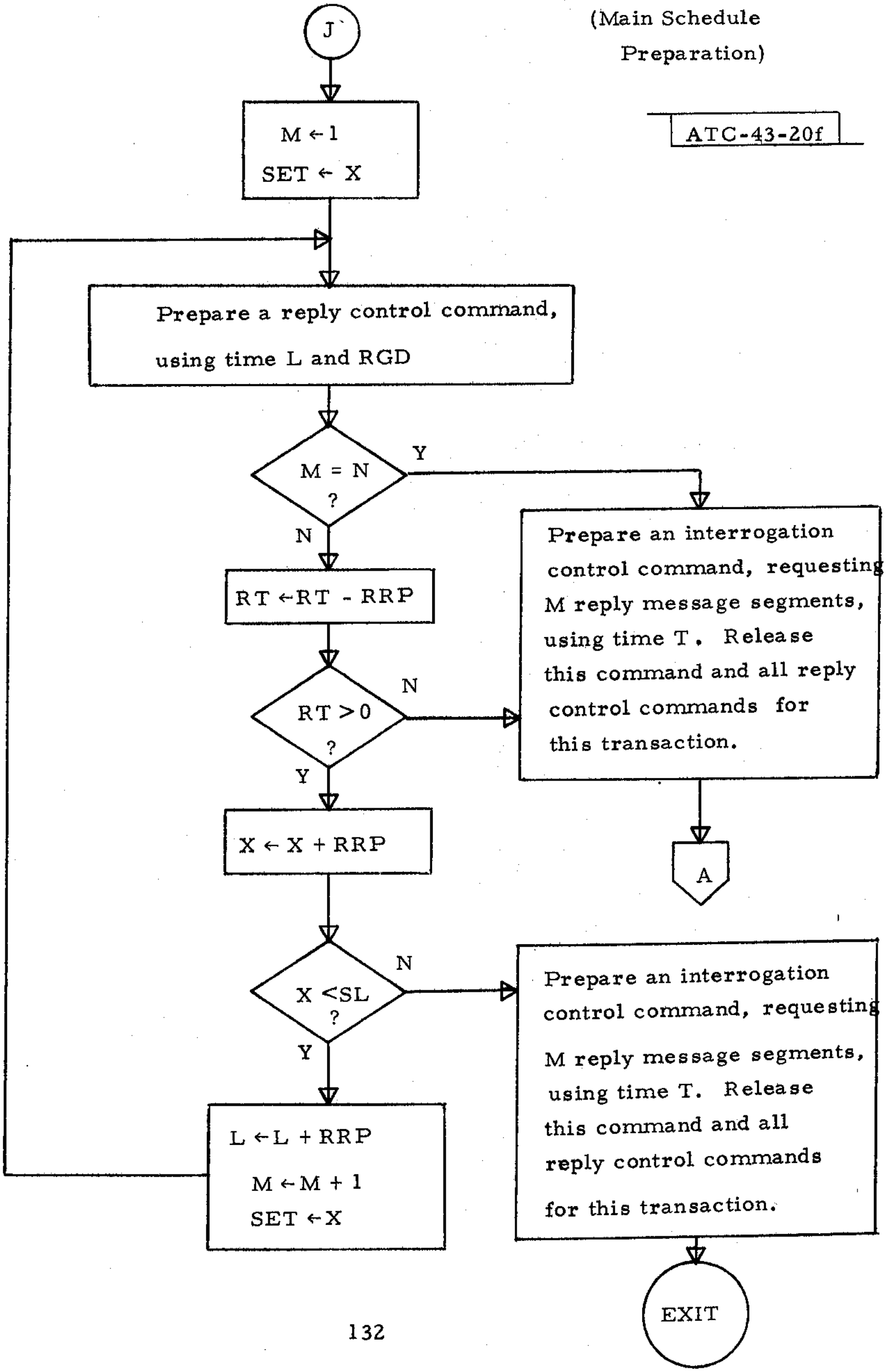


ATC-43-20e

Figure 20f.

(Main Schedule Preparation)

ATC-43-20f



PART III

TRANSACTION PREPARATION, TARGET LIST UPDATE AND TRANSACTION UPDATE

III-1. Overview

The channel management subfunctions of transaction preparation, target list update, and transaction update are discussed together in Part III, which completes the report begun with Parts I and II. These subfunctions are specified separately in the DABS engineering requirement (Reference 1), but are treated together here because of the close relationship and high degree of interplay between them. The general principles which have guided their design are explained below, in Section III-2.

The central file with which all three tasks are concerned is the active target list. As described in Section II-2, this list contains data on each target currently within the antenna beam and with channel activity still pending. The data for a given target is organized into a "target transaction block." The data structure intended for this block is described in Section III-3, and the preparation of the block, from data contained in the surveillance file, is described in Section III-4.

As discussed in Part I, channel control activates transaction preparation intermittently, to generate data for targets about to enter the antenna beam. Transaction preparation is implemented in the scan-to-scan processor, where it has ready access to the surveillance file. The target transaction blocks which contain all the data necessary for subsequent scheduling operations, are then sent to the channel management processor to be merged onto the active target list. The structure of the active target list is described in Section III-3, supplementing the discussion of this list in Part II.

Section III-5 is devoted to a discussion of the target list update function, which merges new data blocks onto the active target list, while deleting the blocks of completed targets and those which have fallen behind the beam. These latter blocks are grouped into a "released target list" which accompanies the reply data blocks themselves to the scan-to-scan processor for the benefit of the data link and surveillance processing functions. Flowcharts for transaction preparation and target list update are provided in Section III-14, together with flowcharts for transaction update.

The remainder of Part III is devoted to transaction update, whose many sub-tasks are treated in separate sections. Most of these sections trace an entire protocol, such as the handling of Comm-B messages (Section III-11) through the channel management function, as well as describing the role played by transaction update in this protocol. Sections III-6 and III-7 treat the preparatory stages of transaction update, and Section III-14 offers an illustrative transaction update algorithm and flowchart.

III-2. Design Principles

The sensor has a number of responsibilities; some to the ATC system on the ground, and some to the aircraft participating in the ATC system. The responsibilities which fall upon channel management may be broadly classified into two classes: surveillance and communication. In order to discharge these responsibilities, channel management is given complete control of the RF ground/air channel. Its tools are the various transactions afforded by the DABS signal formats and the link protocol itself.

The signal formats are specified in Reference 2, and it will be assumed that the reader of this report is familiar with that document in some detail. DABS transactions, which are particular pairings of interrogation and reply signals, have been discussed frequently in the preceding Parts of this report, and they are discussed again below, from the point of view of the objectives they are intended to accomplish.

The major surveillance responsibility is the provision of a position update, on every target each scan, to the ATC system (i.e., NAS, including ARTS facilities at terminals) and to the local IPC processor. The sensor itself, of course, measures slant range and azimuth, while the aircraft reports its altitude in the normal surveillance reply. For certain aircraft, the sensor also performs a kind of surveillance service to the aircraft. The service is called "synchronous service", and it is rendered to aircraft designated by network management. Although more than one sensor provides position updates to ATC on a given target, usually only one sensor provides synchronous service to that aircraft. Synchronous coverage is regulated by network management, which sets a "synchro bit" for appropriate targets in the surveillance file. Synchronous service is defined as the delivery of synchronized interrogations, including an altitude echo (ALEC) field, to an aircraft.

A special service, which results in ATCRBS identity replacing altitude in a surveillance or Comm-B reply, may also be considered a surveillance responsibility, although it combines the features of both surveillance and communications. The request for ATCRBS identity transfer (on the downlink) may come either from the ground or from the pilot. A ground request, recognized by the setting of the "AI bit" in the surveillance file, will usually be made for the purpose of correlating the DABS and ATCRBS data on a target which may be received at a control facility. A pilot request, recognized by the setting of the "A bit" in a surveillance or Comm-B reply, will usually be made in an emergency situation, in which the ATCRBS identity field is used as a special data link channel with a limited message vocabulary (e.g., emergency, radio out, etc.).

Communications responsibilities relate to the use of the DABS data link, under ground or pilot request. The ground can initiate the transfer of a short, single-segment message, using a Comm-A interrogation on the uplink or a Comm-B reply on the downlink. The ground can also initiate a multiple-segment, uplink extended length message (ELM) with a series of Comm-C interrogations. Similarly, the aircraft can request the transfer of a single-segment short message or an ELM of one or more segments. Pilot-initiated short messages, transferred on the downlink with a Comm-B reply, are requested by the setting of the "B bit" in a surveillance or Comm-B reply. Downlink ELM's, transferred by means of one or more Comm-D replies, are requested by the use of the "D bit" and the "DCOUNT" field (number of message segments) in either an unsynchronized surveillance reply or in a Comm-B reply. Pilot-initiated exchanges are finally terminated by the sensor, which sets a "CB bit" (Clear Comm-B) in a surveillance or Comm-A interrogation, or sends a special Comm-C interrogation to clear the downlink ELM (Clear Comm-D). An analogous Comm-C interrogation is used to terminate an uplink ELM (Clear Comm-C).

An additional communications responsibility involves the so-called "PBUT (pilot button) channel." A dedicated two-bit field, PBUT, present in all surveillance and Comm-B replies, is used to convey pilot acknowledgment of certain uplink commands. These commands, generally associated with IPC, are delivered using a Comm-A interrogation whose message field (MA) contains a specific request for the pilot acknowledgment. The sensor must report the meaning of the PBUT field (if not blank) to all ATC facilities and then send a "clear PBUT" indication, by means of the "CP bit" in any surveillance or Comm-A interrogation.

It will often fall upon the sensor to exercise several of the responsibilities, described above, to a given target in a single scan. A whole series of transactions will then be needed, many of which can be anticipated (from data in the surveillance file, including the active message sublists) in advance of the brief time interval during which the target will be visible in the coming scan. Other responsibilities may become apparent while the target is within the antenna beam, if the appropriate bits are found to be set in replies received to planned interrogations. Therefore, channel management must have the capability to add transactions (or modify planned transactions) dynamically before the target leaves the beam. The planning of transactions in advance is the function of the transaction preparation. The addition or modification of transactions while the target is within the beam is one of the functions of transaction update. Management of the list of transactions pending for targets presently within the beam is the function of target list update.

Certain general principles of priority govern the activities of these functions, and hence, regulate the initiation of transactions designed to meet the sensor's responsibilities. Transactions in support of surveillance objectives are generally given priority over those whose major task relates to communications. It must be realized, however, that the signal formats themselves are sufficiently flexible to allow a single transaction to accomplish several objectives simultaneously. For example, a transaction may consist of a Comm-A interrogation answered with a Comm-B reply, providing position update data while transferring one message in each direction on the link. In addition, the reply can contain a PBUT response and have the "A bit" set, requesting another interrogation whose reply will contain ATCRBS identity.

Transactions themselves are classified as "priority" or "normal" by channel management. Any transaction supporting the basic surveillance objectives listed above will be a priority transaction. The position update function is considered accomplished when range, azimuth and altitude have been once measured successfully, and this will usually occur with the first transaction of each scan for a given target. The first transaction is called the "surveillance transaction." Synchronized transactions (one per scan for designated aircraft, normally) and transactions requesting ATCRBS identity are always priority transactions.

Transactions which involve Comm-C interrogations and Comm-D replies are called "ELM transactions," since they always relate to ELM activity. ELM transactions are always classified "normal." Non-ELM transactions are called "standard transactions," and they may be priority or normal.

Comm-A messages and Comm-B message requests from the ground are divided into two classes: "high-priority" (or "urgent") and "standard" (or "normal"). If a (standard) transaction involves a message which has "high-priority," then the transaction itself is classified priority. The message may be a Comm-A or a Comm-B message, or both may be involved.

Uplink messages and downlink message requests are obtained from the active message sublists associated with the surveillance file. These sublists are managed by data-link processing, which places high-priority messages ahead of standard messages on the sublists. When transactions are formulated by channel management, this ordering is preserved, so that high-priority messages will be delivered ahead of standard messages. Moreover, channel management exhausts the message sublists which contain Comm-A messages and Comm-B message requests, before it formulates ELM transactions for a given target.

Thus, the sequence in which transactions are prepared for a given target, before that target enters the beam, conforms to definite priority rules, which affect the nature of the service provided by the sensor. Similar rules affect the dynamic formulation of transactions while the target is in the beam.

Priority rules are exercised in a different way when different targets compete for channel time. Priority, standard transactions are scheduled in preference to normal, standard transactions, and standard transactions are preferred over ELM transactions. The operation of these rules is explained in detail in Section II-4, in connection with the allocation portion of roll-call scheduling.

III-3. Structure of the Target Transaction Block and the Active Target List

The concept and general structure of the active target list have been described in Section II-2. The function of this list is to provide channel management with a convenient source of data on all targets which are "active," i.e., within the antenna beam and with transactions pending. The requisite data comes from the surveillance file, which includes the active message sublists, and it must be extracted and reorganized into "target transaction blocks" by transaction preparation, before it can be transferred from the scan-to-scan processor to the channel management processor, where it is assimilated into the active target list.

A target transaction block contains data on a single target, some of which is copied from the surveillance file and some of which is generated, by transaction preparation, in the form of a series of "transaction records." Each transaction record corresponds to a specific transaction which is planned in advance of the appearance of the target in the beam. The information in a given transaction record is used by channel management in the scheduling of that transaction and in the evaluation of the resulting reply. The transaction records are organized in the target transaction block in a way which corresponds to the intended order of their execution in roll-call schedules.

In addition to the transaction records, the target transaction block has a header, called the "target record," which contains data and control information that relate to the target itself, and the progress made in accomplishing the DABS surveillance objectives during the course of the scan.

The data content of the target transaction blocks will be discussed in connection with their preparation in Section III-4. This Section is devoted only to the structure and organization of these blocks and of the active target list.

The intended structure can best be explained by an illustration using pointers and linked-list techniques. The target transaction block is accessed through its header, the target record, and blocks are linked together by means of forward and backward pointers from one target record to others. Internally, the transaction records form a linked list, and each record has a pointer its successor. The target record will have a pointer to the first transaction record, which always corresponds to the surveillance transaction. Each transaction record has an additional pointer to the target record, so that target data contained in that record (e.g., DABS identity) can be accessed easily, along with transaction record data, by roll-call scheduling.

For every target new to the active target list, the surveillance transaction will be the first transaction attempted. When this transaction is successful, the next transaction in the target transaction block will be attempted, and so on. This procedure is facilitated by the "current transaction indicator," which is another target record pointer which links the target record to the record of the next transaction to be executed. The pointer which links the target record of a newly-prepared target transaction block to the surveillance transaction record could be modified to serve as a current transaction indicated, but it seems best to use separate pointers so that subsequent users of the target transaction block (e.g., data link processing) can thread through the entire block easily.

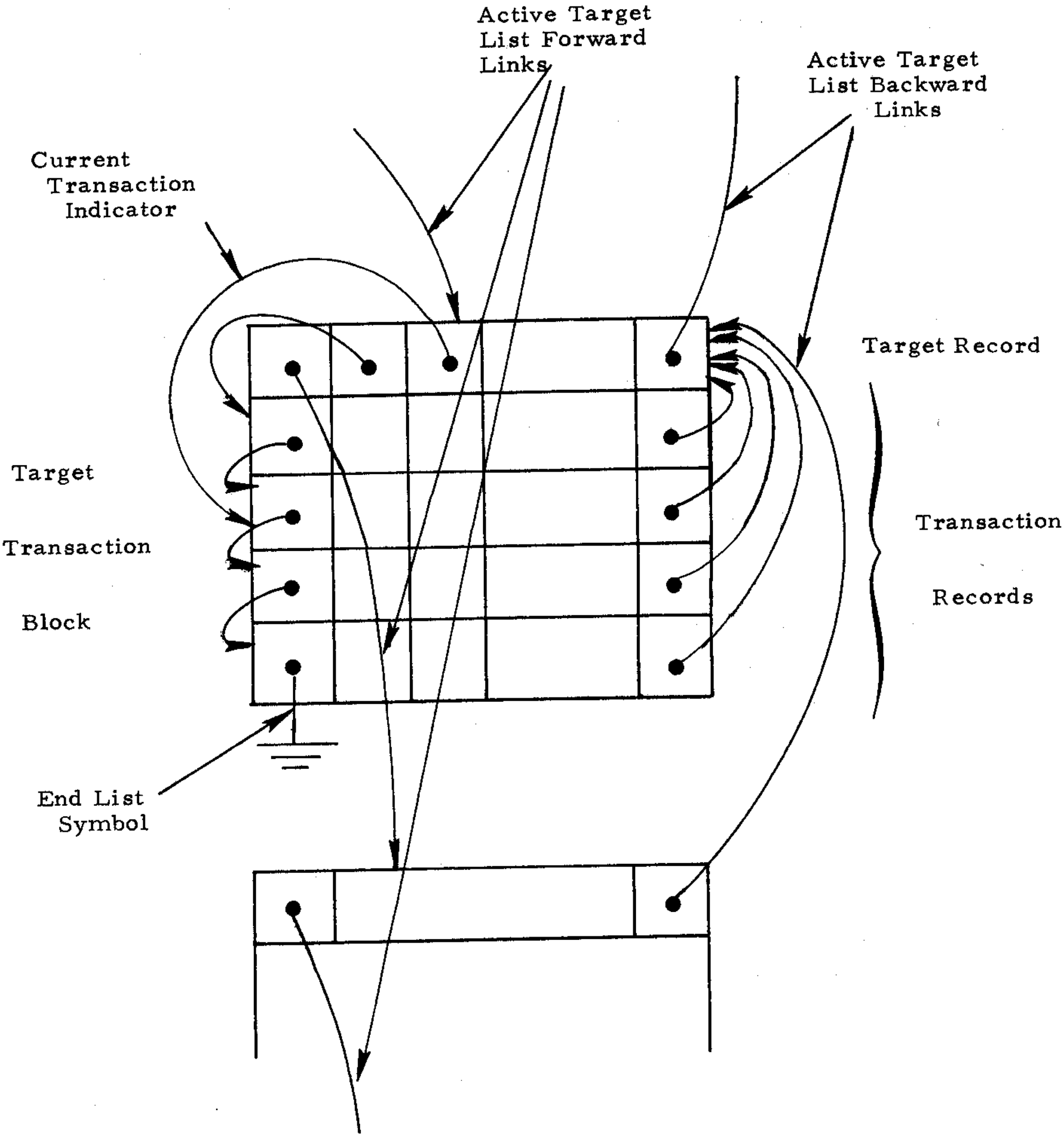


Figure 21. Active Target List Structure

The data structure described above is schematically illustrated in Figure 21. Obviously, sequential allocation may replace some of the linked allocation shown here, in which case some of the pointers would be replaced by equivalent data fields containing indices.

III-4. Preparation of the Target Transaction Block

This Section is essentially a description of the transaction preparation subfunction, as specified in Reference 1. Transaction preparation is a relatively leisurely operation, compared to the other channel management tasks, and it is performed in the scan-to-scan processor under control of channel control. The basic purpose of transaction preparation is to "set things up" for the other subfunctions, doing as much as possible in advance of the appearance of the targets in the beam. Transaction preparation is a straightforward job, and most of its features can be understood on the basis of a general familiarity with channel management. Some of the finer details, however, will be best appreciated after the other topics covered in this Part have been studied, namely target list updating and transaction updating. The interplay between the three subfunctions will be emphasized in Sections III-8 through III-13, which trace individual surveillance and data link features through the system.

Transaction preparation is activated by channel control, which provides a "new azimuth limit" with each enabling command. The value of the new azimuth limit which accompanied the preceding enabling command will have been saved by transaction preparation, and this previous value will be referred to as the "old azimuth limit." When enabled, transaction preparation must extract data from the surveillance file on all targets between the old and new azimuth limits, prepare target transaction blocks for these targets, and send the blocks to the transaction buffer for later transfer to the channel management processor by target list update. The new azimuth limit is set to include all targets which will become visible for the first time, in the present scan, in the next forthcoming DABS period. The computation of the new azimuth limit is explained in Section I-5.

The input data for transaction preparation is contained in the surveillance file, the major file of the scan-to-scan processor. Transaction preparation is concerned only with known DABS targets, represented in the surveillance file as discrete-code DABS tracks. Tracks in the file are grouped by sectors (11.25° in width), on the basis of predicted azimuth, and each record in the file has a field which contains the target's "earliest likely azimuth." Earliest likely azimuth equals predicted azimuth, reduced by a computed term representing azimuth prediction uncertainty. This quantity is used to fetch targets from the surveillance file, and its computation, performed by surveillance processing, contains an adjustable system parameter, which will be set to balance the effects of missing interrogation opportunities because a target is brought onto the active target list late, and wasting channel time by bringing the target on too soon.

The exact data structure to be used for the surveillance file is not specified in Reference [1], except that it must be possible to perform surveillance processing, data link processing and some network management tasks on a sector basis. It must also be possible to execute an efficient between-limits search on earliest likely azimuth, so that transaction update can extract the targets for which this parameter lies between the old and new azimuth limits.

In addition to track data and other position-related information, the surveillance file contains data-link messages and message requests for each DABS target. These data form the active message list for that target, and are probably implemented in a data-link storage area, referenced by a pointer in the surveillance file itself. The active message list for a single target consists of three sublists, containing data on Comm-A, Comm-B and ELM transactions. The Comm-A sublist contains single-segment messages to be transferred on the uplink by means of a Comm-A interrogation. Comm-B sublist entries are requests for single-segment downlink messages (pilot-originated or ground-initiated), which will be transferred by means of a Comm-B reply. The ELM sublist contains uplink ELM messages and downlink ELM message requests. Downlink ELM message segments are stored in this sublist, when received, until the entire message is obtained and sent to ATC.

Transaction preparation begins with the preparation of the target record, which is composed of the following data fields:

1) DABS address	(24 bits)
2) Range delay	(16 bits)
3) Range guard	(16 bits)
4) Last likely azimuth	(14 bits)
5) Altitude pressure correction	(8 bits)
6) Primary indicator	(1 bit)

7) Synchro indicator (1 bit)

8) DABS lockout transition control (1 bit)

9) ATCRBS identity request (1 bit)

10) Range correction (16 bits)

11) Range completion (1 bit)

12) Azimuth completion (1 bit)

13) Target completion (1 bit)

14) Power-programming control (1 bit)

15) Remaining tries (2 bits)

16) Comm-B state (2 bits)

The first nine items are simply copied from the surveillance file; item 10 is not generated by transaction preparation at all (it is later assigned by transaction update); items 11 through 15 are generated by transaction preparation, and item 16 is copied from the target's active message list. The significance of each of these fields is given in the following discussion.

The DABS address is, of course, the 24-bit discrete aircraft identity code characteristic of a DABS target. The target range delay is the predicted two-way propagation time for the target, including transponder reply delay, based on the target's "earliest likely range." Earliest likely range, analogous to earliest likely azimuth, equals predicted range, reduced by a computed term representing range prediction uncertainty. Roll-call scheduling is always based upon this range delay, so that a listening interval for a reply will start soon enough to include the reply even if the target is at the earliest likely range. A listening interval exceeds the actual reply duration by the "range guard", which is twice the propagation time corresponding to the computed range prediction uncertainty. Range delay and range guard are 16-bit words, whose least significant bit is one range unit. The use of these quantities is fully explained in Section II-6, in connection with roll-call scheduling.

When surveillance processing computes a predicted azimuth it also makes an estimate of prediction uncertainty. As mentioned above, earliest likely azimuth is equal to the prediction minus the uncertainty. Similarly, "last likely azimuth" equals the prediction plus the uncertainty. It represents the azimuth at which the sensor should stop interrogating a target which has not responded at all in the present scan. It is used in this fashion by target list update, and last likely azimuth is replaced with an actual azimuth measurement, by transaction update, as soon as such a measurement is available.

As mentioned above, one aspect of synchronous service is the inclusion of the "altitude echo" (ALEC) in any synchronized interrogation. The purpose of ALEC is to allow the pilot to compare the value of altitude which his transponder is automatically transmitting to the ground to the altitude displayed to him visually in the cockpit (see Reference 2 for further discussion). The displayed value is corrected for barometric pressure variation by means of the altimeter setting, made available to the pilot by ATC. The DABS sensor also obtains this information from ATC, and for each region of coverage a pressure correction is computed and stored by network management. The appropriate value is included in the surveillance file for each target, and kept current as the target's location changes, by network management. This correction is copied into the target record, so that it will be available to transaction update, which adds it to the (uncorrected) altitude reported to the ground to form the value of ALEC which is retransmitted to the aircraft. An aircraft will receive synchronous service in a given scan only if the synchro indicator is set at the time.

The primary indicator, like the synchro indicator, is set by network management and varies with time as an aircraft moves through the system. A sensor is permitted to react to B bits and D bits in target replies, initiating downlink message transfers, if the primary indicator is set. Otherwise, these bits are ignored.

The DABS lockout transition control bit is set by network management in order to regulate the transition to a region which lacks coverage by DABS interrogators. This feature is discussed in Reference 2. When this bit is "0" the DABS lockout field itself will simply be copied into most transaction records, and the other case is discussed below.

The ATCRBS identity feature has already been discussed. If the ground requests ATCRBS identity, an appropriate bit (the "AI bit") is set in the surveillance file, and copied into the target record by transaction preparation.

The range correction field is ignored (or set to all zeroes) by transaction preparation, but the three "completion bits" (range, azimuth and target) will be set to "0" in the target record. These bits are records of progress in the course of a scan, and they are used by transaction update and target list update, as discussed below.

The power-programming protocol of the DABS sensor was described in Section II-3. The control bit which appears in the target record has the function of inhibiting the use of high transmitter power for close-in targets. It is set to "inhibit" by transaction preparation for any target whose range is less than the value of a system parameter (PPMINR) whose nominal value is about five miles. Another feature of the protocol is the limitation of interrogations to non-responding targets in any one DABS period. To keep track of the number of attempts made in a given period, a "remaining tries" field is provided in the target record. This field is initialized by transaction preparation. The initial value, NTRY, is a system parameter, nominally set to the value two, which will allow three tries per DABS period.

