COMMUNICATIONS PROCESSOR LOADING STUDIES

MCC-H CCATS

UNIVAC 494 Algorithm

April 10, 1967

Prepared for
National Aeronautics and Space Administration
on behalf of Bellcomm, Inc.

by

Bell Telephone Laboratories, Inc.

BELL TELEPHONE LABORATORIES
INCORPORATED

SUBJECT: Communications Processor - Loading Studies - CCATS at MCC-H - UNIVAC 494 Algorithm - Case 20063

DATE: April 10, 1967

FROM: R. M. Marella

MF7-4262-3

ABSTRACT

This is one of a series of memoranda giving supporting information on the Communications Processor loading studies for the MCC-H installation operating under CCATS. This memorandum describes the algorithmic model of the Communications Processor system employing UNIVAC 494 computers. The algorithm will be used to determine the capabilities and limitations of the Communications Processor in an Apollo mission environment.

The algorithm structure embodies the logic necessary to define the operation of the many synchronous and asynchronous functions performed by the multi-asynchronous CCATS hardware/software configuration. The structure has been designed to be independent of changes in data format, line speed, priority assignment, program operation, etc. Such variables define the total system environment and are among the required input data for the computation of system capabilities.

The algorithm may be used to determine the Communications Processor's performance with respect to character transfer, processing (with special emphasis on interrupt processing), and storage.

BELL TELEPHONE LABORATORIES
INCORPORATED

SUBJECT: Communications Processor - Loading Studies - CCATS at MCC-H - UNIVAC 494 Algorithm - Case 20063

DATE: April 10, 1967

FROM: R. M. Marella

MF7-4262-3

MEMORANDUM FOR FILE*

1.0 Introduction

The UNIVAC 494 Algorithm is an algorithmic model of the UNIVAC 494 Communications Processor (CP) system used in the Communications, Command, and Telemetry System (CCATS) at MCC-H. The algorithm will be used to determine the capabilities and limitations of the UNIVAC 494 Communications Processor in an Apollo mission environment.

The algorithm structure embodies the logic necessary to define the operation of the many synchronous and asynchronous functions performed by the multi-asynchronous CCATS hardware/software configuration. The structure has been designed to be independent of changes in data format, line speed, priority assignment, program operation, etc. Such variables define the total system environment and are among the required input data for the computation of the CP system capabilities.

---

*This document supersedes MF6-4332-21, August 9, 1966.

The algorithm may be used to determine the communications processor's performance with respect to character transfer, processing (with special emphasis on interrupt processing), and storage.

2.0 The UNIVAC 494 Algorithm

The UNIVAC 494 Algorithm computes the required parameter values which are necessary to define the operation of the communications processor under a given CCATS loading. To produce accurate results, the total hardware/software environment of the system must be specified as an input to the structure. The specification of the hardware environment (line speed, channel assignment, data frame length, etc.) may be accomplished rather easily. The specification of the software environment is somewhat more difficult.

A true specification of the software environment requires that we know every computer instruction traversed by the program, together with the value of relevant biasing parameters (e.g., the contents of each control register) and the data necessary to choose paths at decision points. This task has been simplified by defining irreducible program segments whose content, length, nominal execution time, etc., are specified both by program coding and our *a priori* knowledge of the system environment (e.g., we "know" that a particular data frame contains a telemetry summary message by

virtue of having specified this fact in our input data). If we are unable to use the actual program coding, we may specify the information flow in terms of irreducible segments composed of "average instructions," whose duration, priority, and decision points have been estimated from the required processing task.

The algorithm is shown in Appendix A, Figure A1 through A20.* Figure 1 is a block diagram of the algorithm, divided into six sections: Initialization, I/O Hardware Model, Processing Model, Channel Operation Model, Peripheral Device Models, and Decision Time Model. These are discussed below. Note that the operational flow through the algorithm is left to right and top to bottom. The results produced by the algorithm pertain to character transfer, processing, and storage. Events are identified in space (Which channel and device caused the event?), time (When did the event occur?) and context (What data frame and processing segment is associated with the event?). The algorithm computes the time of occurrence of system response to input stimuli (data requests and interrupts) and to internal stimuli (processing events), identifying both significant time parameters and the portion

---

* The index for Appendix A appears on page 21.

of the system resource which has been used or is in use at a point in time. From the time performance of the system, the delay and/or loss incurred by each data character and frame due to transfer, processing, or storage can be determined.

2.1 Initialization

The initialization of the algorithm sets values for all input data of the desired system environment as well as all internal variables of the model so that proper initial conditions exist.

2.2 Input/Output Hardware Model

Within the I/O section of the UNIVAC 494, there exist separate priority schemes for the selection of interrupts and data transfer requests. Separate scan cycles are used to evaluate these priority lists. The I/O Hardware Model contains the logic to specify the interrupt scan and data scan operations, as well as the choice of the next action performed by the computer. The overall priority for the actual computer action is as follows:

(1) if a data transfer request has been selected for service by a data scan, perform this transfer next;

(2) if no data transfer requests have been selected for service by a data scan, if an interrupt has been selected for service by an interrupt scan, and if interrupts are allowed to occur, interrupt the

program being executed and take the next instruction from the appropriate interrupt memory location;

(3) if no data transfer requests or allowable interrupts have been selected for service, then continue with the program being executed and perform the next subinstruction in the instruction stream.