The final target record field, Comm-B state, relates solely to the status of pilot-initiated, short downlink messages. The state has three values: inactive, Comm-B pending, and clear pending. The protocol used for these messages allows their transfer (including the initial request and final clearing) to extend over several scans, and the Comm-B state variable provides the memory required by channel management, regarding the progress achieved in preceding scans. The Comm-B state is modified by transaction update during the course of the scan and copied into the active message list header by data link processing after the target has left the beam. The variable is then picked up again, from the active message list header, and placed in the new target record (in the next scan) by transaction preparation. The Comm-B protocol itself is the subject of Section III-11.

After establishing the target record, transaction preparation must generate a transaction record for each transaction required to accomplish all the activity pending for the target. At least one transaction is always required, in order to obtain a position update. A second transaction is required for any target designated to receive synchronous service. If the ground wishes to obtain ATCRBS identity, a special transaction must be prepared for this purpose, since the first transaction (called the "surveillance transaction") and any synchronized interrogation must request aircraft altitude. Altitude and ATCRBS identity are both 12-bit fields, and they are multiplexed in non-ELM replies. The surveillance transaction demands altitude, as a part of the position update for the given scan, and synchronized replies always contain altitude for the benefit of listening aircraft, which use these replies for collision avoidance purposes (see Section II-7).

If an ATCRBS identity request is present for a target designated for synchronous service, then a minimum of three transactions will be prepared, in the order:

1) Surveillance transaction; unsynchronized, reply containing altitude.
2) "A-transaction"; unsynchronized, reply containing ATCRBS identity.
3) "S-transaction"; synchronized, reply containing altitude.

All of these transactions are classified as "priority" transactions, and together they discharge the surveillance requirements of the sensor, as far as these may be foreseen before the target enters the beam. Either the A-transaction, the S-transaction or both may be absent for a given target. The need for an A-transaction may arise while the target is within the beam if the pilot makes such a request in an earlier reply. The procedure followed in all cases is treated in Section II-9 where further details will be found.

The contents of the active message sublists will determine the data link activity which is pending for the particular target whose target transaction block is under preparation. The link protocols for all the DABS message types, and the uses of the message fields which control the protocols, are specified in detail in Reference 2. These features are not defined here, but a familiarity with them will be assumed in the following discussion.

A basic objective in transaction preparation is to combine the surveillance and communications tasks of the sensor in the minimum possible number of transactions. Any non-ELM interrogation, synchronized or unsynchronized, may contain a Comm-A message field. Such an interrogation is called a "Comm-A interrogation," while one without such a message is called a "surveillance interrogation." Surveillance interrogations are 56 bits in length, while Comm-A interrogations contain an additional 56-bit message field. Non-ELM replies may contain a Comm-B message field, provided they are unsynchronized. When the 56-bit message field is included, they are "Comm-B replies;" without it, they are called "surveillance replies."

One, two or three transactions are required for surveillance, as explained above. Any or all of these transactions can be designed to transfer Comm-A messages on the uplink, by specifying Comm-A interrogations. At least one transaction (the surveillance transaction) is unsynchronized, and it can be designed to transfer a Comm-B message on the downlink, by specifying a Comm-B reply. If an A-transaction is required, it too can be given a Comm-B reply, if two Comm-B message requests are found in the active message list.

If there are more Comm-A messages, and/or Comm-B message requests than can be accommodated by means of transactions already required for surveillance purposes, then additional "standard" transactions (i.e., non-ELM transactions) are provided to exhaust the Comm-A and Comm-B sublists. These extra transactions will all be unsynchronized, and hence each can pair a Comm-A interrogation with a Comm-B reply, if necessary, to expedite the transfer of these short messages in both directions.

After the short messages and message requests are dealt with, transaction preparation will turn to the ELM sublist, and generate one or more ELM transactions if that sublist is not empty. An ELM transaction always pairs Comm-C interrogations with Comm-D replies. There are four kinds of ELM transactions:

1) Clear Comm-C, which terminates an uplink ELM message transfer.
2) Clear Comm-D, which terminates a downlink ELM message transfer.
3) Uplink ELM, of one or more segments.
4) Downlink ELM, of one or more segments.

The "clear" transactions require a single interrogation and a single reply, while the ELM message-transferring transactions usually require either multiple interrogations or multiple replies. Transactions are prepared for execution in the same order as the entries appear in the ELM sublist. In this list, "clear" transactions (if present) will precede message-transferring transactions. There may be several uplink ELM messages waiting in the sublist for delivery, but transaction preparation will ignore all but the first of these. This procedure allows data link processing to verify completion of the message transfer and insert a new clear Comm-C command ahead of the next uplink ELM. Only one downlink ELM will ever be present at one time in the ELM sublist, since the ground cannot be aware of a new downlink ELM request until the existing one is cleared. These features are discussed in more detail in Section III-12.

The structure of the transaction records can now be understood, in view of the objectives of the transactions themselves. In addition to control bits, which govern the subsequent activity of the roll-call scheduling algorithm, the transaction record contains a large part of the intended interrogation message bits, preformatted to save time for roll-call scheduling. The formulation of interrogation control commands, which include the message bits, was discussed in Section II-3.

For standard transactions, the transaction record may contain the following fields:

1) Priority	(1 bit)
2) Channel time	(16 bits)

3)	Surveillance transaction indicator	(1 bit)
4)	Transaction type	(3 bits)
5)	Comm-A entry indicator	
6)	Comm-B entry indicator	
7)	Format type (F)	(1 bit)
8)	Data-block length (L)	(1 bit)
9)	Interrogator type (IT)	(1 bit)
10)	DABS Lockout (DL)	(2 bits)
11)	ATCRBS Lockout (AL)	(1 bit)
12)	Synchronization indicator (S)	(1 bit)
13)	Altitude/Identity designator (AI)	(1 bit)
14)	Reply length (RL)	(1 bit)
15)	Air-to-ground data link message source (MSRC)	(4 bits)
16)	Synchronous reply time (EPOCH)	(6 bits)
17)	Clear PBUT (CP)	(1 bit)
18)	Clear Comm-B (CB)	(1 bit)
19)	Ground-to-air data link message (MA)	(56 bits)

The first six fields contain control information for channel management use, while the others are intended to become part of the actual transmitted interrogation. Fields 13 through 15, totalling six bits, are multiplexed with field 16, in the sense that EPOCH is present in synchronized interrogations (of either the surveillance or Comm-A type), while AI, RL and MSRC fill the corresponding positions in unsynchronized interrogations. The last field, MA, is present in Comm-A interrogations, and absent in surveillance interrogators. The terminology and mnemonics of all interrogation message fields are consistent with Reference 2. The meaning and setting of these fields is given in the following paragraphs.

Transaction priority is used by roll-call scheduling and target list update in connection with the priority-ordered use of channel time (see Section II-4). Two levels are recognized: "priority," and "normal." Any of the three transactions set up for surveillance purposes, as explained above, is given priority. ELM transactions are always normal. Standard transactions prepared to transfer Comm-A and/or Comm-B messages are given priority if either of these messages (or both) is classified as a "high-priority" message, otherwise the transactions are normal.

Channel time is also used by roll-call scheduling for allocation purposes. It is an estimate of RF channel time required to include the transaction in a schedule. The basis for the estimate is discussed in Section II-6, and transaction preparation assigns this quantity according to a simple rule: one value (CTSS) if the transaction makes use of the short reply, and another value (CTSL) if the reply will be long. The two values are adjustable system parameters.

The surveillance transaction bit is set to "1" (i.e., "yes") for the first transaction of any target transaction block, and to "0" (or "no") for all others. The bit is used for the easy recognition of surveillance transactions in the operation of transaction update.

Transaction type is used to identify one of the eight kinds of transaction, four standard and four ELM. The four ELM transaction types are listed above, in connection with ELM transaction generation. The standard transaction types correspond to the possible pairings of a short (56-bit) or long (112-bit) interrogation with a short (56-bit) or long (112-bit) reply.

The final control bits, the Comm-A and Comm-B entry indicators, are pointers (or indices) which identify the entries in the Comm-A and Comm-B sublists (of the active message list) corresponding to the message and/or message request included in the transaction. They are placed in the transaction record for the use of data link processing, which examines all replies and target transaction blocks after the target has left the beam.

The remaining transaction record fields, which constitute a preformatted portion of the interrogation message, are easily assigned once the function of the transaction has been established. The F bit distinguishes "normal" (surveillance or Comm-A) interrogations from "special" (ELM) interrogations. In channel management terminology, F is set to "0" for standard transactions and to "1" for ELM transactions. The L bit is "0" for short interrogations (surveillance interrogations of 56 bits), and "1" for long (112-bit) interrogations of the Comm-A or Comm-C types. Interrogator type (IT) is set to "1" in all cases; it identifies the source of the interrogation as a DABS sensor.

The lockout fields, DL and AL, are copied from the surveillance file into all transaction records. An exception occurs when DL = "00" and the DABS lockout transition control bit is set to "1." Transaction preparation will then set DL equal to "00" in the surveillance transaction record, and to "01" in any other transaction records.

The S bit is set to "1" to identify synchronized interrogations (S-transaction), while AI="1" similarly identifies the interrogation of an A-transaction, requesting ATCRBS identity in the reply. The RL bit will be "0" if the reply is to be short (56-bit surveillance reply), or "1" if it is to be a long, 112-bit Comm-B reply. The long, Comm-D reply is always elicited by a Comm-C interrogation, which does not contain an RL bit.

The MSRC field is used in connection with Comm-B messages on the downlink, which can only be present in unsynchronized replies. If no message is intended, RL will be "0" (surveillance reply) and the MSRC field is meaningless. If a ground-requested message is intended, RL will be "1" and MSRC will designate the on-board device whose readout constitutes the downlink message. When the pilot requests a device readout (which he does by setting the B bit in some reply), a subsequent interrogation will authorize the Comm-B reply by setting RL equal to "1" and MSRC equal to "0000", which corresponds to "pilot option" regarding message source.

In a synchronized transaction, no Comm-B message can be sent; that is, the reply must be short. Such a reply must also contain aircraft altitude, hence, it is possible to replace the AI, RL and MSRC fields with the six-bit EPOCH field required in any synchronized interrogation. The value of this field is generated by roll-call scheduling (Section II-7), and hence it is ignored by transaction preparation.

The clear PBUT and clear Comm-B bits are used to terminate data transfers on the PBUT and Comm-B channels. The CP bit is always set to "0" by transaction preparation, while the CB setting is determined by the Comm-B state variable copied into the target record from the active message list. CB will be "1" if Comm-B state indicates "clear pending," otherwise it is set to "0". The operation of these bits and the corresponding data channels is given in Sections III-10 and III-11.

The final field, present only in a transaction involving a Comm-A interrogation, is the actual data link message text, MA, to be delivered to the aircraft. This message text is copied from the Comm-A sublist to facilitate interrogation message formatting by channel management, although in principle this could be done directly from the Comm-A sublist, since the Comm-A entry indicator is available. The computer configuration envisioned for the DABS sensor puts most of channel management in a separate processor, hence the approach adopted should be preferred, since the MA data transfer takes place in the transaction preparation phase instead of the more time-critical roll-call scheduling phase.

ELM transaction records are much simpler, since the Comm-C interrogation is dedicated to fewer applications and has fewer fields. The first four fields are identical to those of standard transaction records, although priority is always set to "normal," and the surveillance transaction indicator will always be "0" (i.e., "no"). Transaction type identifies the ELM nature of the transaction and distinguishes between the four possibilities defined above. Channel time is set to CTSL, the value corresponding to a standard transactions with a long reply, for any ELM transaction involving a single interrogation and a single reply, such as a "clear" command. The channel time field is not used for multiple-segment ELM transactions, since these are dealt with in a special way, in terms of channel time allocation, by roll-call scheduling.

The remaining ELM transaction record fields are:

5)	ELM entry indicator	
6)	Length	(4 bits)
7)	Format type (F)	(1 bit)
8)	Data-block length (L)	(1 bit)
9)	Reply type for Comm-C interrogations (RTC)	(2 bits)
10)	Segment number of ground-to-air ELM segment (SNC)	(4 bits)
11)	Ground-to-air ELM segment (MC)	(80 bits)

The ELM entry indicator is a pointer, analogous to the Comm-A and Comm-B entry indicators, which identifies the location of the message or message request in the ELM sublist. Length is the number of segments which characterize an ELM transaction; it is copied from the ELM sublist data. Bits F and L are both set to "1" for any ELM transaction.

The RTC and SNC fields characterize an ELM interrogation, while MC represents a message segment text. These fields are copied from the ELM sublist of the active message list. For an uplink ELM of N segments (i.e., "length" = N), the RTC, SNC and MC fields will each be replicated N times. RTC will be "00" in the first segment (which initializes the transponder, besides delivering a message segment), and "10" in the last segment, which is the only segment to elicit a reply. The intermediate segments all carry RTC equal to "01". The RTC code of an uplink ELM segment may later be changed by transaction update, but the initial values set by transaction preparation are copied from the ELM sublist.

In a downlink ELM, there will be only one RTC, SNC and MC field, and RTC will equal "11", SNC will equal "0000". The MC field specifies the particular segments of the downlink ELM which are authorized for delivery in this transaction. The initial value, copied from the ELM sublist by transaction preparation, will request all segments not yet transferred on previous scans. This value may later be changed by roll-call scheduling, if insufficient channel time is available, and by transaction update as message segments are successfully delivered.

The preparation of an individual target transaction block has been described in some detail. One block is prepared for each of the targets fetched from the surveillance file between the old and new azimuth limits. These blocks are placed in a storage area, called the transaction buffer, where they are accessible to target list update, described in the next Section. It is expected that transaction update will be activated sufficiently in advance by channel control, and given sufficient priority of operation by the scan-to-scan processor executive, that all required blocks will be available by the time they are needed by target list update (see Section I-5 for discussion of the timing of these activities). There is no need, however, for transaction preparation to send a completion notice to channel control, since target list update will take whatever blocks are available when it is enabled, and any blocks which are entered in the transaction buffer late (through some error condition) will be picked up in time for the next DABS period by target list update. At worst, only a portion of the period of visibility of a target would be lost in this case. A flowchart of the transaction preparation algorithm appears in Section III-14.

III-5. Updating the Active Target List

Target list update is the simplest of the three channel management subfunctions described in this report. The responsibilities of this subfunction are:

1) Transfer of the new target transaction blocks from the transaction buffer to the channel management processor.
2) Merging of the new blocks into the active target list, maintaining the range-ordered character of the list.
3) Release of the blocks of completed targets from the active target list, thereby forming a released target list.
4) Updating of certain fields in the active target list header and also in the target records of transaction blocks remaining on the list.

The active target list itself is the central file used by channel management, and its nature and purpose have already been discussed in Sections II-2 and III-3.

Target list update is activated by channel control in advance of each DABS period. Its first task is to transfer any new target transaction blocks which it finds in the transaction buffer to the channel management processor for merging into the active target list. Nothing more can be said at this point regarding the data transfer itself, since further details obviously depend upon the specific configuration of processors and memories used to implement the DABS sensor.

The active target list has already been described as a linked list of target transaction blocks. The three remaining tasks of target list update can be accomplished in a single pass through the list, and it will be assumed that the subfunction is implemented this way, even though the tasks are described separately here. A flowchart will be found in Section III-14.

The list is approximately ordered, using range delay as a key parameter, according to the following rule (see also Reference 1). Let D_1 and D_2 be the range delays of two targets, T_1 and T_2. If

$$D_1 < D_2 - d ,$$

where d is an adjustable internal system parameter, then T_1 must follow T_2 on the active target list. If

$$D_2 < D_1 - d ,$$

then T_2 must follow T_1 . If

$$\left| D_1 - D_2 \right| \leq d ,$$

the order is immaterial. If the "mismatch parameter," d, were set to zero, conventional ordering by decreasing value of range delay would result. The purpose of allowing the mismatch is to speed up the process of updating and reordering the active target list and the most effective way to achieve this end is to mask an appropriate number of low-order range delay bits and perform rigorous ordering on the result, using the "bubble sort" technique which stops as soon as the desired order is established.

It should be clear that target transaction blocks can be deleted from the list (as when targets are released) without destroying the approximate ordering of the remaining list. New target transaction blocks can easily be merged into the list in a single pass if the set of new blocks is first ordered according to the same rules, and it is also assumed that this will be done. At a later stage of system design it may become obvious that the computation time saved by using an approximate ordering is not worth the cost in channel time (see Section II-6), and the value of d may be set to zero.

When activated, target list update is provided with a cutoff azimuth (see Section I-5). Any target whose last likely azimuth is smaller than the cutoff azimuth is assumed to have fallen behind the beam, and will be removed from the active target list. Last likely azimuth is initially computed by surveillance processing and copied into the target record by transaction preparation. As soon as the target's azimuth is measured, the new value is entered into the last likely azimuth field by transaction update, so that targets will be removed at the appropriate time.

Targets are also removed from the list if there is no further business pending, and this case is recognized by the setting of the target completion bit to "1" in the target record. This is done by transaction update, and the targets are removed by target list update.

Target transaction blocks removed from the active target list for either of the reasons just described are linked together (in any order) to form the "released target list" for the current DABS period. The target transaction blocks of released targets contain information useful to the subsequent operation of surveillance and data link processing, including pointers to the locations of actual replies. Again, the details of storage allocation and the transfer of both replies and the blocks of the released target list are implementation-dependent, hence, no further discussion will be given here.

The remaining task of target list update involves the target records and the active target list header. In certain cases, the target completion bit of a target record will be set to "1" (i.e., complete), while either range completion or azimuth completion will indicate "0" (incomplete). In this case, target list update must reset the target completion bit to "0", and the target must not be removed from the active target list. This action is part of the regulation of reinterrogations to targets for which range or azimuth has not yet been obtained, and it is discussed below.

The active target list header fields which are updated by target list update are the channel time estimate sums, the first target range delay, and the uplink ELM indicators. Channel time sums are the totals of the channel time estimates of priority, synchronized and normal transactions. Their use in allocation is described in Section II-4. As targets are added and deleted from the list, the values of the appropriate sums are increased or decreased by the amounts contained in the corresponding target record channel time fields. The range delay of the first target on the list is also used during the allocation computation and this is easily updated, if necessary, as the active target list is processed.

In the preparation of the schedule precursor, by roll-call scheduling (Section II-5), use is made of pointers to uplink ELM transaction records, as well as segment counts and channel time estimates, and these must be updated by target list update. The pointers may be linked together in the order in which uplink ELM transactions are added to the list, which will result in a first-come, first-served type of service for uplink ELM's.

An alternative is to keep the pointers in the same order as the transactions to which they correspond. For each uplink ELM, a segment count (equal to the "length" field of the transaction record) is provided, with the intention of speeding the allocation task. The channel time estimate is the amount of channel time that would be required to deliver the entire uplink ELM. If there are N segments, this channel time is equal to

$$(N - 1)\ 800 + CTSL\ ,$$

in range units, where CTSL is the usual channel time estimate for a transaction involving a long reply. The basis of this estimation is given in Section II-5 in connection with precursor preparation.

III-6. Reply Transfer and Reply Appraisal

The transaction update subfunction of channel management consists of a number of small tasks, which are individually described in Sections III-6 through III-13. The intention of this approach is to facilitate the tracing of individual threads and protocols through the system. To this end, certain actions of transaction preparation and target list update are described or repeated here whenever an individual logical thread involves the coordinated action of these three subfunctions.

The first step for transaction update is to "acquire" the replies and examine them in a general way, before the more detailed business of transaction update can begin. Transaction update is enabled by channel control, as soon as roll-call scheduling is finished. Channel control also enables the transfer of reply data blocks, which are the outputs of the DABS reply processor, from the hardware to the channel management processor, by issuing what amounts to a "read" command. This command enables a block data transfer of reply data blocks, as the replies arrive and are processed by the DABS reply processor. The reply processor is given a starting location in a storage area accessible to channel management and it initiates the transfer of each reply data block as the block is formulated.

When transaction update is enabled, some reply data blocks may already have been transferred to the reply storage area. If not, transaction update will await their arrival. Transaction update begins by threading through the active target list, passing over all but qualified, i.e., actually scheduled, transactions (see Section II-4 for a discussion of qualification rules and qualification indicators). For each transaction that was scheduled, a pointer to the location in storage of the corresponding reply data block is appended to the transaction record. Even if no reply is received, a reply data block will be prepared and transferred by the DABS reply processor.

Reply data blocks must be accessed by the functions implemented in the scan-to-scan processor, as well as by channel management. If reply data block images must be stored in two storage areas for this purpose, then only a pointer to the location used by the scan-to-scan functions needs to be added, since transaction update examines replies only once, and it determines their location from the starting address assigned by channel control. This starting address must accompany the enabling command to transaction update.

The treatment of reply data block transfer described here differs in detail from that implied by a literal interpretation of the specifications of

Reference 1. The logical operation of transaction update, however, is unchanged.

The reply data block format is given in Reference 1. The information fields are as follows:

1) Control	(1 byte)
2) Monopulse estimate	(1 byte)
3) Antenna boresight azimuth	(2 bytes)
4) Range Correction	(2 bytes)
5) DABS reply bits (except address)	(4 bytes for short reply, 11 bytes for long reply)
6) Time of Day	(3 bytes)

The control byte contains a three-bit "type" field, which will always indicate either a short or long DABS reply, depending on which reply length was requested by the interrogation. It also contains a two-bit "failure field" which distinguishes among the following four possibilities:

1) Valid reply:- a reply with no uncorrectable decoding errors and containing a monopulse measurement,
2) No azimuth:- a reply with no uncorrectable decoding errors, but lacking a monopulse estimate,
3) Invalid:- a reply with an uncorrectable decoding error,
4) No reply:- no reply preamble was detected.

The other fields are more fully defined in later Sections, except for time of day, a coarsely quantized measurement time which is not used by channel management.

"Valid" replies and "no azimuth" replies are subjected to a further screening step by transaction update, called conformance testing. It will be appreciated that the nature of any reply, i.e., whether it is a surveillance, Comm-B or Comm-D reply, whether it is synchronized or unsynchronized, contains altitude or ATCRBS identity, etc., is completely controlled by the interrogation itself. Conformance testing is simply a check to be sure that the reply has all the expected characteristics. If the reply fails this test, the interrogation will be repeated, and the performance monitoring function will be informed. Presumably, such a failure will be due to a fault in the transponder.

The DABS reply processor is told which reply length to expect, and will not accept a reply of the wrong length because it will certainly find an uncorrectable decoding error. The reply processor, however, will accept any reply of the proper length which decodes properly. Transaction update makes whichever of the following tests is appropriate, based on information contained in the transaction record:

1) The S-bit must be "0" in an unsynchronized surveillance reply, and "1" in a synchronized surveillance reply.
2) If ATCRBS identity was requested, the AI bit must be "1", otherwise "0".
3) Surveillance replies must have F = "0" and L = "0".
4) Comm-B replies must have F = "0", L = "1", and S = "0".
5) Comm-D replies must have F = "1" and L = "1".
6) A Comm-D reply to an uplink ELM transaction (not a clear command) must have K = "1".

7) A Comm-D reply in a downlink ELM transaction (not a clear command) must have K = "0".

Replies which pass the conformance test are called "accepted replies," and only these are subjected to further processing by transaction update, as detailed in the Sections below.

"Not accepted" replies consist of replies which failed the conformance test and invalid replies. Transaction update takes no further action for these replies (with one exception) replacing the reply storage pointer with an indication of non-acceptance and leaving the current transaction indicator untouched, so that the transaction will remain to be repeated at the next opportunity. The exception is the case of a synchronized reply, in which the receipt of a reply, even if it is not accepted, is presumed to verify the successful reception of the synchronized interrogation. No information is extracted from such a reply, but the current transaction indicator is moved on to the next transaction record in the target transaction block. If no further transactions remain, the procedure is the same as that which follows the successful receipt of a reply to the last transaction of any transaction block, as explained in the Sections to follow.

If no reply is reported by the DABS reply processor, the reply is assumed to be missing, although it could have arrived outside the preamble detection window included in the original reply control command. The reply storage pointer should make a suitable indication for a missing reply. The screening and classification of replies, referred to as reply appraisal in connection with transaction update, are illustrated schematically in Figure 22.