The algorithm computes the time values for the initiation of each data and interrupt scan. At the appropriate scan time, the algorithm logic is exercised to perform the scan operation and generate information as to the result (the identity and type of request detected by the scan, if any). If a data transfer request exists and was detected by the data scan, the time values for completion of the data transfer and initiation of the next data scan are generated. If no data transfer request exists or none was detected by the data scan, the time value for the initiation of the next program subinstruction is generated and the processing Model selects the next subinstruction to be executed. If an interrupt exists and was detected by the interrupt scan, the Processing Model replaces its current processing chain with that pertaining to the interrupt. No further interrupt scan operation will occur until the interrupt lockout is released by the Processing Model at some future point in time.

2.3 Processing Model

The Processing Model breaks the processing performed by the UNIVAC 494 into predefined processing entities called α segments. For ease of computation, each α segment is composed of a series of β segments and the number of times each β segment is to be executed. The β segment is a list of subinstructions defining an instruction sequence. Each subinstruction has associated with it a function tag which defines any action required by the algorithm.

The α segment is a list of β segments and the number of times each is to be executed. Each α segment operates at a single priority level and is a predetermined processing entity with all branching removed. The priority levels used are identical with those used within the UNIVAC 494. (One additional level has been defined for operation with a hardware interrupt lockout). To define the processing flow path for data frames and α segments, an A segment is used. The A segment is a list of α segments and the additional variables necessary to produce the correct flow path through them. There are two types of A segments, worker and executive/switcher. (The type of A segment used to model UNIVAC 494 processing is not correlated with the names worker and executive/switcher. In general, a "worker" program would use a worker A segment; an executive program would use

either a worker A segment or an executive/switcher A segment; a switcher program would use an executive/switcher A segment.

Each worker A segment specifies the chain of processing to be executed for a specific type of data frame. (The processing chain is initiated by predefining the A segment used for each input data frame and associating the name of the A segment with the data frame). Throughout the processing, each α segment must have associated with it the data frame information defining the particular data frame being processed.

The processing chain defined by a worker A segment recognizes three function tags within β segments which change the information flow path. These are a jump, a return jump, and an executive return. The jump causes the algorithm to replace the current α segment and A segment with those specfied in the worker A segment. The return jump causes the algorithm to replace the current α segment and A segment with those specified in the worker A segment, returning to the current α segment at the completion of the new one. An executive return causes the algorithm to replace its current A segment with an executive/switcher A segment. At some later time the executive/switcher A segment returns to the worker A segment initiating it (which is now its predecessor).

Each executive/switcher A segment defines the flow path to be taken in the case of a yes or a no answer to a specific directed question. The new flow path information contains the identity of the question to be asked in proceeding to the next entry in the executive/switcher A segment. The questions which have been defined are queue related. Processing queue entries contain information defining the α segment to be executed, the α segment which is its predecessor in the processing flow path, and the data frame being processed. Queues have been gathered into queue groups. Queue groups have been gathered into queue super groups. Each has an entry counter and a busy indicator. The directed questions can determine if the queue (queue group, queue super group) exists, has entries, or is busy. (The busy indicator is used to prevent immediate processing when an executive return requests action by a busy I/O handler.) One additional directed question determines the suspendability of the predecessor α segment.

Additional function tags are defined to perform the required I/O actions (such as reinitiating an input channel after a monitor interrupt), as well as housekeeping operations required only by the algorithm.

2.4 Channel Operation Model

The UNIVAC 494 has two types of I/O channels, normal (fast) and compatible (slow). The maximum normal channel

transfer rate is identical to the maximum UNIVAC 494 transfer rate. The maximum compatible channel transfer rate is less than the computer rate, compatible channels being used with older, slower peripheral devices. Channels are grouped according to type, with each type of channel group requiring a different model.

Each data transfer alters the basic priority structure of the succeeding data scan. (This is required by timing considerations and the desire to prevent seizure of I/O capability by either input or output.) The Channel Operation Model specifies a mask time for each I/O channel group (i.e., a data request will not be honored until the mask time for its channel group has been exceeded). This mask time effectively alters the priority structure of the succeeding data scans, corresponding to the lockout time imposed by the I/O section of the computer. Each input and output channel may be individually inactivated by a monitor interrupt in the Peripheral Device Model and reactivated by the Processing Model.

2.5 Peripheral Device Models

The peripherals which have been modeled include three communications peripherals, four standard peripherals, and two clocks. Each peripheral consists of a control unit and one or more devices. Where the computer is able to

converse with a single device for a full data frame, an Internally Specified Index (ISI) mode is used with 30-bit transfers. Where the computer must interleave characters from several devices, an Externally Specified Index (ESI) mode is used with 15-bit transfers (the remainder of the 30-bit computer word is used to identify the device). Where the control unit operates with a number of devices (communications peripherals), the control unit selects a device for service according to a set of priority rules. While a large number of devices may desire to converse with the computer, only a single request is presented by the control unit to the computer I/O channel.