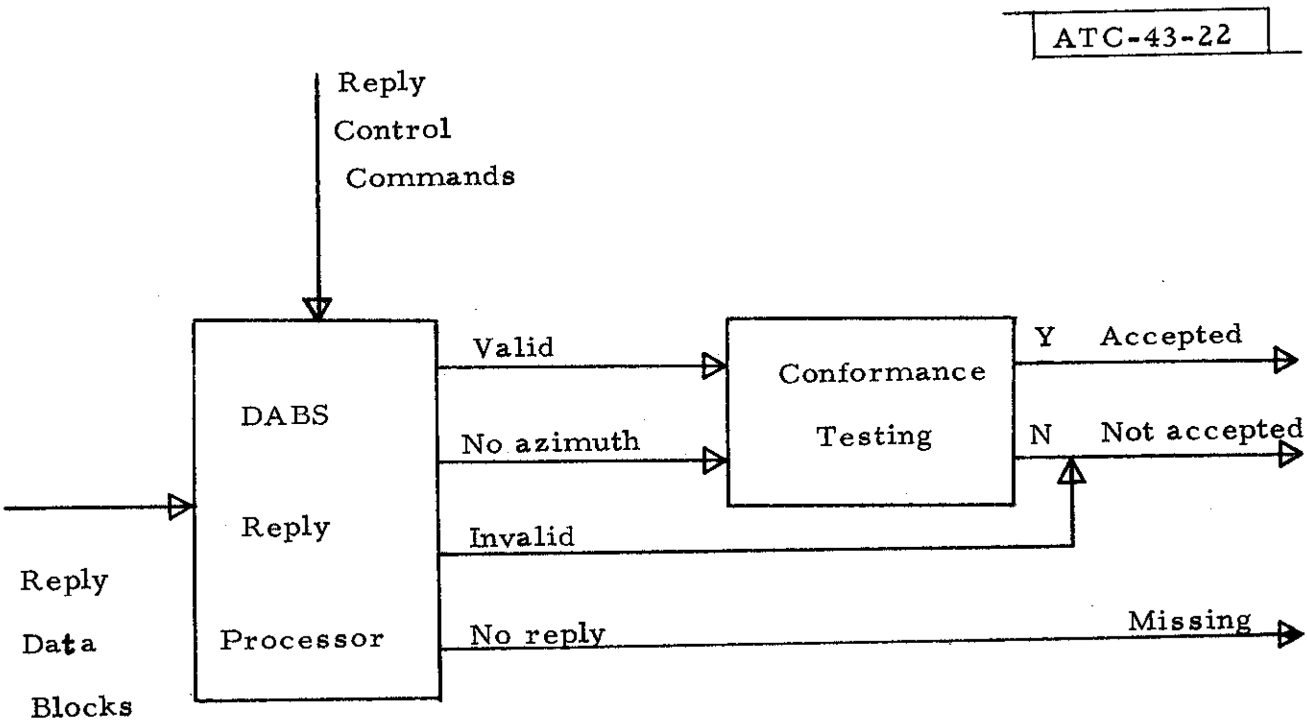


Figure 22.

Reply Appraisal

If a reply is declared missing, and the corresponding transaction was not the surveillance transaction for the target in question, then no further action is taken by transaction update. The effect will be the same as with a non-accepted unsynchronized reply, and the transaction will remain to be repeated. Missing synchronized replies are also repeated. If the missing reply was part of a surveillance transaction, then a special protocol is followed, as discussed in the remainder of this Section.

If a reply is missing in a surveillance transaction, which will contain the first interrogation of the current scan for that target, it could mean either a link failure or a premature interrogation, due to a large estimate of azimuth prediction uncertainty. Because of the first possibility, the sensor will increase the transmitter power on subsequent interrogation attempts, and because of the second possibility, the number of attempts in any one DABS period will be limited.

Two power levels are used in the DABS ground transmitter: 100 watts and 800 watts. As mentioned before, the use of high power is inhibited altogether for close-in targets. All targets are interrogated at low power first, but are reinterrogated at high power after the first failure, unless they are designated as close-in targets. Non-responding targets are given a fixed number of attempts in a single DABS period, the total value being an adjustable internal system parameter. The "remaining tries" field in the target record is initialized to the value "NTRY" (nominally equal to two) by transaction update. After a failure, this field is read by transaction update and a repeat transaction is allowed if "remaining tries" is not zero. "Remaining tries" is then reduced by one, so that an initial value of two corresponds to a total of three tries per DABS period. In addition, a "high power flag" is set to "on" for this target, so that the following interrogations can be made at the high transmitter power. A special bit for this purpose is not identified in the target record, since it is not set by transaction preparation, but the flag generated by transaction update must be available to roll-call scheduling, and the target record (or an extension of it) would be the logical place for it. As pointed out in Section II-3, the high power flag is a request not a command for high power, and it will be carried out by roll-call scheduling only if sufficient time has elapsed since the last use of high power, to prevent overloading the high power transmitter.

The process is repeated in subsequent DABS periods, beginning again with low power. The reason for the reversion to low power is the presumption that premature interrogation was the cause of the failure. This mechanism is implemented in the following way. When transaction update encounters a missing reply and the "remaining tries" field indicates zero, then the target completion bit is set to "1" (i.e., "complete") to inhibit the inclusion of that target in any further schedules of the current DABS period. The range completion indicator remains set to "0" (i.e., "incomplete"). Transaction update will also reset the high power flag to "off" and reset "remaining tries" to NTRY, so that everything is ready for the next DABS period.

The special combination: target completion = "1", range completion = "0", is recognized by target list update when it processes the active target list at the end of the DABS period. Target list update then completes the procedure by resetting the target completion bit to "0".

The action of transaction update has now been fully described in the cases of missing or non-accepted replies. The further processing of accepted replies will be found in the remaining Sections.

III-7. Range Completion and the Altitude Echo

The expression "range completion" refers to the successful measurement of target range, and, so far as channel management is concerned, it is synonomous with the success of the surveillance transaction. Whenever a reply preamble is detected, a range measurement is possible, but the surveillance transaction will not be accepted by transaction update unless the reply also decodes properly. The range measurement made from the surveillance transaction (or any subsequent accepted reply) may therefore be assigned with very high confidence to the target which was interrogated. Since the surveillance transaction always requests altitude in the reply, the first successful determinations of range and altitude, in the current scan, are obtained from the same reply.

Channel management is satisfied that a range measurement for the target has been made when the surveillance transaction is completed, and this transaction will be repeated until an accepted reply is received or else the target drops behind the beam. Surveillance processing, however, will examine all the replies, and exercise selection rules of its own to obtain a position update.

As emphasized elsewhere, roll-call scheduling is always based upon range delay, which equals the two-way propagation delay, including transponder reply delay, corresponding to the minimum likely value of target range.

Preamble detection is enabled in the DABS reply processor at a time corresponding to the expected time of arrival of the reply, based on this propagation delay. The time interval from the opening of the preamble detection window to the actual arrival of the leading edge of the first reply pulse is called the range correction for this reply. It is the value of range correction which is included in each reply data block for a DABS target (ATCRBS reply data blocks report range in a different way). By adding the range correction to the original range delay, the actual propagation delay is found and hence true slant range can be measured, as discussed in Section II-6.

Once it is satisfied that the surveillance transaction was successful, channel management has no need to determine range itself. However, if the target is flagged for synchronous service, then a synchronized transaction will have been prepared, and the timing of the corresponding synchronized interrogation will make use of the range correction accompanying the surveillance transaction reply. The way in which range correction is used in these interrogations is fully explained in Section II-7. In order to make this quantity available to roll-call scheduling, when it is needed later during the scheduling process, transaction update will copy the range correction from the surveillance transaction reply data block into the target record of the target transaction block. The range correction field was introduced into the target record, even though it is not given a value by transaction preparation.

The synchronized interrogation is also used as a vehicle for the delivery of the altitude echo (ALEC), since the sets of aircraft designated for sychronous service and for ALEC service coincide. ALEC is a 12-bit field, which is preceded by the field "0000", to occupy the 16-bit special data (SD) field of the synchronized interrogation. In any other interrogation, the first four bits of the SD field are set to "1111" by roll-call scheduling, to inhibit the display of the following 12 bits in the airborne ALEC display.

The altitude echo, as displayed to the pilot, allows him to verify that the aircraft's reported altitude agrees with that indicated by his direct-reading altimeter (see Reference 2 for further details). Since the altitude reported in any DABS reply is uncorrected for local barometric pressure, this correction must be made (for aircraft below 18,000 feet, MSL) to the reported altitude before ALEC is formulated. The altitude displayed in the cockpit will be corrected for barometric pressure (when appropriate) by means of an altimeter setting, provided to the pilot by ATC. Equivalent information is to be sent from ATC facilities to DABS sensors to enable the required corrections for ALEC. The pressure data will presumably be regionalized in some way, and converted into a set of values, corresponding to map grid points, by network management within the DABS sensor.

Network management checks the location of each DABS track, once per scan, against the aforementioned grid, and copies the appropriate value into the surveillance file of the target. The correction, called ALTCOR, is copied again into the 8-bit altitude pressure correction field in the target record, when the latter is formulated by transaction preparation. The final step in this chain occurs when transaction update combines the reported altitude, obtained from the surveillance transaction reply data block, with the altitude pressure correction from the target record to form the desired value of ALEC. A specific field for ALEC has not been specified in the target record (in Reference 1), but an extension of that record should be used by transaction update to make ALEC available to roll-call scheduling, in analogy to the provision of high power flags, when they are required.

It remains to discuss the coding of the quantities involved in the ALEC computation. Reported altitude represents aircraft pressure altitude, digitally encoded in the same format used for a standard ATCRBS Mode C reply. When decoded, the result expresses altitude in 100-ft. increments. The altitude pressure correction provided by network management is a quantity to be added algebraically to reported altitude, in order to make an approximate correction for the effect of local barometric fluctuations. ALTCOR is expressed in a special 8-bit BCD field, which uses one bit for sign, three bits to express thousands and four bits to express hundreds of feet of correction. ALEC is the sum of these quantities, expressed in another modified - BCD code of 12 bits, as follows:

1) Bits 0-3 represent the decimal integers 0-12, corresponding to 10,000-ft. increments of altitude, through 120,000 ft.
2) Bits 4-7 represent the decimal integers 0-9, corresponding to 1000-ft. increments of altitude.
3) Bits 8-11 represent the decimal integers 0-9, corresponding to 100-ft. increments of altitude.

The required computation is probably best accomplished in BCD arithmetic, with reported altitude being decoded directly into BCD, and proper account being taken of the sign representation in ALTCOR.

III-8. Azimuth Completion

Unlike range, an azimuth measurement is not automatically possible with every accepted reply. An off-boresight azimuth signal is measured, using amplitude-comparison monopulse, on each bit of a DABS reply by the reply processor, but only those measurements made on high-confidence bits are used to form the final value (see Reference 1 for details). If an insufficient number of high confidence bits is found, no monopulse estimate will be found in the reply data block. This is the "no azimuth" case mentioned in Section III-6. The azimuth completion bit in the target record is initially set to "0" and is unchanged until a valid reply (i.e., a reply with a monopulse estimate) is received. Each valid reply is examined and subjected to certain tests, described below, which either leave the completion bit unchanged or set it to "1". Once the completion bit is set to "1", no further azimuth processing takes place by channel management. As with range and altitude, surveillance processing examines all replies in order to extract a position update, using criteria different from those of channel management.

In order to explain the selection rules used by transaction update, with regard to azimuth, it is necessary to discuss the mechanism of azimuth measurement briefly. Further details may be found in Reference 1. The DABS antenna subsystem forms two beams (sum and difference beams) from which a signal is derived which depends in a unique way on the azimuth of the target relative to the antenna boresight direction at the time. This "monopulse signal" is eventually digitized to produce an 8-bit number, contained in the DABS reply data block in the "monopulse estimate" field. At the time of reply detection, the reply processor also samples the reading of the antenna shaft encoder, entering the value in the 14-bit field, "antenna boresight azimuth." The latter field represents the boresight pointing direction in "azimuth units" (1 azimuth unit is approximately equal to 0.022°).

In order to convert the 8-bit monopulse estimate into a target azimuth angle relative to boresight, a calibration table is maintained, called the off-boresight azimuth lookup table, or "lookup table" for brevity. The table is entered with monopulse estimate and returns a signed 16-bit word representing off-boresight azimuth in azimuth units. The table is maintained, and occasionally changed, by performance monitoring, on the basis of measurements on special transponders installed for the purpose. Normally, a monopulse estimate value of about 128 (the mid-point value) will correspond to a zero value of off-boresight azimuth. However, small variations are expected to occur which can be corrected by adding a quantity called the "table offset value" to the monopulse estimate, before entering the lookup table. The table offset value is also maintained by performance monitoring. As in Reference 1, the sum of the monopulse estimate, contained in a valid reply, and the current value of the table offset value will be denoted by SIG.

When processing the azimuth information of a valid reply from a target whose azimuth completion has not yet been established, transaction update compares the value of SIG with two parameters, MBL and MBH, which represent the lower and upper limits of the useable portion of beam (for monopulse purposes) in the same units as SIG. These parameters are adjustable internal system parameters. High values of SIG correspond to positions to the right (i.e., clockwise) of boresight, hence the corresponding targets are still in the leading half of the beam. Targets will respond with successively lower values of SIG as the beam scans past and eventually beyond them.

If the value of SIG, for the given reply, lies between the values of the parameters, i.e.,

$$MBL \leq SIG \leq MBH \ ,$$

then the monopulse measurement will be accepted by channel management and an off-boresight azimuth will be determined by reference to the lookup table. Moreover, the azimuth completion bit will be set to "1", and the off-boresight azimuth added (algebraically) to the antenna boresight azimuth (right-justified in a 16-bit word) to produce the actual target azimuth. Finally, this azimuth is placed in the aircraft's target record, overwriting the previous contents of the last likely azimuth field. It will be recalled that last likely azimuth is used by target list update to sense the passage of a target behind the antenna beam.

If SIG is smaller than MBL, i.e.,

$$SIG < MBL \ ,$$

then the target is out of the useable portion of the beam and falling further behind. In this case there is no hope of obtaining an azimuth measurement in the current scan, as none has yet been made, hence azimuth completion is set to "1" to inhibit further attempts. Also, to insure that the target will be removed from the active target list upon the next update of that list, a fictitious target azimuth is computed and entered in the last likely azimuth field. The fictitious value is obtained by setting off-boresight azimuth equal to a negative value representing at least the trailing-edge half beamwidth point. The value used is a system parameter which must be set in conjunction with the rule for computing cutoff azimuth (Section I-5) to force the removal of the target by target list update.

The remaining case,

$$SIG > MBH \quad ,$$

represents an early acquisition of the target, with respect to antenna boresight. To obtain a better measurement, further interrogation of this target is deferred until the next DABS period, since it is expected that the antenna will scan through only a fraction (e.g., one-fourth) of its beamwidth from the start of one DABS period to the start of the next. To implement the delay, azimuth completion is unchanged but target completion is set to "1". This inhibits further scheduling until target list update, recognizing the combination: azimuth completion = "0", target completion = "1", resets target completion to "0". The situation is analogous to the management of range completion for targets not responding to the surveillance transaction interrogation.

The only case not yet mentioned is the one in which all planned and dynamically prepared transactions are executed and an acceptable azimuth has not yet been obtained. The advance of the current transaction indicator will reveal that no more transaction records exist, but the azimuth completion bit is still "0". In an attempt to measure azimuth, a new transaction record is prepared, in a manner analogous to the dynamic preparation of transaction records described in Sections II-9 and II-11 below. The current transaction indicator will be set to the new transaction and, if an acceptable azimuth is not then obtained, the process will be repeated until the target is removed from the active target list, on the basis of its original last likely azimuth.

The new transaction record is the simplest possible kind. It is a standard transaction, using an unsynchronized surveillance interrogation paired with a surveillance reply (unsynchronized, of course). Thus, the interrogation bits F, L, S and RL are all set to "0", and IT is set to "1". Further, altitude is requested (AI="0") and MSRC, CP and CB are arbitrarily set to "0". The lockout fields, DL and AL, are copied from the record of the last transaction executed.

III-9. ATCRBS Identity Request Protocol

The ATCRBS identity of a DABS-equipped aircraft will be assigned by ATC in the same way that it is assigned for an aircraft equipped only with an ATCRBS transponder. When not locked out to interrogations by ATCRBS interrogators, the DABS transponder will reply to Mode A ATCRBS interrogations with this assigned identity code. Discretely-addressed DABS interrogations will normally elicit replies containing altitude, coded in a 12-bit field exactly as for an ATCRBS Mode C reply. Provision has been made, however, to allow altitude to be replaced by the 12-bit ATCRBS identity code in a DABS reply, under ground interrogator control. The AI bit of an unsynchronized surveillance or Comm-A interrogation controls the contents of altitude/identity field in the corresponding reply, and the significance of this field is identified in the reply itself by a bit, also denoted AI. In either context, AI= "0" signifies altitude, while "1" denotes ATCRBS identity.

Under ordinary circumstances, ATC can ask a DABS sensor to request and report the ATCRBS identity of a particular target, presumably to confirm or establish a correlation between that identity and other target identifiers, such as DABS address. Such a request from ATC will result in the setting of an ATCRBS identity request bit (also called AI) in the target's surveillance file record, from which it is copied into the target record when one is next generated by transaction preparation. As explained in Section III-4, transaction preparation will then proceed to formulate an "A-transaction," specifying an unsynchronized interrogation with AI set to "1". This transaction record will follow the surveillance transaction record, so that the requested ATCRBS identity will be downlinked only after target altitude has been obtained. The A-transaction, when present, is designated priority. Any further standard unsynchronized transactions prepared for the target transaction block will specify AI = "0" in the interrogations.

After transaction preparation inserts an A-transaction in the target transaction block, it then resets to "0" the ATCRBS identity request bit in the target record. This step is necessary, since the AI bit in the target record is used by transaction update to regulate the response to a pilot request for permission to downlink ATCRBS identity, as explained in the following discussion.

Aside from voice radio, the main channel available to the pilot for reporting emergencies is the ATCRBS identity report. The standard codes "77XX" (emergency) and "76XX" (radio failure) may be supplemented by other code assignments to provide a dedicated channel with a small vocabulary for emergency purposes.

To initiate the transfer of an emergency code of this kind, the pilot must take either of two actions which will result in the setting of the A bit to "1" (instead of its normal value of "0") in subsequent surveillance or Comm-B replies. The actions are:

1) Setting either "77XX" or "76XX" into the ATCRBS code wheels will set the A bit, which will remain set until the ATCRBS code is changed.
2) Pushing the "alert button" on the DABS transponder control panel will directly set the A bit (regardless of the current ATCRBS code). In this case, the A bit will be reset to "0" as soon as the transponder decodes and accepts an interrogation requesting ATCRBS identity.

In any case, an alert light will be displayed to the pilot as long as the transponder is conditioned to put A = "1" in appropriate replies.

Transaction update checks every non-ELM reply for a pilot request to downlink ATCRBS identity. On finding A = "1" in such a reply, a further test is made to see if the AI bit in the target record is set to "0". If the target record bit is "0", and only then, transaction update will arrange for an interrogation to the target requesting ATCRBS identity. This process is described as the "dynamic preparation" of an A-transaction, and whenever it occurs, the target record AI bit is changed to "1". The setting of AI to "1" in the target record then inhibits further response to a pilot-initiated ATCRBS identity request in the current scan. Delivery of the interrogation of a dynamically prepared A-transaction will itself turn off the pilot's request, if made by means of the alert button, as in case 2) above. In case 1), if the ATCRBS code wheels remain set as described, replies will continue to have A = "1", perhaps for a duration of many scans. In each of these scans the A bit will be sensed in the surveillance transaction reply and an A-transaction will be prepared which will be executed next, and presumably transfer ATCRBS identity in the reply. For the rest of the scan (i.e., while the target remains in the beam), the A bit will be ignored.

To prepare an A-transaction, transaction update checks the next transaction record after the current transaction in the target transaction block. If there is another transaction, and if it contains an unsynchronized surveillance or Comm-A interrogation, the AI bit is simply set to "1" in this (and only this) transaction record (it could already be "1", if ATC had made a similar request). If a transaction cannot be modified in this way, a new transaction record will be prepared and inserted before any others already present in the transaction block. The new transaction will be of the unsynchronized, priority, standard type, pairing a surveillance interrogation with a surveillance reply. The interrogation bit settings are exactly the same as those listed at the end of the preceding Section (for extra interrogations attempting to measure azimuth), except that AI will be set to "1".

If the attempt to obtain ATCRBS identity is unsuccessful, regardless of the source of the original request, the sensor will try again on the next scan. To assure this, data link processing tests for this case, and sets AI = "1" in the surveillance file record of the target if an unfilled request has occurred. It is because of this possibility that channel management always gives A-transactions priority status.

III-10. The Pilot Button Channel Protocol

Whenever an acceptable reply is received to an interrogation which contained a Comm-A message, it is assumed that the message was received by the transponder and delivered to the appropriate on-board device for display (the device address is internal to the MA message field). The reply itself constitutes a "technical acknowledgment" that the link and its terminal equipment has functioned properly. In some cases it is important for the ground to know that the pilot has actually been made aware of the message and to obtain from him a definite response in acknowledgment. A particular bit within the MA field, called AR, is dedicated to this function, and any Comm-A message which arrives at the transponder with the internal AR bit set (to "1") is understood to demand a pilot-generated acknowledgment. Only Comm-A messages have this property (see Reference 2 for details on the internal structure of the MA message field).

The DABS transponder itself provides the pilot with a "pilot button" for the acknowledgment of messages arriving with AR = "1". This "button" is capable of generating any of three messages, "yes" (will comply), "no" (cannot comply), and "test", a special-purpose input not used in response to an acknowledgment request. The state of the pilot button controls the two-bit PBUT field, present in all surveillance and Comm-B replies. The coding for PBUT is as follows:

1) "00"	Inactive
2) "01"	Cannot comply
3) "10"	Will comply
4) "11"	Request test transmission

The normal state of the pilot button is inactive, with "00" in the PBUT field of all appropriate replies. As soon as the transponder decodes a Comm-A interrogation which requests pilot acknowledgment, the PBUT field is cleared of any previous setting, so that PBUT will read "00" in the immediate reply, and after a one-second delay, a light on the pilot button will convey the request to the pilot. After reacting to the uplink message, the pilot will push either "yes" or "no", which will cause the corresponding code to be entered into the PBUT field of subsequent surveillance and Comm-B replies. The pilot's acknowledgment is thus downlinked as soon as the transponder receives an appropriate interrogation.

Messages demanding acknowledgment can originate with the local IPC processor, or with any ATC facility served by the sensor. It will be observed that the pilot button channel for acknowledgment contains no identifier of the message being acknowledged, and that the pilot's response will be found in replies by sensors other than the sensor that uplinked the original message. For this reason each sensor broadcasts the PBUT field value, if "01" or "10", to all ATC and IPC facilities with which it communicates. The special PBUT message includes, of course, the DABS identity of the sending aircraft.

Because of the possibility of downlink failure, the PBUT field cannot be reset to "00" by the transponder after a reply is transmitted containing a pilot acknowledgment. Instead, the ground must clear this field deliberately, and this is the function of the "clear PBUT" bit, CP, present in surveillance and Comm-A interrogations. This clearing action will be undertaken by any sensor, on its own initiative, after receipt of a "non-zero" PBUT value. If the pilot pushes "test" for reasons of his own (and not discussed here), any sensor noting this value will generate a special "test request" report, send it to the local IPC facility, and attempt to clear the PBUT field in the transponder.

In order to "clear PBUT", which becomes the responsibility of transaction update, it is only necessary to set CP = "1" in a suitable interrogation of a pending transaction. Any standard transaction will suffice, and if one or more of these await execution in the current scan, the transaction record of the first of these will be modified so that the CP bit will be "1" in the transmitted interrogation. Only one transaction is modified, and if no standard transaction is pending, no further action is taken by transaction update.

The rules of operation of transaction update are therefore very simple. The resulting character of the pilot acknowledgment channel deserves some further comment. At present, only messages addressed to the IPC display are expected to make use of the AR bit and the PBUT response channel. Other messages may request and receive acknowledgment in other ways, using Comm-B replies. At any one time, only one agency of control, either IPC or ATC, should be in a position to issue commands to an aircraft which require acknowledgment through the PBUT channel. It devolves upon this agency not to send a command before the receipt of acknowledgment to a previous command, unless it is prepared to deal with possible ambiguity regarding the non-specific pilot acknowledgments. The possibility of ambiguity is minimized by several features of the design, including the following:

1) Receipt of a message requesting acknowledgment clears out any old PBUT message, thus ending the generation of "old" responses, even if they have not been successfully downlinked.

2) A new message will not light up the pilot button immediately, but a built-in delay of about one second must elapse before the button is armed, i.e., responsive to new input.

3) One or more sensors will be interrogating each aircraft, and at least one sensor will provide multiple interrogations each scan. Thus PBUT, once set by the pilot, should find its way to the ground within a scan time.

4) Sensors are required to send output messages, such as the pilot button message, within a fraction of a scan of their decoding from a reply, hence the overall channel from pilot to original source should respond promptly.

5) The automatic clearing of PBUT by the sensor will usually result in almost immediate resetting by the sensor which reads the "non-zero" PBUT in a surveillance transaction reply, and clears the field in the ensuing synchronized transaction.

6) If a sensor reports PBUT but fails to deliver the clear PBUT message in the same scan, the sensor does not remember its failure and place CP = "1" in the first interrogation of the next scan. With several sensors operating independently, and with a string of commands being issued relatively rapidly, such behavior could result in the clearing of an unread PBUT response. If PBUT is not cleared by another sensor, the non-zero value will be sensed next scan and another attempt made to clear it.

III-11. The Comm-B Channel

The DABS transponder provides a data link between the ground and a number of on-board devices, through the on board "wire link". At least three classes of on-board devices are expected to be employed: displays for the pilot, pilot-message input devices, and on-board sensors. Displays, such as the IPC display, are controlled from the ground by means of Comm-A messages. The display device address is contained within the Comm-A message field (MA) itself, and a sequence of Comm-A interrogations may be employed to convey a longer (intermediate-length) message. The direct dual of the single-segment Comm-A uplink message is the ground initiated Comm-B message request, which is used to read out an on-board sensor. The pilot need not be aware of this activity, and the desired device is addressed by means of the MSRC code included in the interrogation itself. This type of Comm-B message is, like the single-segment Comm-A message, complete in a single transaction. In either case, the reply constitutes technical acknowledgment of the message or message request contained in the interrogation and, in the latter case, it also includes the desired message.

In the third category, the Comm-B format is used to transfer the contents of a pilot-activated message input device. This use of the Comm-B reply is referred to here as the "Comm-B channel," and it employs a protocol distinct from the simple procedure which suffices for the other devices. The reason a more elaborate protocol is needed is that the pilot initiates the message, while the ground retains full control of the link. There is no way the pilot can independently and directly address the ground. Thus, the pilot must first make the ground aware of his desire to send a message, since he can neither send a reply without an interrogation, nor include his message in a reply on his own initiative. A single bit, the B bit, is reserved in the surveillance and Comm-B reply formats for this purpose. The pilot originates a message on his input device and pushes an appropriate "send" button. The transponder reacts by setting B = "1" in subsequent replies of the types described. The sensor must look for the setting of the B bit, recognize a new downlink message request, and arrange to transfer the message. This task is the responsibility of transaction update, whose operation is discussed below. After the message is received, a third transaction is used to send a "clear Comm-B" command, informing the pilot and his input device that the exchange is complete. This last step, which constitutes technical acknowledgment of the message transfer, is accomplished by setting a special bit, CB, to "1" in a later interrogation of the surveillance or Comm-A types.

Unlike the transfer of PBUT, only one sensor may exercise the protocol just outlined for any given aircraft at a particular time. That sensor will be the one which, at the time in question, finds the primary bit set to "1" in the surveillance file record of the corresponding target. The primary bit is updated for each target on every scan by network management, and elaborate precautions are taken to avoid the possibility of a duplicate assignment of primary sensors. The primary bit is copied into the target record by transaction preparation, hence transaction update is always aware of the current value of this indicator for any target it is processing.

If the primary bit is "0" for some target, then the B bit in that target's replies will be ignored in the current scan, and none of the Comm-B protocol will be executed. If the primary bit is set for a given target, then transaction update will check the B bit of any reply from that target and also maintain the currency of a state variable called Comm-B state. Comm-B state is represented by a two-bit field in the target record, which stands for one of the three states:

1) Inactive
2) Comm-B pending
3) Clear pending

The Comm-B state variable is used by transaction update to keep track of its progress through the three transactions involved in such a Comm-B exchange, and to prevent unnecessary repetition of its actions. When the target has left the beam, the value of Comm-B state is copied into the target's active message list header by data link processing. On the next scan this field is re-copied into the target record by transaction preparation, providing continuity and scan-to-scan memory for Comm-B channel activities. It is possible and desirable to begin a scan with a clear Comm-B command, if the channel was left uncleared last scan, since no other sensor is independently trying to manage the message transfer. This feature contrasts the PBUT protocol described in Section II-10.

The normal Comm-B state is inactive. If, in this state, transaction update detects B = "1" in a given reply, and the target's primary bit is set to allow Comm-B activity, then a transaction record will be modified or dynamically prepared to attempt to comply with the pilot's request in the same scan. First, the Comm-B state is changed to "Comm-B pending," which will inhibit any new reaction to B bit settings in future replies until this message is received. Next, transaction update will search the target transaction block for a pending standard, unsynchronized transaction with a surveillance reply. If one of these is found, it is changed to specify a Comm-B reply. The interrogation message bit RL is changed to "1", MSRC is set to "0000" (source device is pilot's option), and the other transaction record fields are modified to reflect these changes. Thus, transaction type will be adjusted, channel time changed to the value used for long replies, and the Comm-B entry indicator set to a special value, referred to as "PILOT", to identify this Comm-B transaction (to data link processing) as a dynamic response to a pilot request.

If no suitable transaction record is found, transaction update will generate one for the purpose, inserting it into the target transaction block between the last pending standard transaction and first pending ELM transaction, if any. The new transaction will be of the unsynchronized standard type, with corresponding values of transaction record fields, including normal priority. The interrogation message bits include RL = "1" and MSRC = "0000", as before, the additional values F = L = S = AI = "0" and IT = "1", while the lockout fields are copied from the record of the current transaction. The CB bit must be "0", and CP is also set to "0".

If the Comm-B transaction provided in response to the B bit is not successfully executed while the target is still within the beam, this fact will be recognized by data link processing when it evaluates the replies. Data link processing will then provide a Comm-B request, identified as pilot-initiated, in the active message sublist. This request will be recognized, along with ground requests for Comm-B message transfers, in the next scan. A ground request will be distinguished by the presence of a non-zero MSRC code field, which identifies the device in question. Pilot-initiated requests will have MSRC = "0000", even if they are acquired by transaction preparation from the active message lists. It should be noted that only one of these can be pending at any one time.

Whenever a Comm-B reply is received, transaction update will check the MSRC code that was included in the interrogation (retained in the transaction record). If this code indicates a pilot-initiated exchange, and if the sensor is presently primary for this aircraft, then transaction update will proceed with the protocol. Comm-B state, which must have been "Comm-B pending", is changed to "clear pending", and a clear Comm-B command is prepared. The B bit in any reply from this target will continue to be ignored until a clear command is delivered.

The clear Comm-B command is executed by setting the CB bit to "1" in an appropriate interrogation. Any surveillance or Comm-A interrogation may be used, hence transaction update will search for a suitable transaction, among those pending for the target, and change the corresponding CB bit to "1" in the first of these transactions only. If no suitable transaction is found, none is dynamically prepared. In this case transaction preparation will note the "clear pending" Comm-B state on the next scan, and set CB = "1" in the surveillance transaction for that scan.

If a transaction record is dynamically modified by transaction update to include the clear Comm-B command, it will automatically be the next pending transaction, since only ELM transactions lack the CB bit, and these are always placed after all standard transactions in the target transaction block. Moreover, this relative placement is unchanged by the dynamic addition of transactions under any of the rules discussed above.

The final step of the protocol occurs when a primary sensor receives a reply to an interrogation which contained a clear Comm-B command. Upon finding these conditions satisfied, transaction update will change the Comm-B state to inactive and the message transfer is complete. This reply, which provides technical acknowledgment of the receipt of the clear command, can also have B = "1", which indicates another Comm-B message request from the same or another device. If this is the case, Comm-B state is changed instead to Comm-B pending, and the protocol repeats, starting with the attempt to transfer the new downlink message.

III-12. Extended Length Message Protocols

When messages are transferred by means of the Comm-A and Comm-B formats, each single message segment is treated as a complete exchange, including a technical acknowledgment. Even the so-called "intermediate length message" is sent by means of a sequence of Comm-A interrogations, each of which must be successfully received by the transponder before the next segment can be sent. Similarly, a pilot-initiated message could consist of a string of Comm-B segments, but each would have to be received on the ground and acknowledged (by a clear Comm-B command) before the request for transfer of another segment can be sent to the ground.

For long messages, the ELM formats are used for greater efficiency of transfer. Up to sixteen message segments are sent as part of a single transaction, with a single technical acknowledgment of the group. If some of the segments were not received successfully, a new transaction will be executed which contains only the previous failures, and so on until the exchange is complete. If the message requires more than sixteen segments it is broken up into 16-segment blocks, and each block is treated as a separate ELM.

Only the primary sensor executes the ELM protocol described here. The details of the protocol differ somewhat, depending upon whether the transaction is an uplink or a downlink ELM. The interrogations of an uplink ELM are all Comm-C interrogations, of which three kinds are distinquished by their RTC codes, as follows:

1) RTC = "00" - The first interrogation of the ELM, which delivers a message segment, and initializes the transponder for reception of the message. No reply is elicited.

2) RTC = "01" - Intermediate interrogations which deliver message segments, each identified by a segment number in the SNC field. No replies are elicited.

3) RTC = "10" - Final interrogation of the transaction, which delivers a message segment and elicits the only reply of the transaction.

If the message contains M segments, they will be numbered 0 through M-1, and arranged for delivery in backwards order (starting with segment M-1 and ending with segment 0). This procedure allows the transponder to determine the expected number of segments by making use of the contents of the SNC field of the initializing segment. The single reply is a Comm-D reply, with bit K set to "1", and containing a "cumulative technical acknowledgment" (CTA) in the message field, MD. The first M bits of MD are used to acknowledge receipt of the corresponding uplink segments. If message segment number 0 is received and successfully decoded, the first bit of the MD field will be set to "1" in the reply (otherwise "0"), and so on up to the Mth bit, which corresponds to message segment M-1.

The final interrogation, together with the corresponding CTA reply, is called the "collector" transaction, and it is included in the main portion of a schedule by roll-call scheduling. The other interrogations form part of the precursor, as discussed in Section II-5. Roll-call scheduling may not schedule all of the intermediate interrogations (those with RTC = "01") specified in the uplink ELM transaction record if channel time is inadequate. However, if the transaction is scheduled at all, the collector will be included.

The initializing segment always informs the transponder of the length of the actual message, and clears the transponder CTA field.

Segments which are undelivered during the first transaction attempted will be included in subsequent transactions, regardless of whether they were transmitted but not received, or whether they simply were not sent for lack of time. The responsibility of constructing these subsequent transactions rests with transaction update. If the message is still incompletely delivered when the target leaves the beam, data link processing will arrange for the first attempt on the next scan, while transaction update will again handle subsequent transactions.

When transaction update processes the Comm-D reply of an uplink ELM transaction, it examines the first M bits of the MD field, where M is the length of the ELM, contained in the transaction record. For every "1" in that subfield, transaction update makes an indication of completion of the corresponding message segment. This is accomplished by setting a "delivered" indicator in an appropriate portion of the transaction record. Segments marked "delivered" will be ignored by subsequent operations of roll-call scheduling. When all segments are delivered, the transaction is marked "complete", and it can then be removed from the ELM sublist by data link processing. So long as the ELM remains incompletely delivered and unexpired, it remains in the active message list with the original length, although some segments are marked "delivered".

If the initializing interrogation is not received and the transponder replies to the collector, the CTA field in that reply will indicate all zeroes.

This will be the result of a previous "clear Comm-C" command, as discussed below. In this case transaction update does nothing, leaving the transaction a candidate for repetition. If the initializing segment and one or more others are successfully received, these will be marked "delivered" and not included in subsequent transactions devoted to this ELM. Subsequent transactions of this kind will contain no interrogation with the RTC code "00". When the final segment is received, if others remain undelivered the transaction record will be modified by changing to "10" the RTC code of the last (lowest-numbered) remaining undelivered segment, which then becomes the collector in the next transaction for this ELM.

When the message is either delivered or expired, the corresponding entry will be removed from the ELM sublist by data link processing, which will then formulate a clear Comm-C command, addressed to this target, and insert it ahead of any other ELM transactions on the sublist. Channel management never generates clear Comm-C commands on its own initiative. As a result, the most extensive uplink ELM activity that can take place in one scan is the delivery of a clear Comm-C transaction and the subsequent delivery (with one or more transactions) of a new uplink ELM. The clear Comm-C command is a simple transaction pairing a single Comm-C interrogation with a single Comm-D reply. This interrogation also clears out the CTA field of the transponder, setting all sixteen bits to "0". The contents of the reply to a clear Comm-C command is arbitrary (the reply itself constitutes technical acknowledgment of the command), hence transaction update has no processing to carry out on this reply, limiting conformance testing to the F and L bits.

Downlink ELM transactions originate when the ELM "sent button" is pushed, setting the D bit to "1" in subsequent replies which contain this bit (unsynchronized surveillance and Comm-B replies). If the sensor is primary, data link processing (not channel management) responds by placing a downlink ELM request in the active message list for the target. The number of segments in the pending message is contained in the DCOUNT field (four bits in length) of the reply which had the D bit set.

The downlink ELM transaction consists of a single Comm-C interrogation which elicits a string of Comm-D replies, each of which contains a message segment. The interrogation, identified by RTC = "11", SNC = "0000", specifies which downlink message segments are to make up the reply string. This is done by setting to "1" one or more of the first sixteen bits of the interrogation message field, MC, which correspond to message segments in this context. The actual number of replies to be authorized is fixed by roll-call scheduling on the basis of available channel time. In the original transaction record, as generated by transaction preparation, the MC field will have as many "1"s as there are segments in the downlink message request. When transaction update examines the replies, it knows how many to expect because the number scheduled has been preserved by roll-call scheduling. For each successfully received reply, which contains its own message segment number in the SND field, transaction update will change the corresponding MC bit to "0" in the transaction record and reduce the length field, in that record, by one. Roll-call scheduling, next time it operates, will then see a request for a smaller number of segments, each still identified by a "1" in the transaction record

MC field. Note, that if all requested segments are not actually scheduled in a given transaction, then the MC fields of the actual interrogation and the transaction record will not agree. When all replies are received, after one or more scans, data link processing will remove the transaction from the active message list and insert a "clear Comm-D" command for execution in the next scan, in analogy to the close-out of an uplink ELM exchange.

Downlink ELM replies are sent in rapid succession by the transponder and, since each is a long reply, an excessive number of these replies could deplete the transponder power supply. To prevent this, channel management must keep track of the number of downlink replies requested for each target in each scan. This number must not be allowed to exceed 32 and hence transaction update must check the accumulated number of replies against the number requested, and truncate the latter if necessary to keep the total within this constraint.

III-13. Transaction Block and Active Target List Header Update

The various tasks which make up the processing of accepted replies, described in the six preceding Sections, are largely independent and insensitive to order of execution. However, if the PBUT protocol is exercised after the reaction to the A and B bits, it will occasionally happen that a dynamically inserted transaction can carry a clear PBUT command when no pending standard transactions was available. In many cases a pending transaction can be modified, or a new transaction added, to serve several functions simultaneously.

The final step in transaction update processing of a reply is the setting of the current transaction indicator. If the transaction is to be repeated the indicator is unchanged. This includes the case of a failed surveillance transaction with no more remaining tries in the current DABS period. If the transaction is successful and another, perhaps just added dynamically, is pending, then the current transaction indicator will be made to point to this next transaction. If all transactions have been completed, including any which may have been added to obtain an azimuth measurement, then the target completion bit will be set, and the target transaction block will soon be removed by target list update. In this case, of course, both range and azimuth completion will also be indicated. In the special DABS schedule mode "synchronized, without reply processing," there are no replies but transaction update proceeds as if all transactions were successful.

When transaction update has processed all replies, the active target list will consist of the same target transaction blocks as before, but the blocks themselves will usually have been modified. Some blocks will be unchanged (when the old transaction is to be repeated), others will indicate a new pending transaction and still others will indicate target completion. Therefore, from the point of view of roll-call scheduling, which will next process the list (unless the DABS period has ended) the active target list has changed, and the list header must therefore be updated. The processing involved here is identical to that carried out in the same circumstance by target list update. Whenever a target transaction block is modified, the channel time estimate sums must be changed by subtracting the old channel time estimate and adding in the new one, if any. Uplink ELM pointers, length and channel time estimates are updated in an obvious way, including the modification resulting from partial delivery of such an ELM. This header updating must be carried out even if the DABS period has ended, since target list update will only modify the active target list header in the case of additions and deletions of target transaction blocks. It is expected that transaction update can accomplish both reply processing and active target list header update in a single pass through the list. An illustrative flowchart is given in the next Section.

III-14. Algorithms and Flowcharts

Transaction preparation is illustrated in the flowcharts of Figures 23 through 33. The first flowchart shows a version of the loop for fetching new targets from the surveillance file. The algorithm is entered with a new azimuth limit, θ_{new}, and the last step is to save this value as θ_{old}. The startup procedure is not shown, hence the algorithm always starts with the two azimuths, θ_{old} and θ_{new}, which define the boundaries of the required between-limits search on earliest likely azimuth. It is also assumed that the surveillance file is not empty, so that the target pointer used by this program is initially set to the address of the first target which failed to be accepted in the last operation of the algorithm. The earliest likely azimuth of the indicated target is stored in the variable, θ.

The value of θ is tested against the new θ_{new}, and if $\theta > \theta_{new}$ it means there are no targets in the azimuth wedge in question. In this case no target transaction blocks are prepared, θ_{new} replaces θ_{old} and the operation is complete.

For each target that does lie within the limits of azimuth, a closure bit is set in the surveillance file and a target transaction block is prepared.

The closure bit denies access to this target's surveillance file record by other DABS functions while channel management is concerned with the aircraft. The bit is reset by data link processing after the target has left the beam.

Target transaction block preparation itself is expanded in the subsequent ten flowcharts. When transaction block preparation is complete for one target, the target pointer is advanced to the next target, in order of increasing earliest likely azimuth, the value of which replaces the old contents of θ. The loop continues with the test on this new value of θ. It should be noted that the surveillance file record contains a "roll-call inhibit" bit, controlled by network management to limit the surveillance load of the sensor. Transaction preparation will bypass any record with this bit set, and the track itself will be maintained on data obtained from another sensor.

Target transaction block preparation is outlined in Figure 24. All surveillance file data on the target is available at this time, including the active message list for this target. The step "initialize entry indicators" causes pointers to be set to the first entries on the Comm-A, Comm-B and ELM sublists of the active message list. If a sublist is empty or exhausted, the pointer will so indicate. When a message from one of these sublists is included in a transaction, the value of this pointer (or index) is copied into the entry indicator field of the transaction record.

Target record preparation, expanded in Figure 25, takes place first, followed by preparation of the surveillance transaction (Figure 26). If ATCRBS identity has been requested, an A-transaction is prepared (Figure 27), and if the synchro bit is set, an S-transaction (Figure 28) is also prepared. Additional standard transactions are prepared (Figure 29) until the Comm-A and Comm-B sublists are exhausted.

The preparation of each of these standard transactions involves one or more of the steps labeled "DL procedure," "A procedure" and "B procedure." The DL procedure is invoked if a DABS lockout transition is indicated by the corresponding control bit and lockout state. The A and B procedures cover the inclusion of Comm-A messages and Comm-B message requests, when appropriate. The three procedures are shown in Figures 30, 31 and 32.

The final step, shown in Figure 33, covers the inclusion of ELM transactions. Only one uplink ELM (not counting clear commands) transaction record is ever prepared for a single target, and a check for this is implicit in the last test of Figure 24. The ELM sublist will never contain more than one clear Comm-C, one clear Comm-D, and one downlink ELM at any given time.

Figures 25 through 33 require no further comment, except that target record and transaction record fields are referred to by number as they appear in the lists of Section III-4.

The target list update subfunction is flowcharted in Figure 34. Its input is the old active target list and the set of target transaction blocks obtained from the transaction buffer. The latter is referred to here as the "new target list." The range-ordering operation is not shown in detail, but is indicated as a preliminary ordering of the new target list (NTL) and a fetch of the next target for the new active target list (ATL) from either the old ATL or the NTL. This ordering of the new list and the merging of new and old targets to form the new ATL is a standard merge-sort procedure, using a masked range delay word, as explained in Section III-5.

New targets always have the surveillance transaction pending, which is a priority transaction. Therefore, the only ATL header field requiring update in this case is the priority channel time sum.

Each old target is tested for continued visibility by comparing its last likely azimuth (θ_{LLA}) against the current cutoff azimuth (θ_{cut}). When targets are deleted for this reason, the ATL header fields representing channel time sums must be corrected. The channel time estimate of the pending transaction is subtracted from the appropriate sum or sums. The ensuing range completion test is explained below.

For targets which are still within the beam, target completion is tested, and if range and azimuth are also complete, the target will be deleted from the old ATL and added to the released target list (RTL), along with targets no longer visible. No ATL header update is required, since completed targets are not included when the ATL header update is performed by transaction update.

If target completion is set, but either range or azimuth is incomplete, then target completion is reset to "0" and the channel time estimate added to the relevant ATL header field(s), since this target is about to be restored to the list. These targets, along with new and incomplete targets are merged into the new ATL. After such a merge, or a deletion, the next target is chosen from either the old ATL or the new target list, as appropriate, and the program continues until all targets are processed. Details covering the situations where one (or both) of the input lists is empty are omitted here.

When a target is removed from the ATL because it has fallen behind the beam, a special test is made to single out the case in which range completion is not set and the DABS lockout transition indicator is set. This is the case of a target which requires at least two successful transactions to achieve the desired effect with respect to DABS lockout. The first interrogation must have DL = "00", while the second (and all subsequent) must have DL = "01". Data link processing always looks for the lockout transition indicator in the target record, and when it finds the bit set, data link processing will change the DL field in the surveillance file to "01" to prevent repetition of this lockout sequence. Note that transaction preparation only initiates this protocol if the transition indicator is set and also DL = "00" in the surveillance file. If no interrogations were successful (no range completion), this activity of data link processing must be inhibited, and to accomplish this the transition indicator in the target record is reset to zero. Next scan, transaction preparation will again find DL = "00", with the transition indicator set, in the surveillance file where it has not been changed.

Figures 35 through 45 illustrate the operation of transaction update. The first flowchart, Figure 35, shows an algorithm for passing through the active target list, assessing each reply to a scheduled interrogation and simultaneously updating the relevant ATC header fields. Roll-call scheduling leaves behind an indication of which transactions were actually scheduled so that transaction update will know what replies to expect.

The examination of a reply and the resulting update of the corresponding target's transaction block are shown as "reply assessment," and this step is expanded in the remaining ten flowcharts. There are three possible outcomes of reply assessment:

1) The same transaction will be repeated (perhaps with some modification in the case of an ELM),
2) The transaction was completed and a new transaction is now pending for the target, or
3) All transactions have been accomplished (for the present DABS period, at least) and the target completion bit has been set.

The algorithm sorts out these three cases and makes the appropriate updates of the channel time sums and uplink ELM fields in the ATL header. The final step is the update of the first target range delay field in the ATL header, when required. The flagging of the situation when the first target on the old ATL becomes completed, and the identification of the target replacing it are not shown specifically, but these steps can obviously be carried out along with reply assessment in a single pass through the ATL.

The first act of reply assessment is the appraisal of replies, sorting them into accepted, not accepted and missing categories, as illustrated in Figure 22. This is shown in flowchart form in Figure 36, in which the procedures to be followed after the sorting are summarized. These procedures are expanded in subsequent flow charts. Conformance testing, a straightforward checking of reply bits, is not expanded further. It is described in detail in Section III-6.

The missing reply procedure is given in Figure 37. If the transaction is not a surveillance transaction no action is taken, which causes the transaction to remain pending for execution. In particular, the current transaction indicator is not changed. Surveillance transactions are repeated until a total of (NTRY + 1) attempts have been made, with reinterrogation to all but close-in targets taking place at high power. As explained in Part II, final authority for the use of high power rests with roll-call scheduling. When the allotted number of attempts has been made, target completion is set, the high power flag removed, and the "remaining tries" counter reset.

When a reply is not accepted, the transaction is usually repeated, shown in Figure 38 as no further action. Synchronized transactions, however, are not repeated in this case, and it is assumed that the transaction (including uplink message and control information delivery) was successful, since a reply was actually received. No attempt is made to extract information from such a reply. Although not shown explicitly on the flowcharts, in the case of a synchronized schedule mode, without reply processing, transaction update will designate all replies as "not accepted," although no reply data blocks are generated. This has the effect of treating all the corresponding transactions as if they were successful.

The more involved procedure for accepted replies begins with Figure 39. For standard transactions, range completion processing is shown in detail, but azimuth completion is deferred to Figure 40. The ATCRBS identity request protocol is given explicitly, but the "modification or addition" of a transaction for this purpose is not further expanded. Full details are given in Section III-9.

The Comm-B channel protocol is also summarized, and expanded in Figure 41. The PBUT response is shown, but the details of setting the CP bit in an appropriate transaction are skipped (see Section III-10).

The treatment of ELM transactions is summarized in Figure 42, with expansions in Figures 43 and 44. The final step in reply assessment, except for the case of repetition of the transaction, is the update of the current transaction indicator, shown in Figure 45.

The detailed flowcharts of Figures 40 through 45 are self-explanatory and they follow exactly the protocols described in the corresponding Sections.

Symbols and abbreviations used in the flowcharts are collected below.

Transaction preparation (Figures 23 through 33):

θ_{new}	New azimuth limit received with enabling command
θ_{old}	Previous value of θ_{new}, saved from last action of the algorithm.
θ	Earliest likely azimuth of a given target.
AI	ATCRBS identity request bit (target record)
S	Synchro bit (target record)
AML	Active message list
NTRY	System parameter regulating interrogations per DABS period.
PPMINR	System parameter, regulating the power programming of close-in targets.

CTSS, CTSL	System parameters representing channel time estimates for individual transactions.
DL	DABS lockout bits (target record)
AL	ATCRBS lockout bit (target record)

Target list update (Figure 34):

NTL	New target list
ATL	Active target list
RTL	Released target list
θ_{LLA}	Last likely azimuth of a given target
θ_{cut}	Cutoff azimuth received with enabling command.

Transaction update (Figures 35 through 45):

UELM	Uplink ELM transaction
DELM	Downlink ELM transaction
CTE	Channel time estimate of a given transaction
CTI	Current transaction indicator
SIG	Monopulse signal
MBL, MBH	System parameters bounding usable portion of monopulse characteristic.
θ_{off}	Off-boresight azimuth
θ_{half}	Half-beamwidth
θ_{bore}	Boresight direction

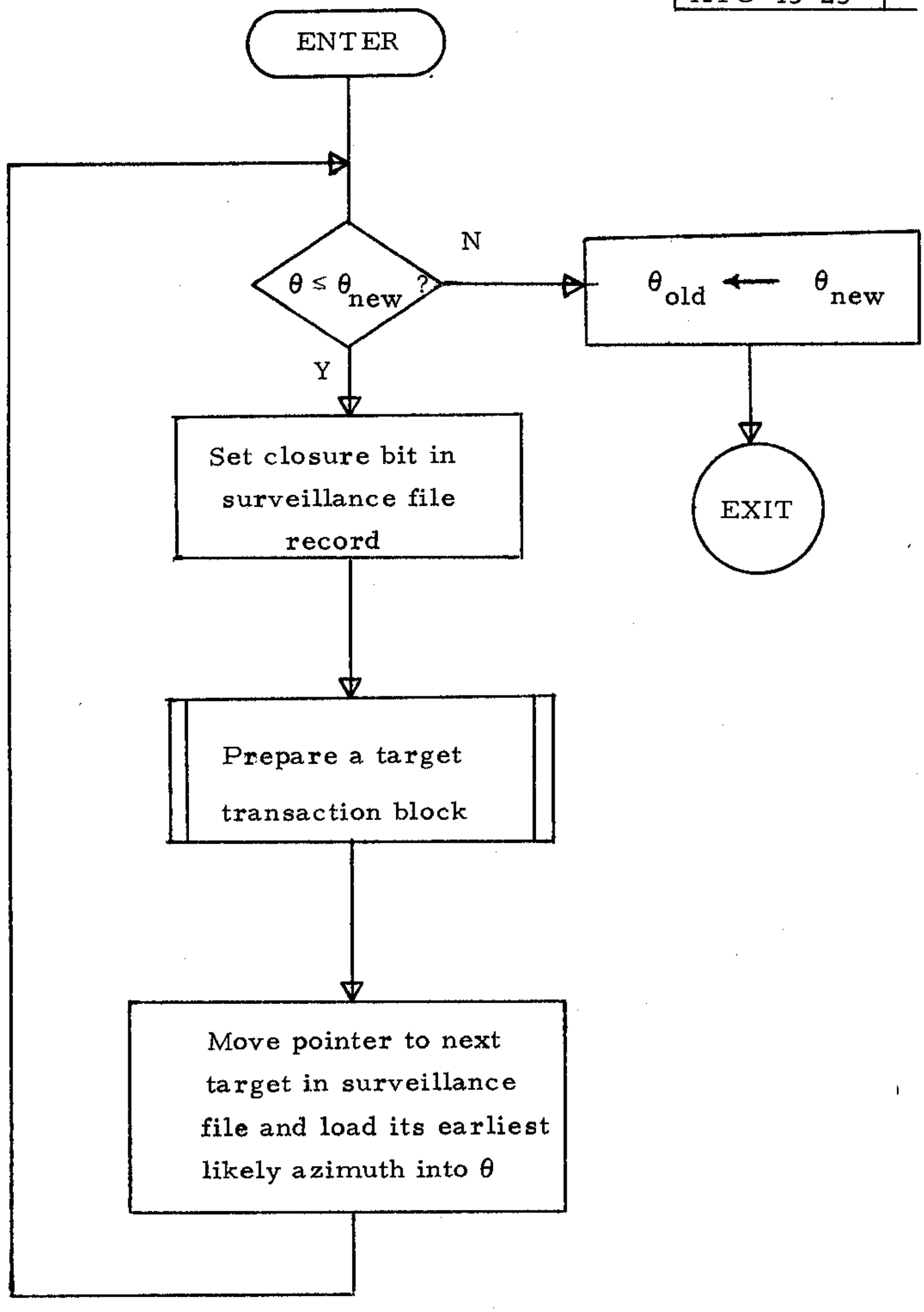


Figure 23. Transaction Preparation

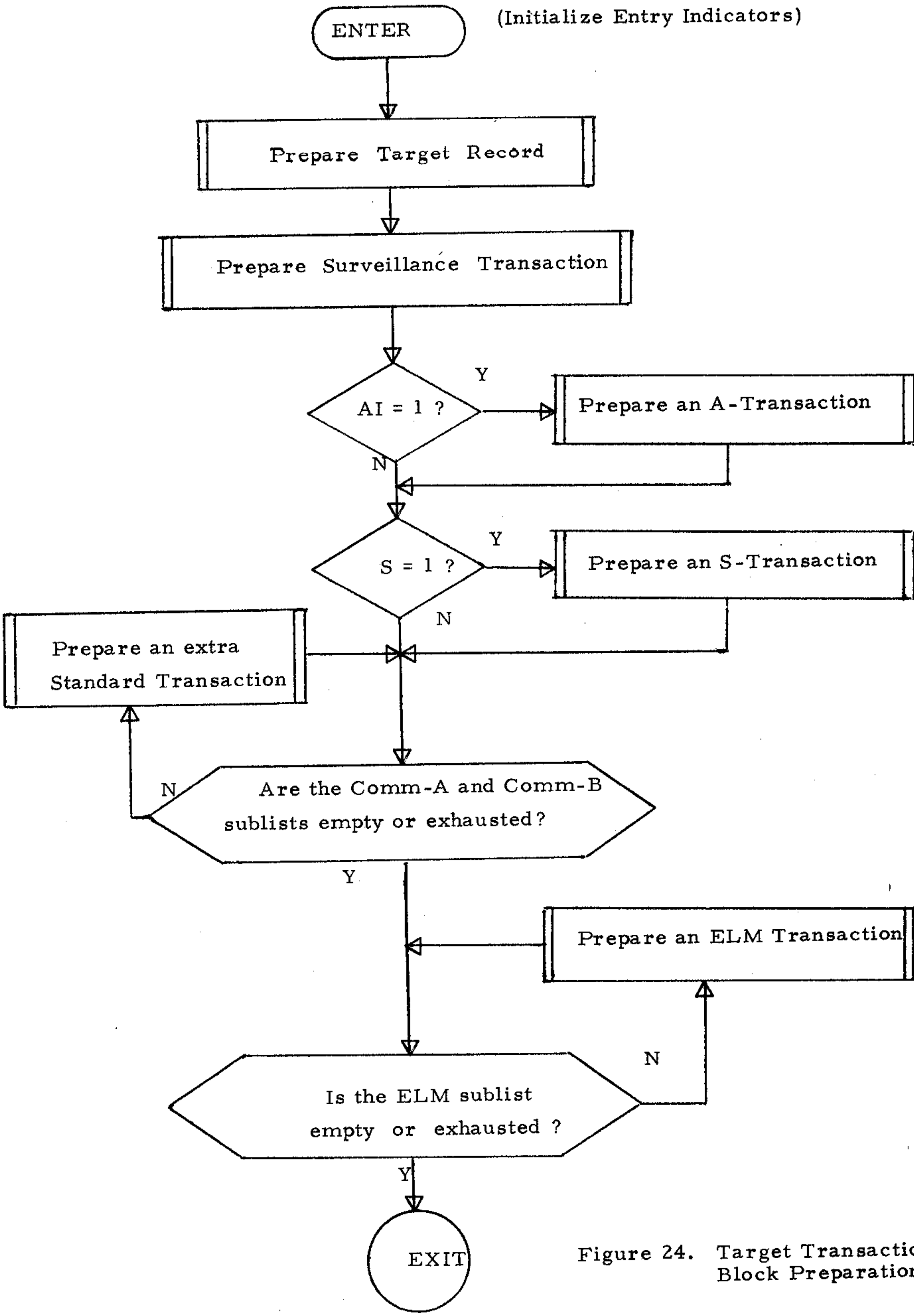
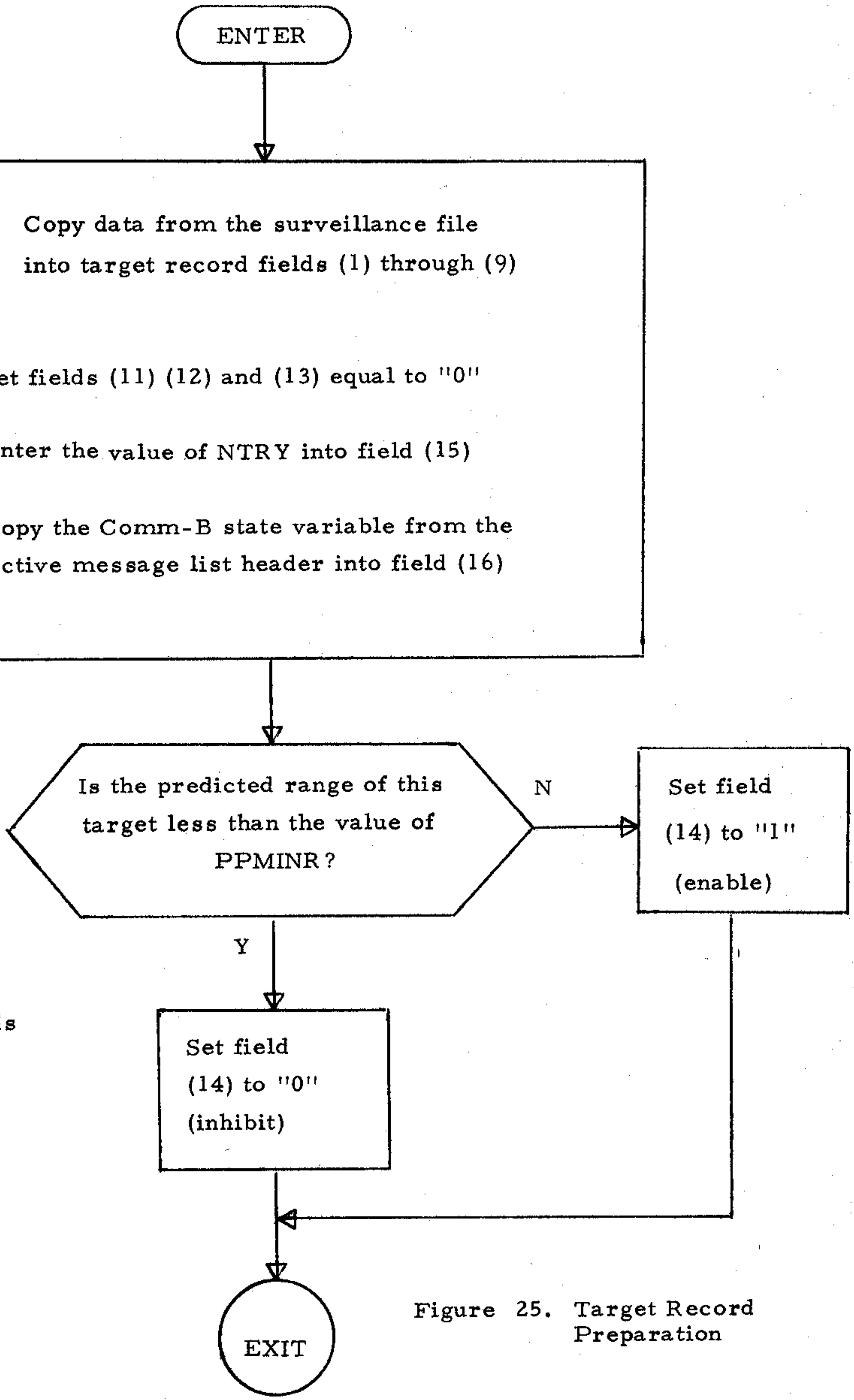


Figure 24. Target Transaction Block Preparation



Target record fields are listed in Section III-4

Figure 25. Target Record Preparation

Transaction Record fields are listed in Section III-4

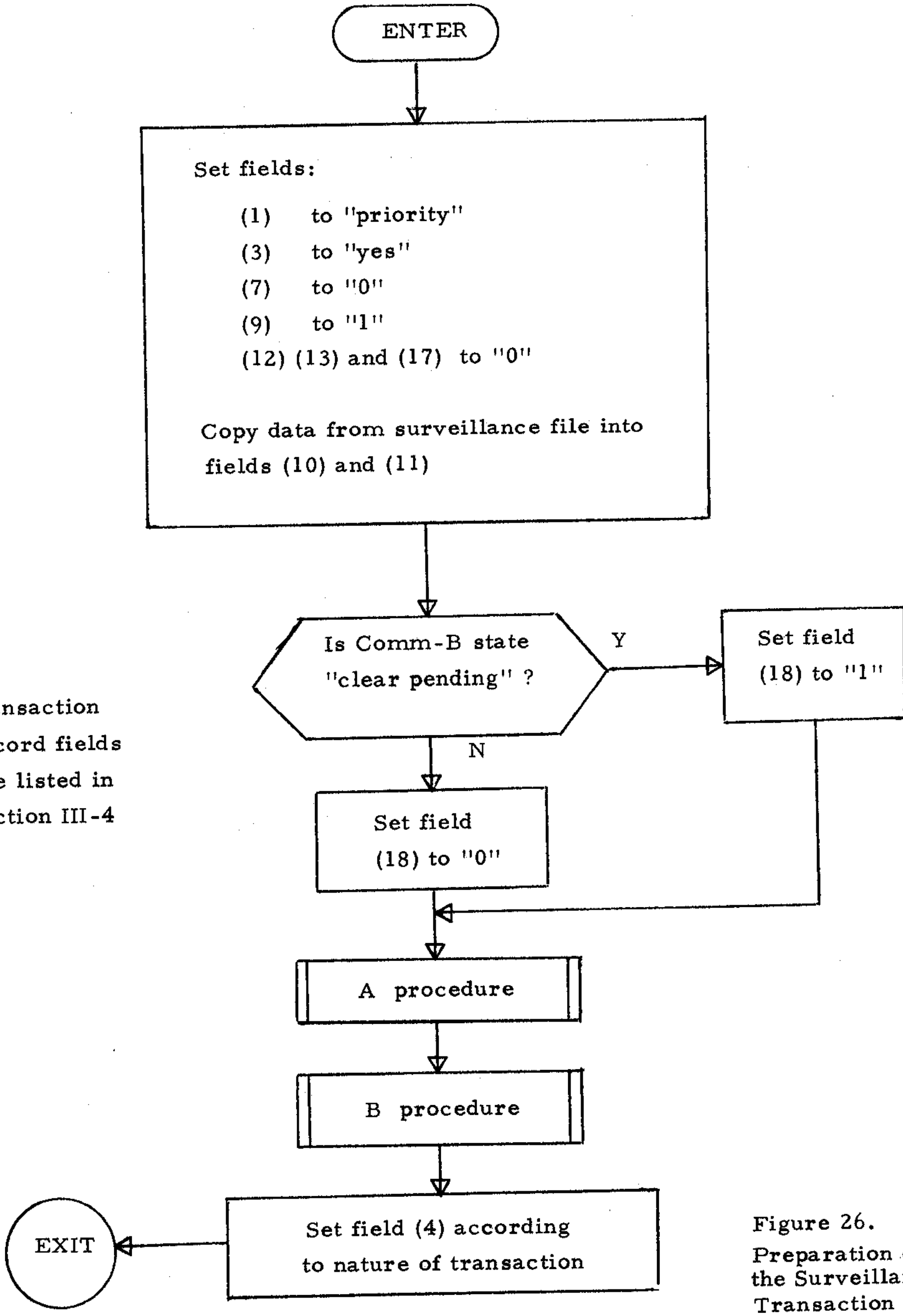
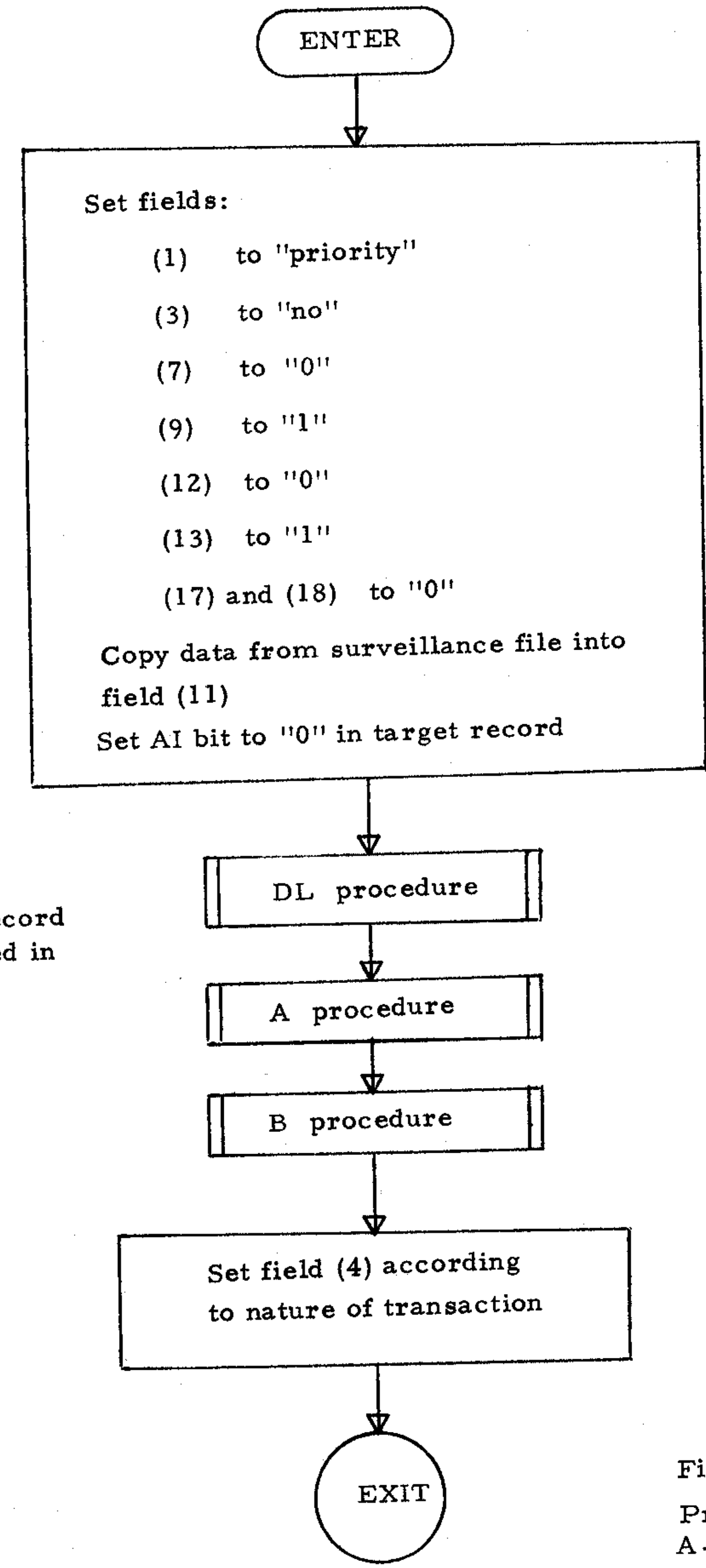


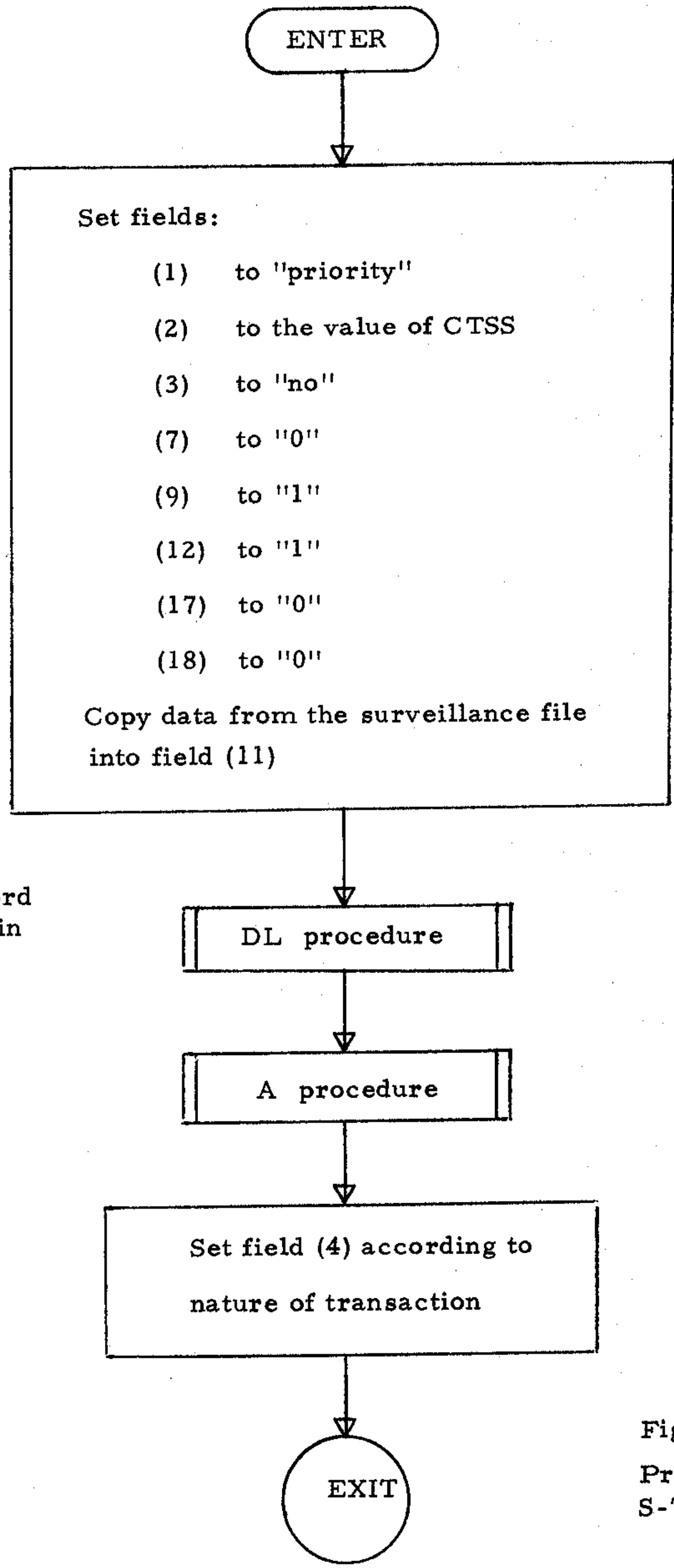
Figure 26. Preparation of the Surveillance Transaction



Transaction Record fields are listed in Section III-4

Figure 27.

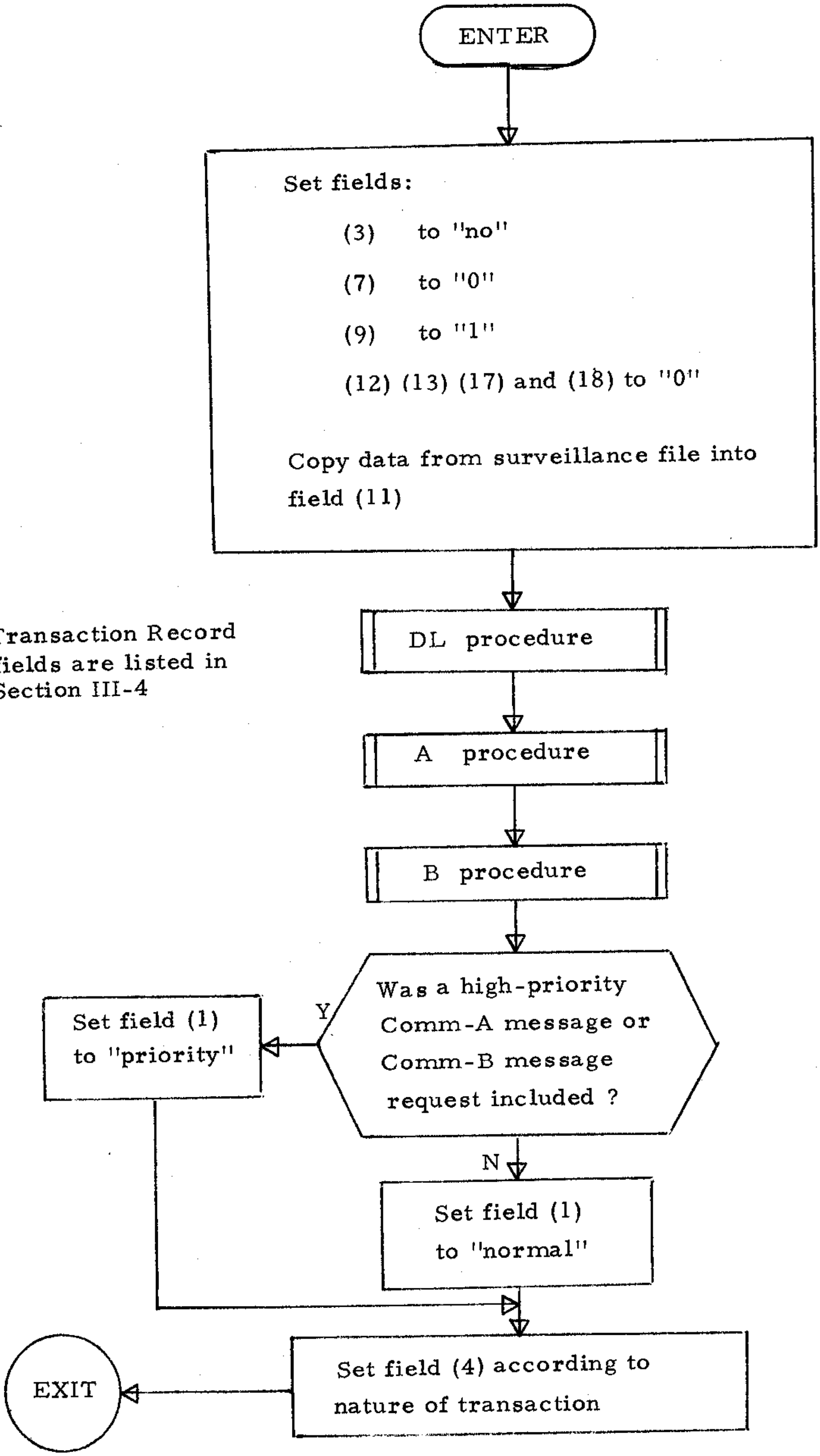
Preparation of the A-Transaction



Transaction Record fields are listed in Section III-4

Figure 28.

Preparation of the S-Transaction



Transaction Record fields are listed in Section III-4

Figure 29. Preparation of Extra Standard Transactions

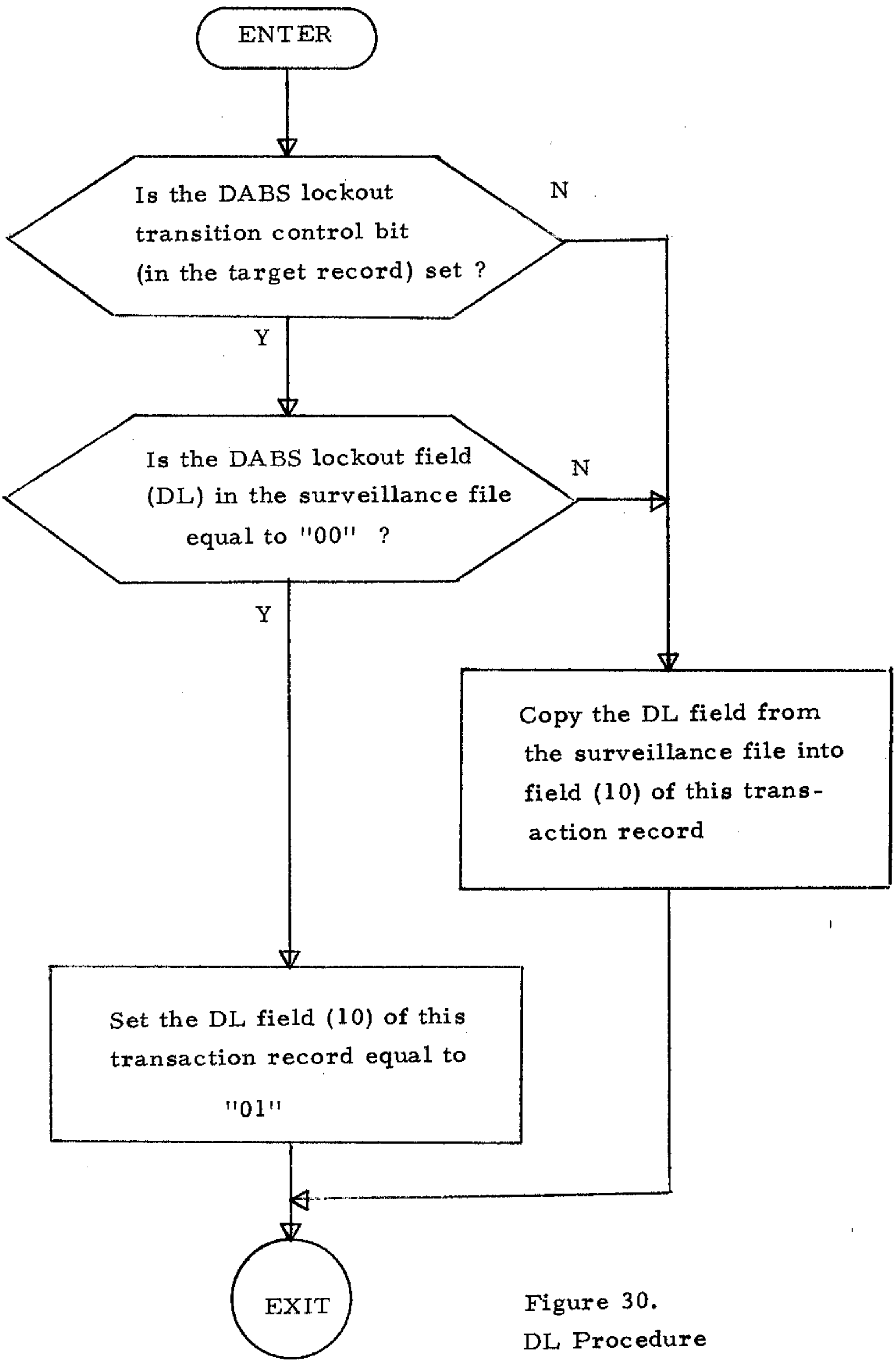


Figure 30.
DL Procedure
(DABS lockout transition control)

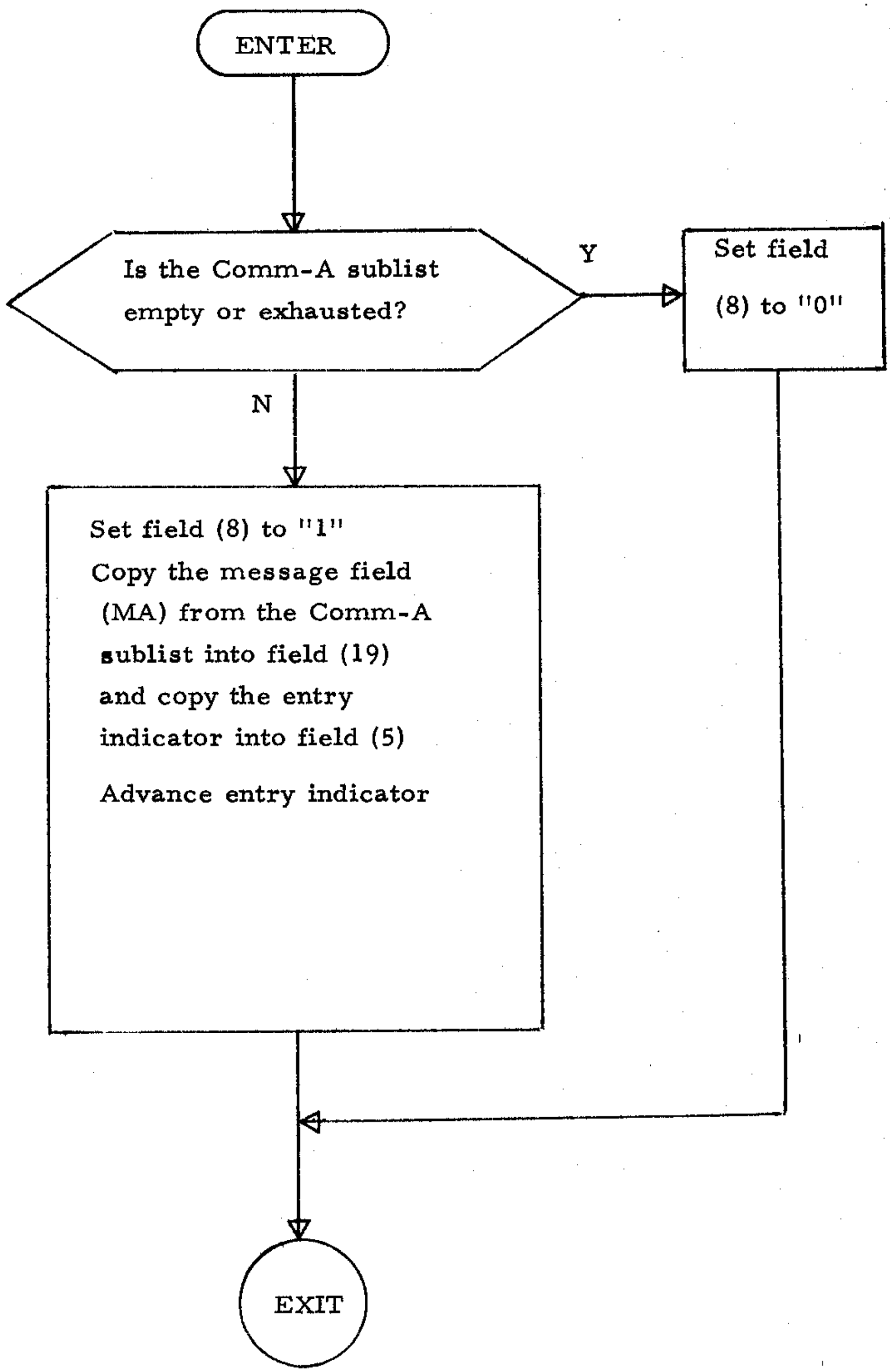


Figure 31. A Procedure

(Inclusion of Comm-A message)

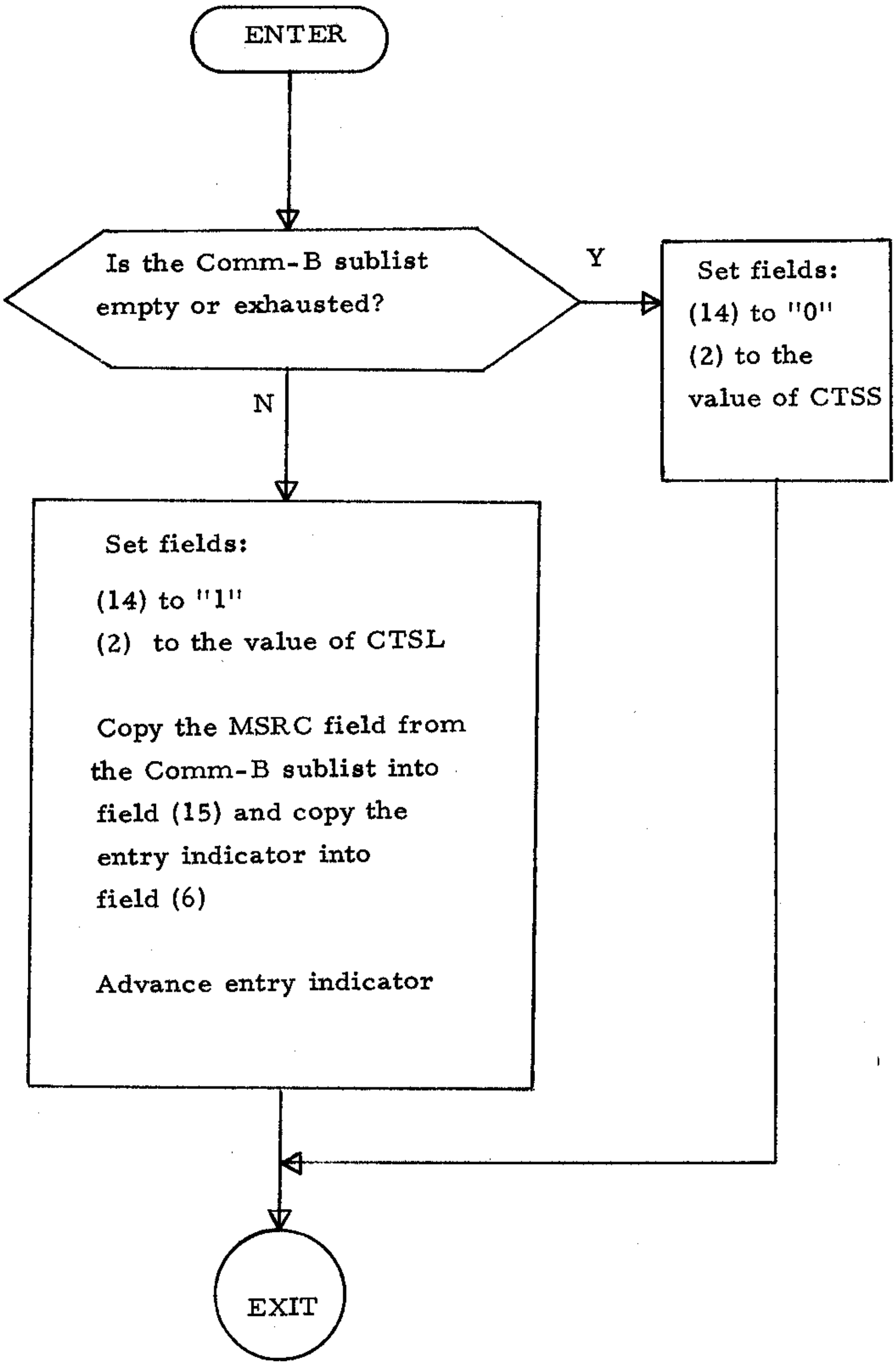


Figure 32. B Procedure

(Inclusion of Comm-B message request)

ENTER

Set fields:

(1) to "normal"

(2) to value of CTSL

(3) to "no"

(7) and (8) to "1"

Copy information from the ELM sublist into field (6) and copy the entry indicator into field (5)

Copy information from the ELM sublist into fields (9) (10) and (11) -- once for a DELM or clear command, and once for each segment of an uplink ELM

Set field (4) according to nature of transaction

Advance entry indicator

EXIT

Note: field (2) will be ignored by roll-call scheduling for a DELM transaction

Figure 33.
ELM Transaction Preparation

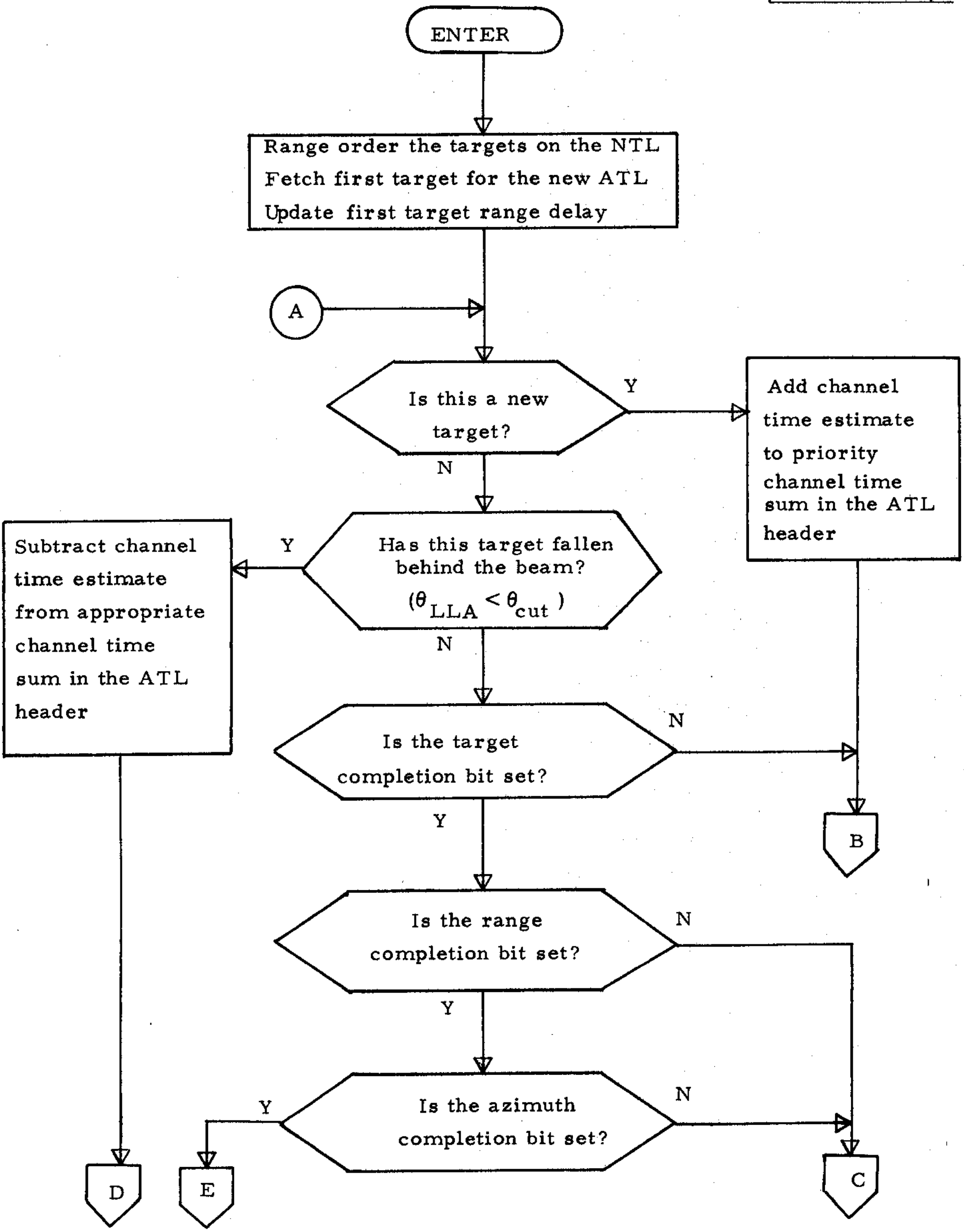


Figure 34a. Target List Update

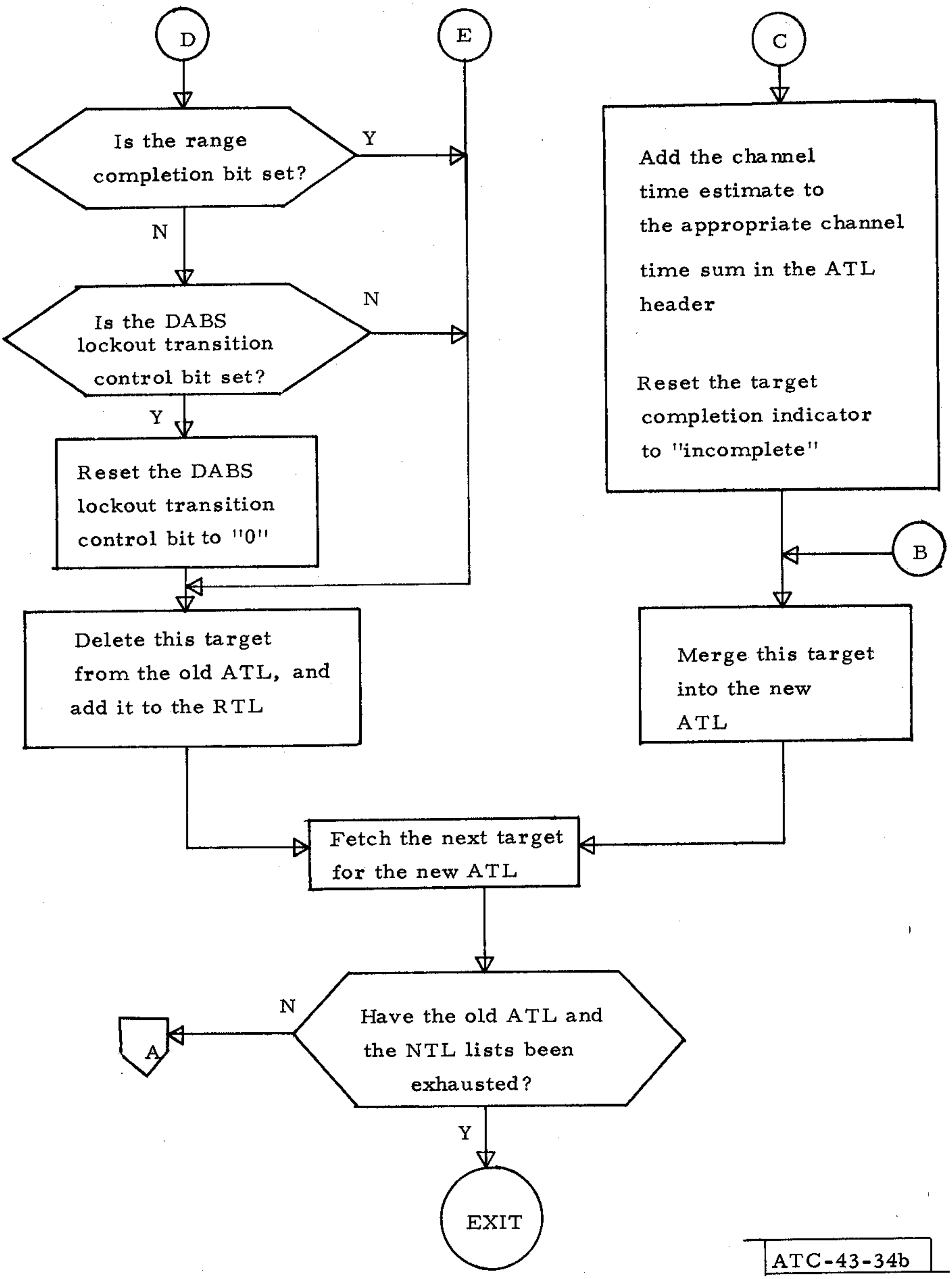


Figure 34b. (Target List Update)

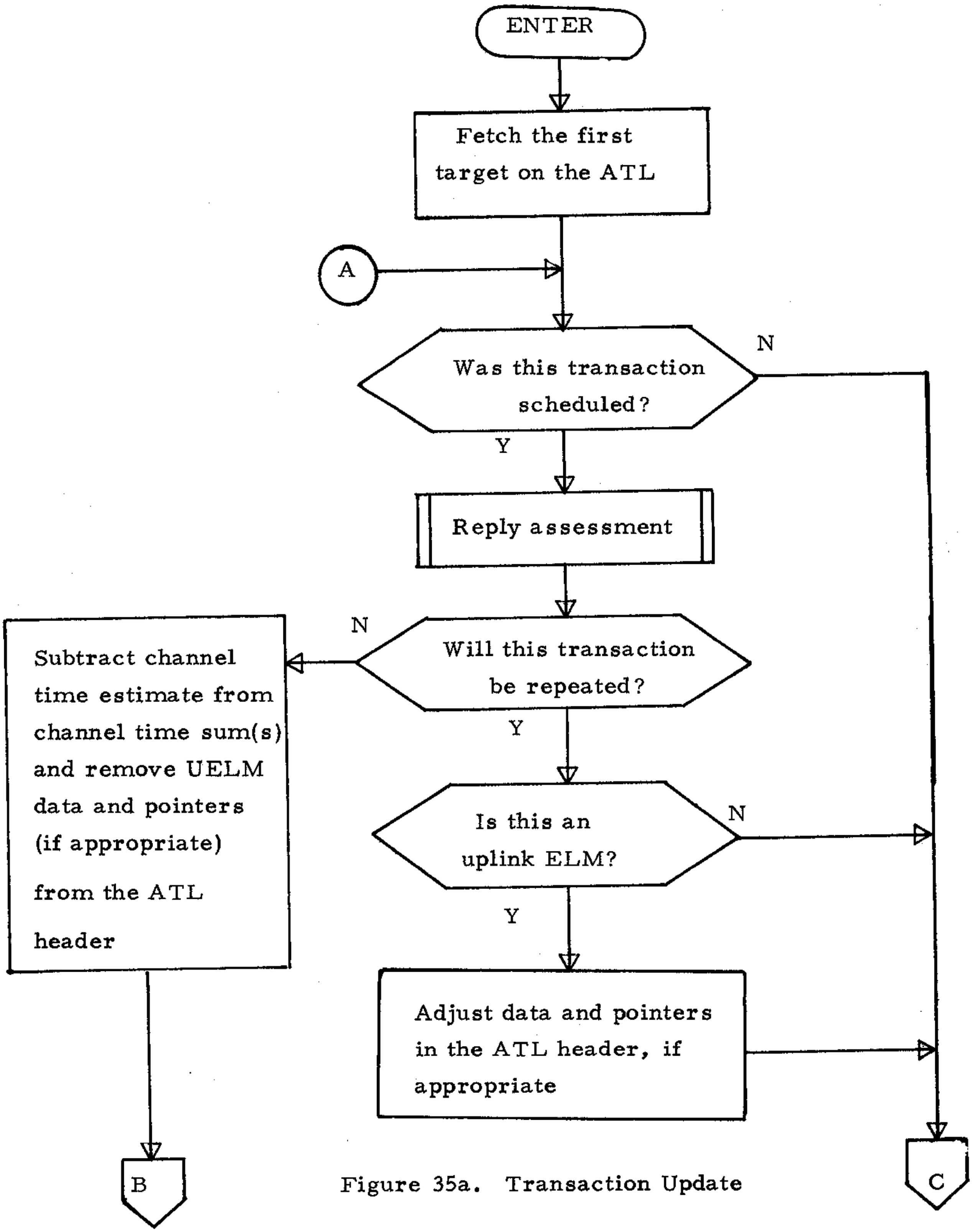


Figure 35a. Transaction Update

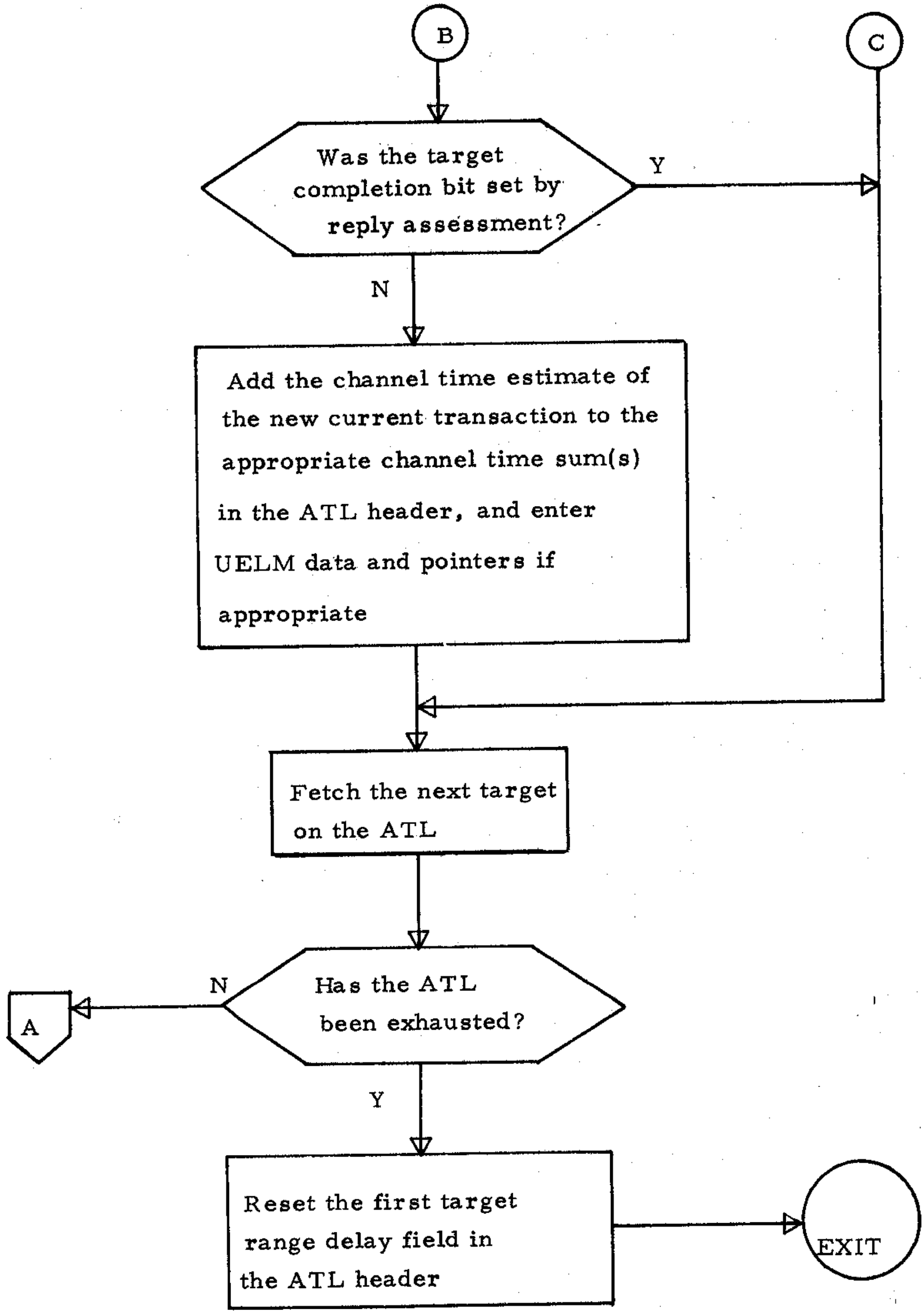


Figure 35b. (Transaction Update)

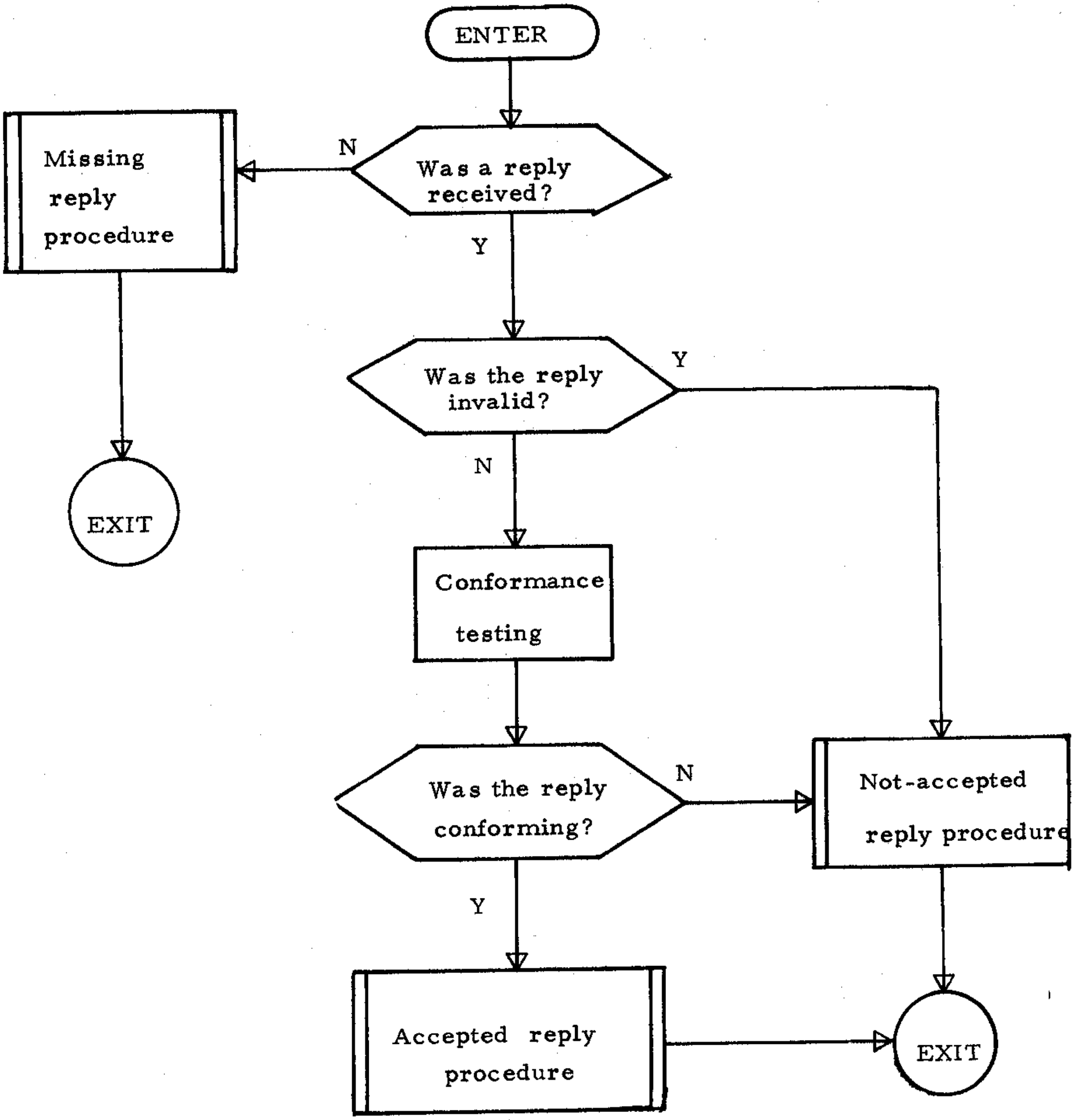


Figure 36. Reply Appraisal

ATC-43-37

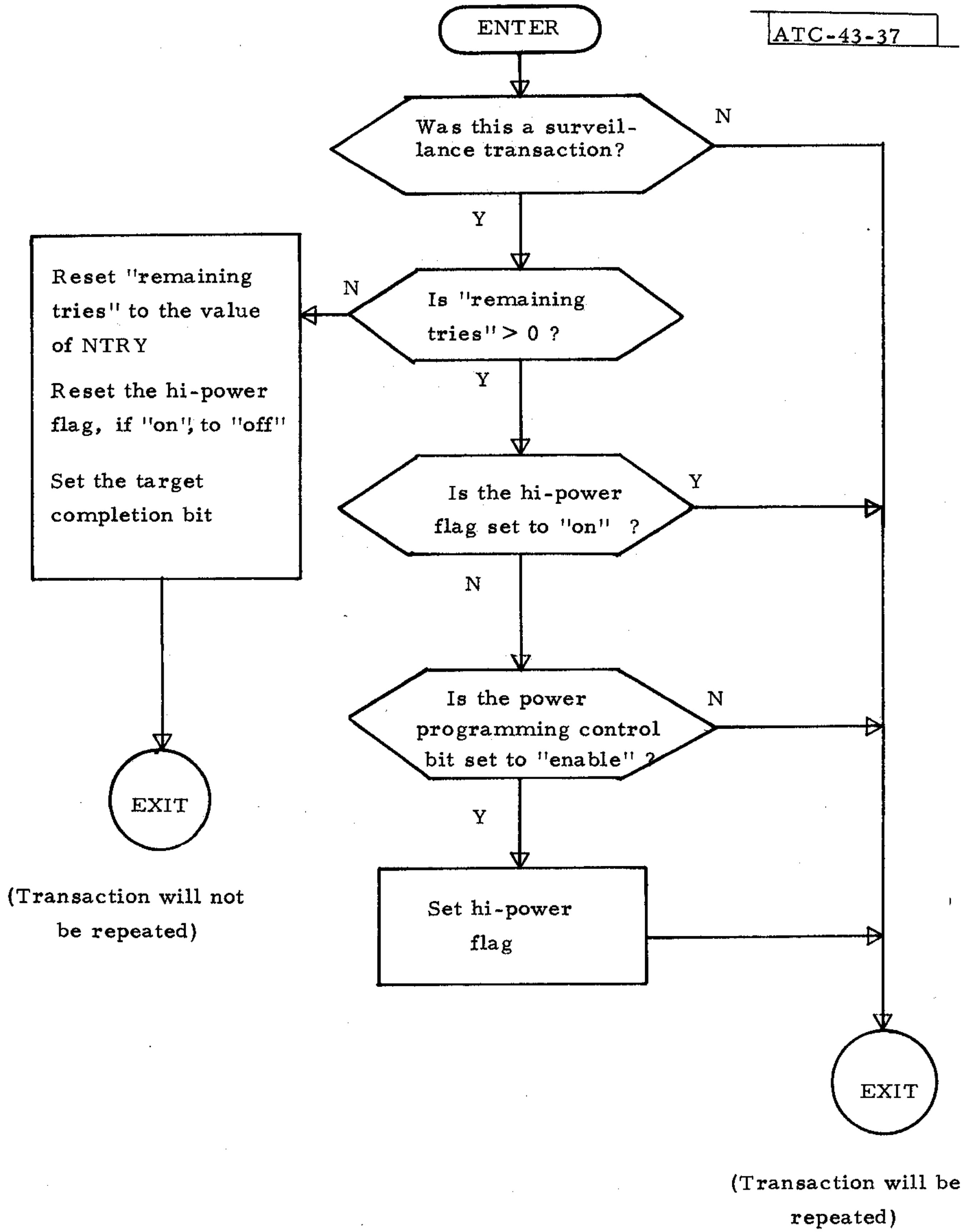


Figure 37. Missing Reply Procedure

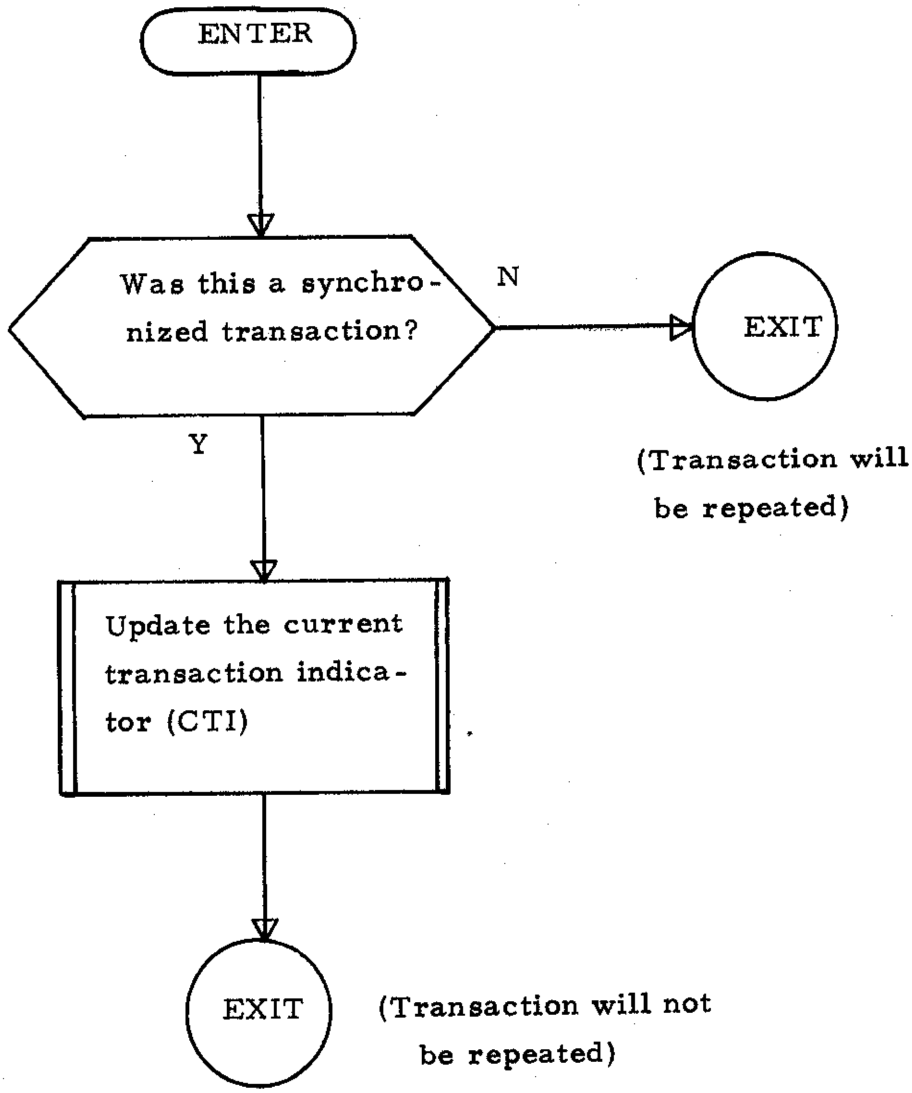


Figure 38. Not-accepted Reply Procedure

ATC-43-39a

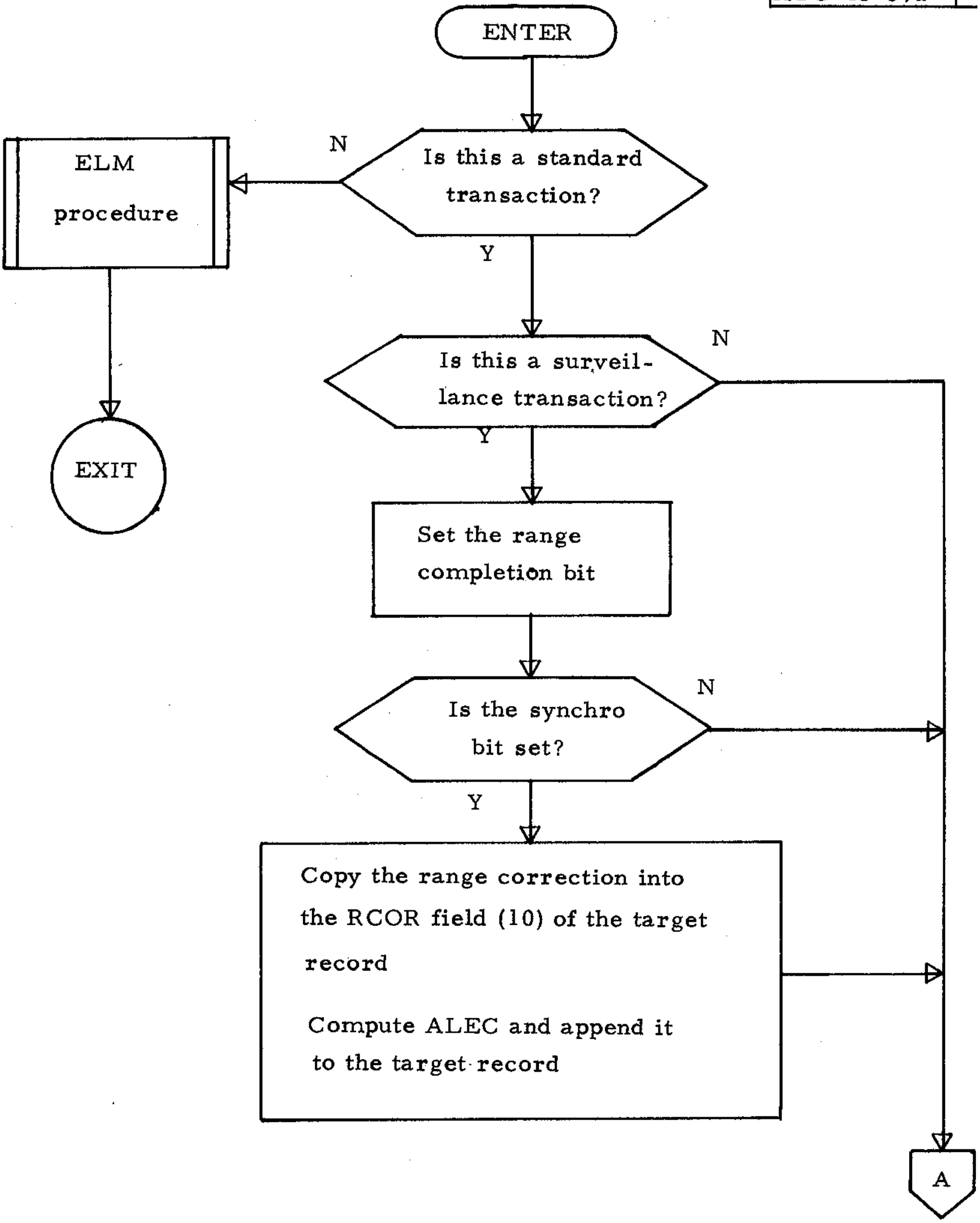


Figure 39a. Accepted Reply Procedure

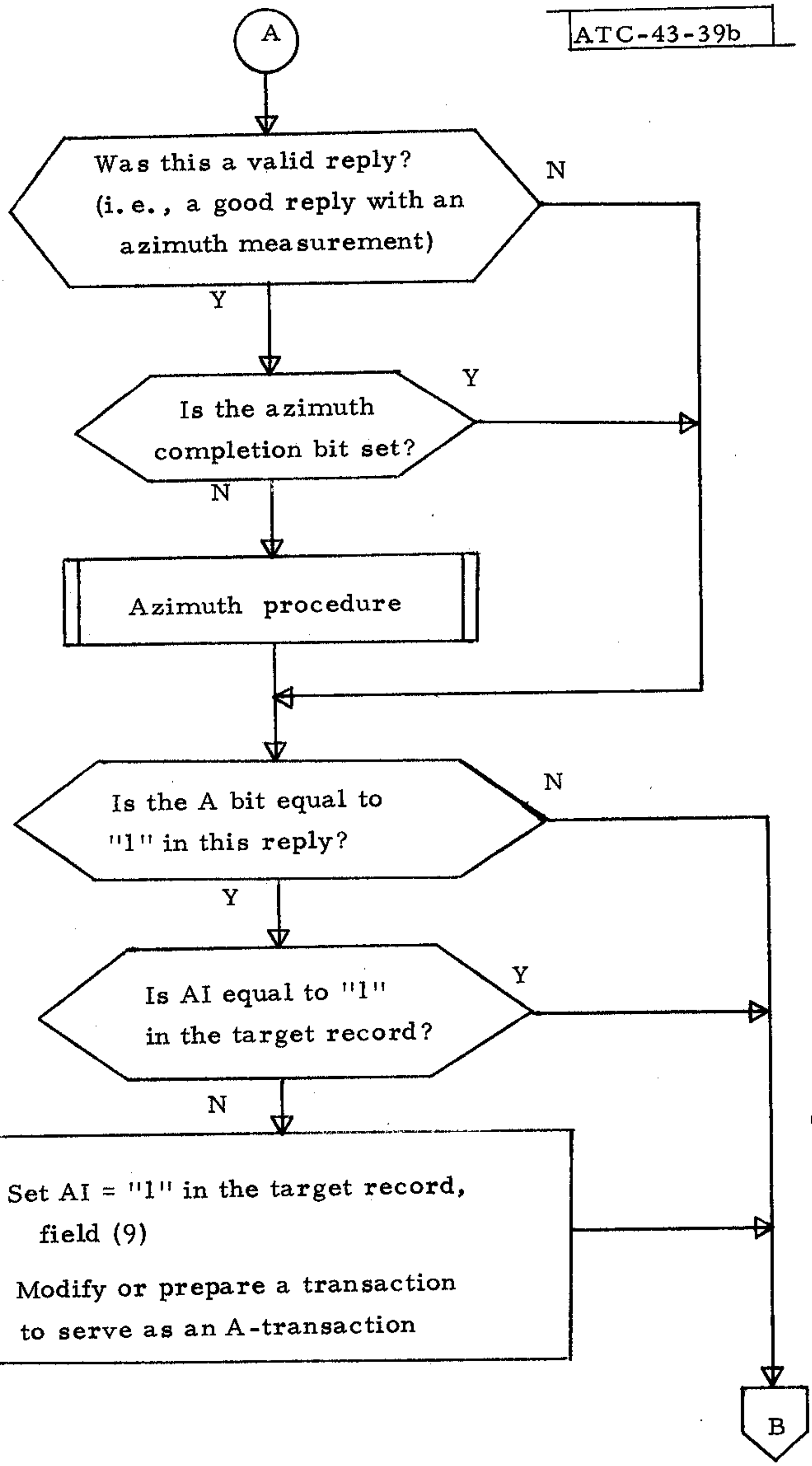


Figure 39b. (Accepted Reply Procedure)

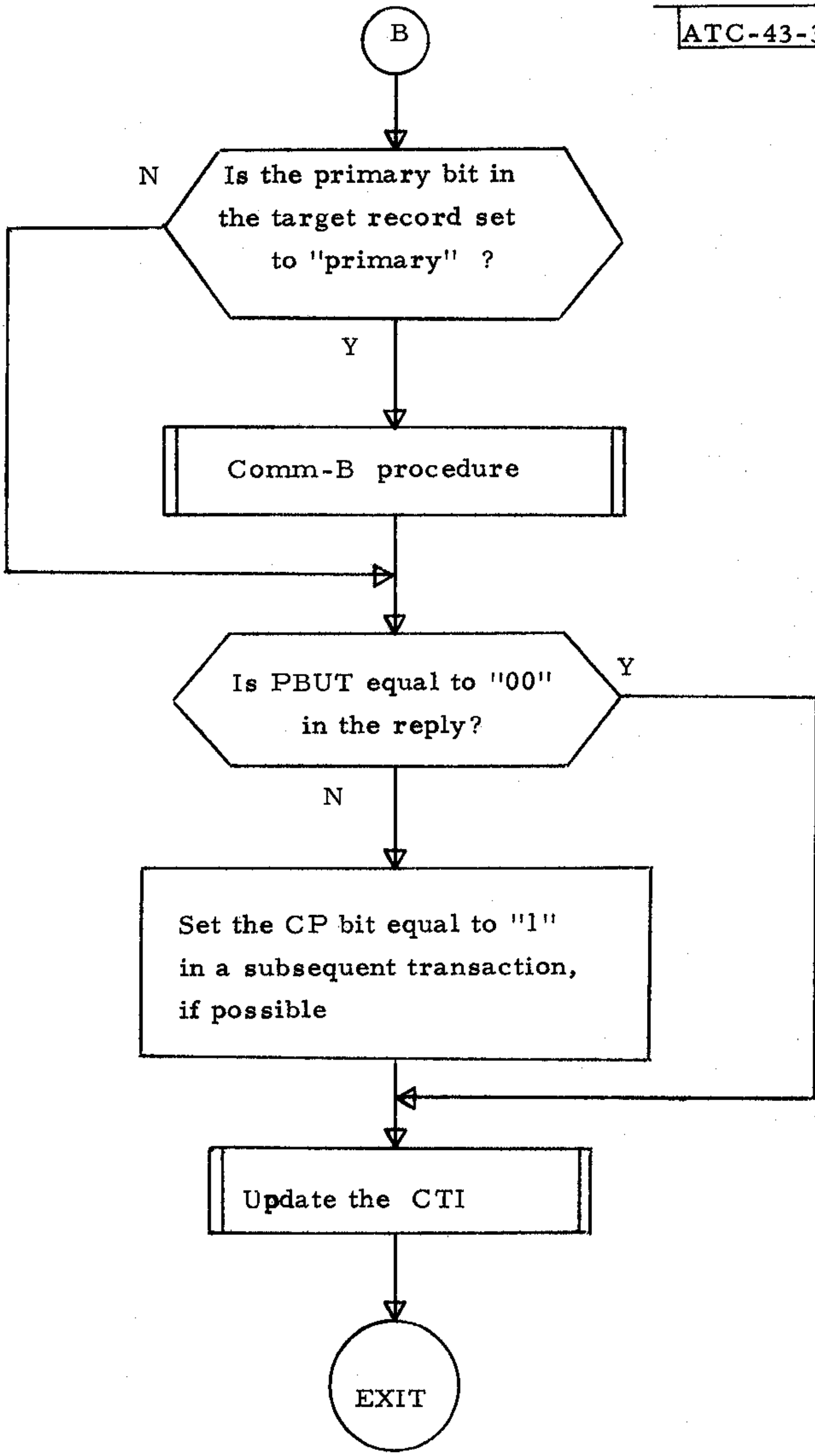


Figure 39c. (Accepted Reply Procedure)

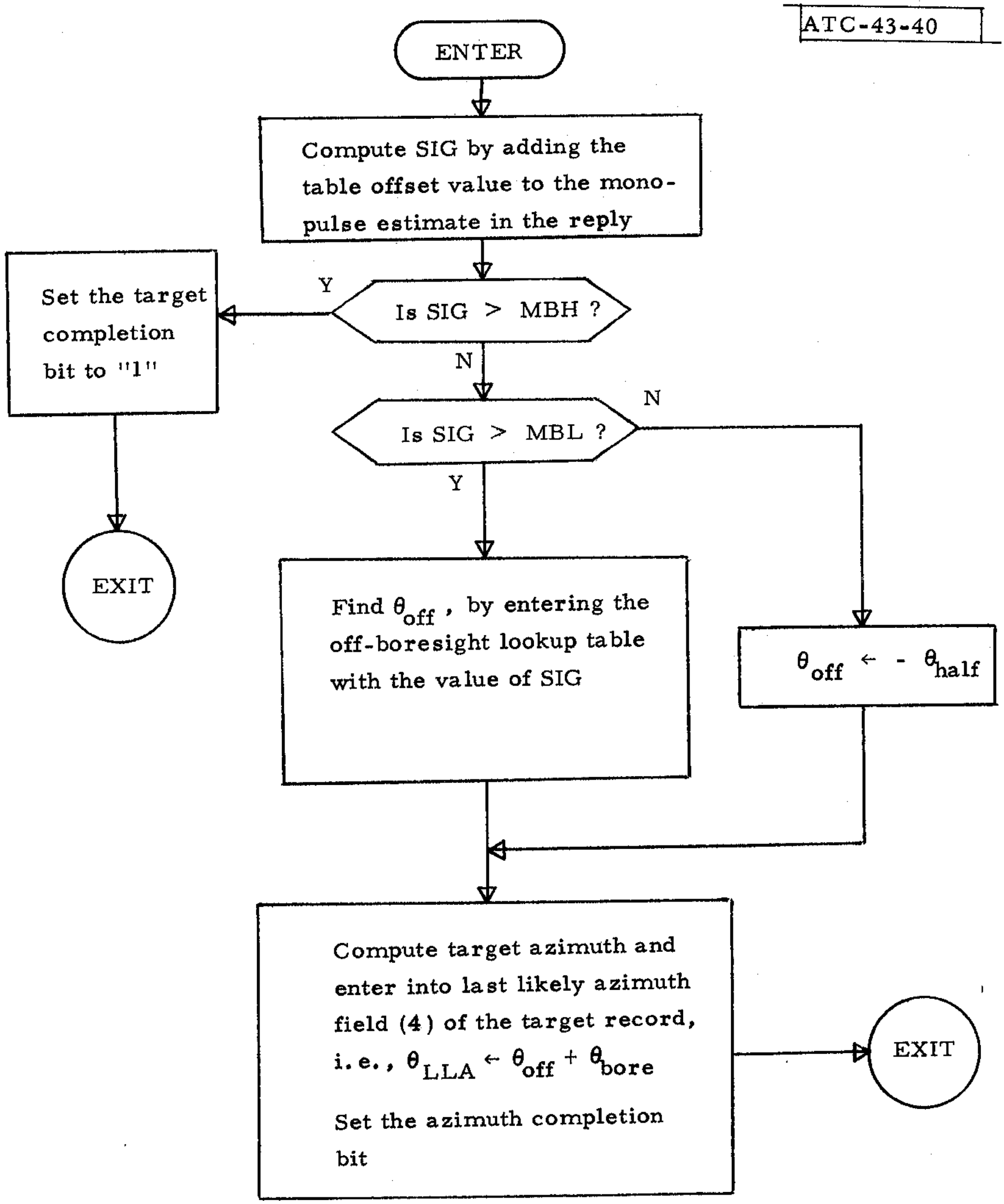


Figure 40. Azimuth Procedure

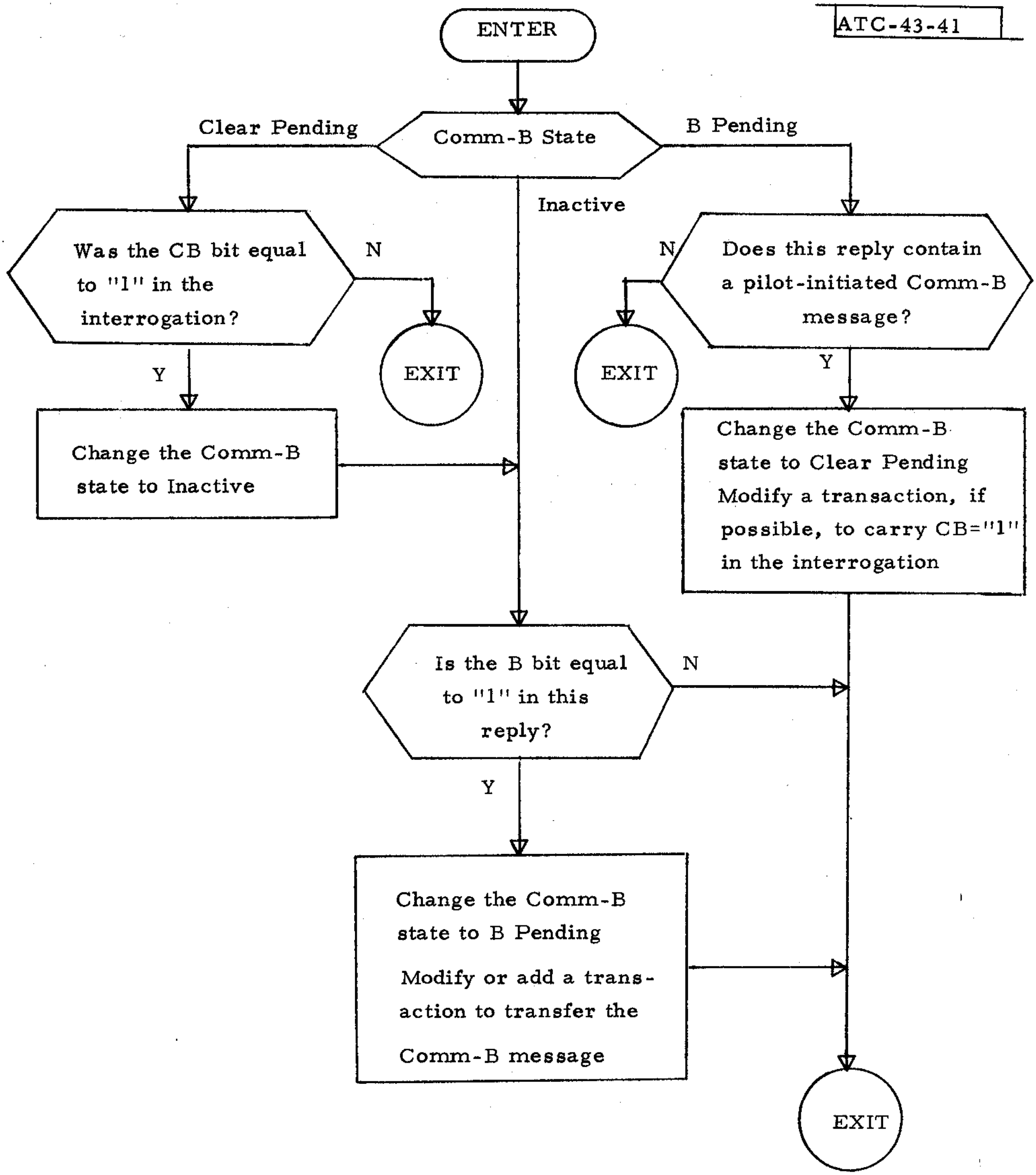


Figure 41. Comm-B Procedure

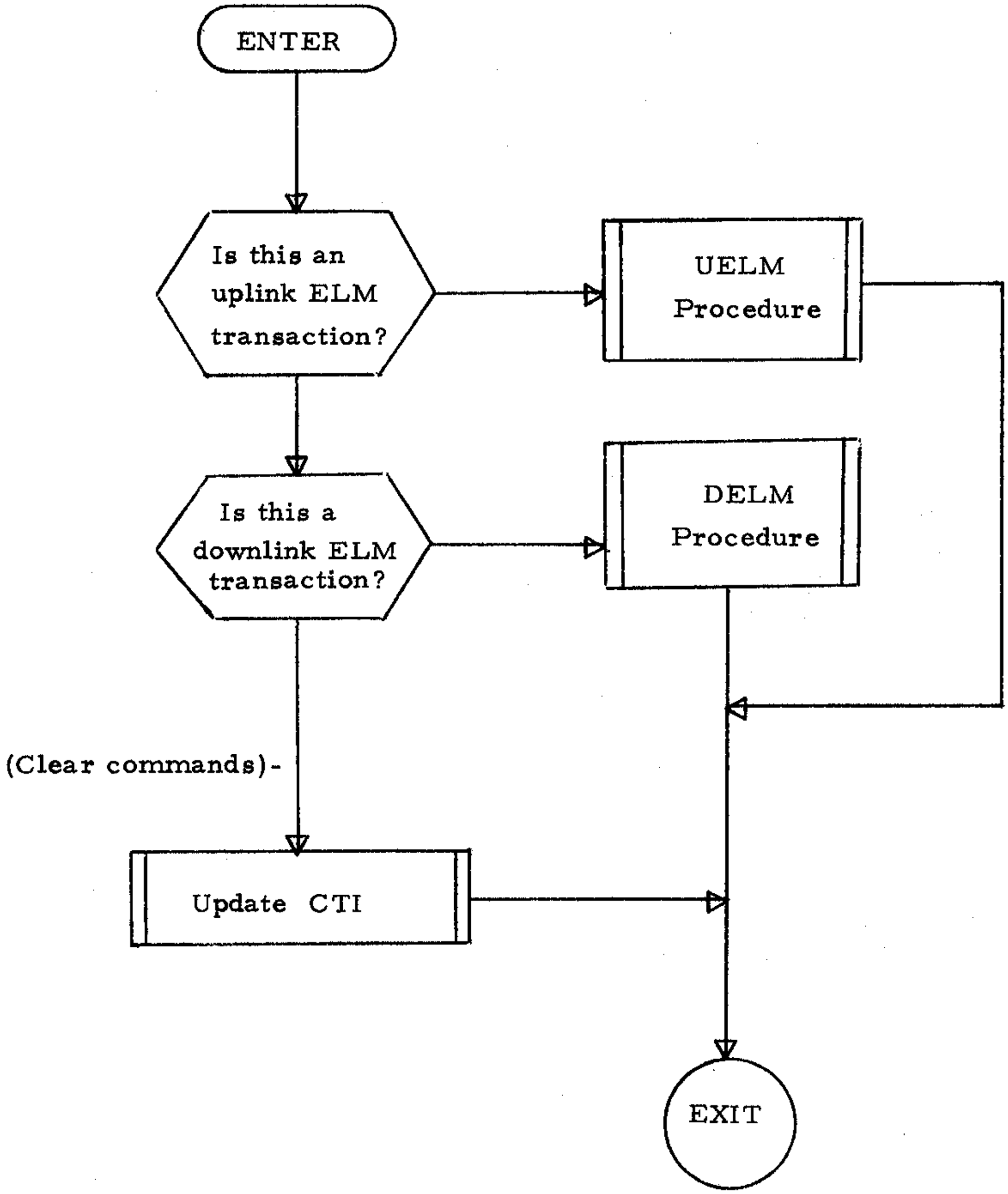


Figure 42. ELM Procedure

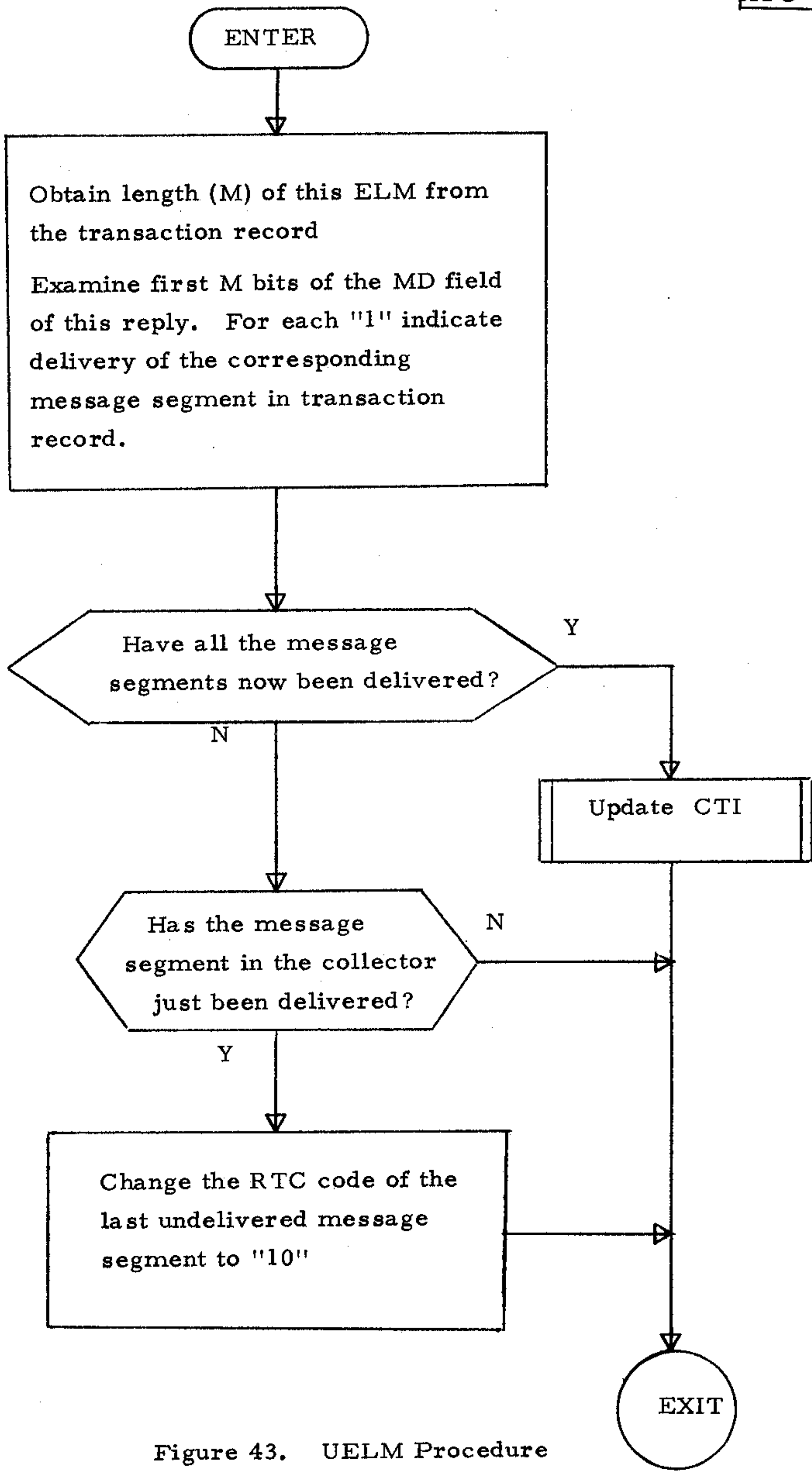


Figure 43. UELM Procedure

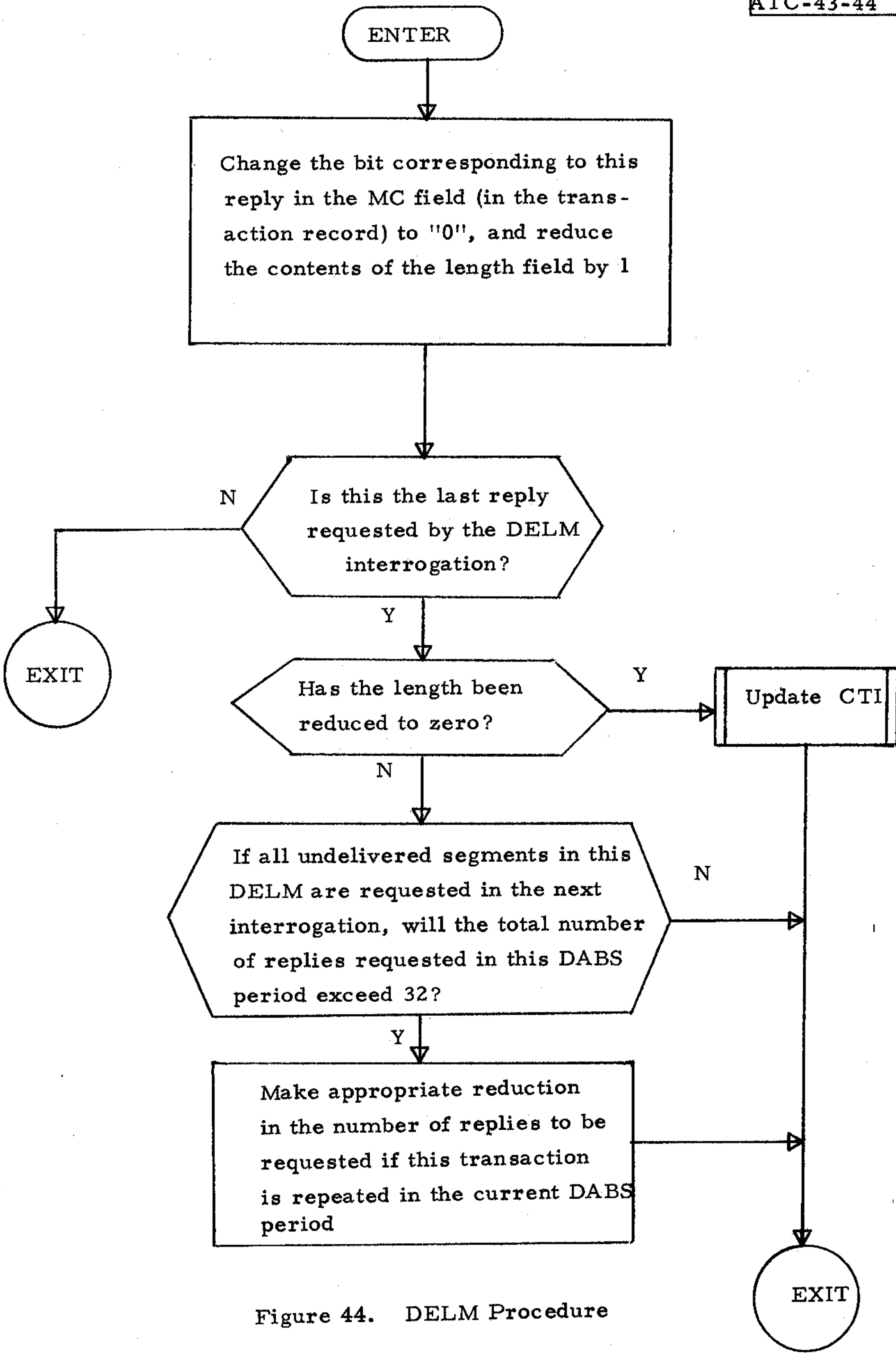


Figure 44. DELM Procedure

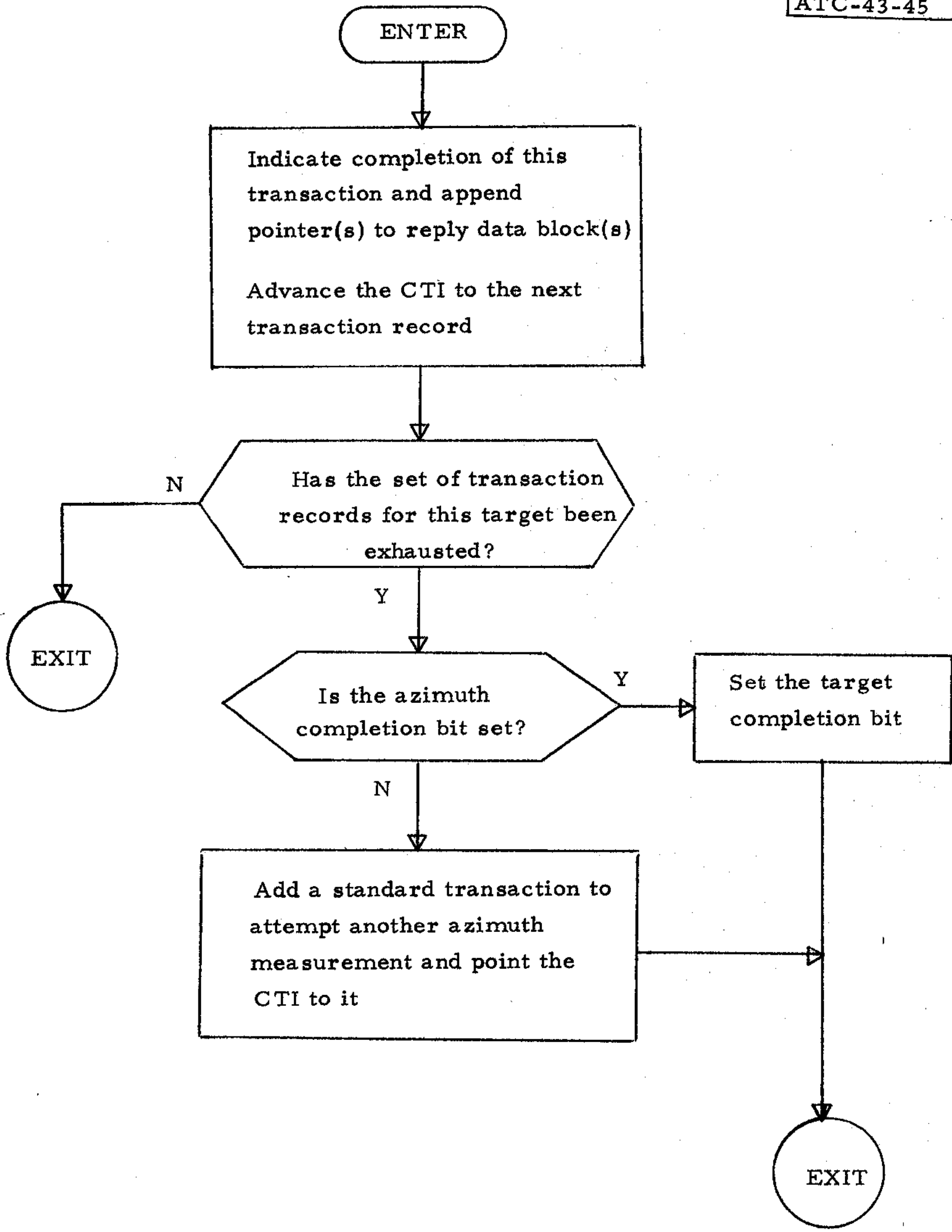


Figure 45. Current Transaction Indicator Update

REFERENCES

1. FAA-ER-240-26, "Discrete Address Beacon System (DABS) Sensor," 1 November 1974. (This document, the Engineering Requirement for the DABS sensor, is not generally available.)

2. FAA-RD-74-62, "Provisional Signal Formats for the Discrete Address Beacon System" (Revision 1), 25 April 1974 (Lincoln Laboratory Project Report ATC-30, Rev. 1).

3. "DABS Timing: Clocks, Synchronization and Restart," E. J. Kelly, Lincoln Laboratory Project Report ATC-29, 13 December 1974.

APPENDIX A

THE ISOLATION PROBLEM

Suppose that it is required to provide isolation between the replies of two targets which are included in a schedule cycle. Let the targets be at ranges ρ_1 and ρ_2 from the sensor, and let the distance between them be ρ_{12}. The first target emits a reply of duration R_1, beginning at time E_1, and the second starts its reply at time E_2. Then, if

$$E_2 - (E_1 + R_1) > \rho_{12} / c ,$$

the two replies will not overlap at any point in space or time, since the wavefront representing the leading edge of the second reply will always be contained within the expanding spherical shell bounding the first reply. If the aircraft are at the same azimuth and elevation, relative to the sensor, then

$$\rho_{12} = \left| \rho_1 - \rho_2 \right| .$$

In general, ρ_{12} exceeds the magnitude of the range difference, but the excess is limited by the nature of the antenna beam, since the targets must be simultaneously illuminated. As we shall see below, the scheduling algorithm automatically provides adequate isolation to satisfy the inequality

$$E_2 - (E_1 + R_1) > (1/c) \left| \rho_1 - \rho_2 \right| ,$$

hence it is only necessary to add an extra delay, δ, to the second interrogation to account for the "cross-range" component

$$\rho_{12} - | \rho_1 - \rho_2 | .$$

The worst-case geometry usually corresponds to two targets at close range, one at high elevation and one near the horizon, since any DABS antenna will have a fan beam. For example, a target at 45 degrees of elevation can have a range of nearly ten miles, and thus, be more the seven miles from another aircraft, also at range ten miles, on the sensor's horizon. The spacing is only exceeded in the azimuthal direction by a pair of very long range targets on opposite sides of a relatively wide beam. To handle all cases with a minimum of computation, a fixed minimum delay is introduced in the second interrogation whenever isolation is required. This delay is an adjustable system parameter whose nominal value is 25 μsec, corresponding to a spacing of about four miles. If the second transaction is unsynchronized, then this delay is added to the interrogation time which would otherwise be used. If the second transaction is synchronized, then a subepoch delay will already have been introduced, and any additional delay will have to be added in multiples of the subepoch value. Thus, the algorithm tests the subepoch delay, to see if it is already adequate for isolation, and then iteratively adds delay in units of the subepoch value, until the isolation delay parameter is exceeded.

In order to prove our assertion about the amount of isolation produced by the scheduling algorithm, we suppose that two consecutive reply listening period start times, as normally scheduled within a cycle, are L_1 and L_2 .

The first reply will actually arrive at some time, $L_1 + K_1$, so that K_1 will be the range correction for the first target. The duration of the first listening period will be $R_1 + G_1$, where R_1 is the reply length and G_1 is the range guard provided for the first target. This range guard will be larger than the largest anticipated value of K_1 by the amount, d, the departure from perfect range ordering allowed of the active target list. Since the scheduler normally places listening periods in contiguous positions, we have

$$L_2 = L_1 + R_1 + G_1 \quad .$$

Let the range delays for the two targets be D_1 and D_2. According to the approximate range ordering principle, we must have

$$D_1 \geq D_2 - d \quad .$$

The one-way propagation time for the first target is

$$\frac{\rho_1}{c} = \tfrac{1}{2} (D_1 + K_1 - \Delta_T) \quad ,$$

hence, the first reply was transmitted at time

$$E_1 = L_1 + \tfrac{1}{2} (K_1 - D_1 + \Delta_T)$$

If the second reply arrives at $L_2 + K_2$, then its emission time is

$$E_2 = L_2 + \tfrac{1}{2} (K_2 - D_2 + \Delta_T)$$

$$= L_1 + R_1 + G_1 + \tfrac{1}{2} (K_2 - D_2 + \Delta_T) \quad .$$

Now we can compute:

$$E_2 - (E_1 + R_1) = G_1 + \tfrac{1}{2} (K_2 - D_2 - K_1 + D_1) \quad .$$

The right-hand side of this equation can be rewritten in two ways. First, we write

$$G_1 + \tfrac{1}{2} (K_2 - D_2 - K_1 + D_1)$$

$$= G_1 - K_1 + K_2 + \tfrac{1}{2} (D_1 + K_1 - D_2 - K_2)$$

$$= G_1 - K_1 + K_2 + (1/c) (\rho_1 - \rho_2)$$

If $\rho_1 \geq \rho_2$ (the usual case), then we have

$$E_2 - (E_1 + R_1) = (1/c) \left| \rho_1 - \rho_2 \right| + G_1 - K_1 + K_2$$

$$\geq (1/c) \left| \rho_1 - \rho_2 \right|$$

since G_1 exceeds K_1 and K_2 is not negative.

It is possible to have $\rho_1 < \rho_2$ and still have approximately decreasing ordering of range delay, since the uncertainty in range for the second target could be large, producing a small value of range delay. In this case

$$\left| \rho_1 - \rho_2 \right| = \rho_2 - \rho_1 \quad ,$$

and we write

$$G_1 + \tfrac{1}{2} (K_2 - D_2 - K_1 + D_1)$$

$$= G_1 + D_1 - D_2 + \tfrac{1}{2} (D_2 + K_2 - D_1 - K_1)$$

$$= G_1 + D_1 - D_2 + (1/c) (\rho_2 - \rho_1)$$

$$= G_1 + D_1 - D_2 + (1/c) \left| \rho_1 - \rho_2 \right|$$

Now G_1 exceeds d, and

$$D_1 - D_2 \geq - d \quad ,$$

hence

$$E_2 - (E_1 + R_1) \geq (1/c) \left| \rho_1 - \rho_2 \right|$$

as before.

☆U. S. GOVERNMENT PRINTING OFFICE: 1975--600-018--43