Each peripheral model, upon completion of a data transfer operation, calculates the portion of transfer capability used and generates the next data transfer request. If the end of a data frame is reached, the appropriate interrupt is generated. New data frames are selected from the data frame queue maintained for each peripheral device. The selection of a request for service by the control unit is performed at a later time by the portion of the model which incorporates the priority scheme of the control unit.

2.5.1 Communications Peripherals

The communications peripherals are as follows:

1. a Standard Communication Subsystem (SCS) Channel, consisting of a Multiplexer control unit and

converse with a single device for a full data frame, an Internally Specified Index (ISI) mode is used with 30-bit transfers. Where the computer must interleave characters from several devices, an Externally Specified Index (ESI) mode is used with 15-bit transfers (the remainder of the 30-bit computer word is used to identify the device). Where the control unit operates with a number of devices (communications peripherals), the control unit selects a device for service according to a set of priority rules. While a large number of devices may desire to converse with the computer, only a single request is presented by the control unit to the computer I/O channel.

Each peripheral model, upon completion of a data transfer operation, calculates the portion of transfer capability used and generates the next data transfer request. If the end of a data frame is reached, the appropriate interrupt is generated. New data frames are selected from the data frame queue maintained for each peripheral device. The selection of a request for service by the control unit is performed at a later time by the portion of the model which incorporates the priority scheme of the control unit.

2.5.1 Communications Peripherals

The communications peripherals are as follows:

1. a Standard Communication Subsystem (SCS) Channel, consisting of a Multiplexer control unit and

Communication Line Terminal (CLT) peripheral devices operating in ESI mode;

2. a Scanner Selector (S/S) Channel, consisting of a Scanner Selector control unit and 494/7094 adapter peripheral devices operating in ESI mode;
3. a Polynomial Buffer Terminal (PBT) Channel, consisting of a Scanner Selector control unit and Polynomial Buffer Terminal peripheral devices operating in ISI mode.

2.5.1.1 SCS Channel

The SCS Channel control unit uses a matrix selection scheme, examining all service requests simultaneously (data transfer or interrupt) and performing its selection by strict numerical priority. The CLT may be either input or output, Low Speed (LS), High Speed (HS), or Wide Band (WB).

LS CLT's have a single character assembly register and operate with asynchronous teletype equipment.

HS CLT's have both a single character assembly register and a queuing register. They operate with synchronous data sets in the 2.4 kb/sec. range, with framing provided by a data set control line.

WB CLT's are identical with HS CLT's, except for operation at 40.8 kb/sec. Framing is provided by recognition of sync characters for all data frames. At the completion

of a data frame, the central processor's worker program must send an external function command to the CLT to allow sync character recognition.

Note that input is an uncontrolled interface, while output is controlled by an external function command to the CLT to request output data.

2.5.1.2 S/S Channel

The S/S Channel control unit operates in a round-robin polling fashion. Each 494/7094 adapter is a full duplex device with an internal request selection priority scheme. The 494/7094 adapter operates with a controlled 40.8 kb/sec. facility. Failure to service a data transfer request within its specified limit causes delay rather than data loss, since a gap is inserted in the bit stream by the control line.

Note that input is a controlled interface, since the central processor must activate a Ready-To-Receive (RTR) control line before input will commence. However, failure to activate the RTR line within 10 ms will cause the RTCC to throw data away. Output is controlled as in the SCS Channel.

2.5.1.3 PBT Channel

The PBT Channel control unit is similar to the S/S Channel control unit in its polling operation. However, once a PBT has been selected, an entire block of data is transferred. Each PBT contains two block (600-bit) registers for

each transmission direction. These buffers are emptied or filled by twenty successive 30-bit transfers plus an interrupt.

Note that input is an uncontrolled interface, while output is controlled as in the SCS Channel.

2.5.2 Standard Peripherals

The standard peripherals are as follows:

1. an FH880 Magnetic Drum Subsystem (FH880) Channel, consisting of a control unit and an FH880 drum operating in an ISI mode;
2. a IIIC Magnetic Tape Subsystem (IIIC) Channel, consisting of a control unit and several IIIC tape units operating in an ISI mode.
3. a 1004 Card Processor Subsystem (1004) Channel, consisting of an interface adapter and a 1004 card processor operating in an ISI mode;
4. a 494 Operator's Console Subsystem (CONSOLE) Channel, consisting of a 494 operator's console operating in an ISI mode.

2.5.2.1 FH880 Channel

The FH880 Channel control unit performs buffering, character assembly/disassembly, and control functions for FH880 drums. Both input and output are controlled by the central processor. Note that failure to service a data transfer request within its specified limit causes a drum skip and a 34 ms gap is inserted in the data transfer request stream.

2.5.2.2 IIIC Channel

The IIIC Channel control unit performs buffering, character assembly/disassembly, and control functions for IIIC tape units. Both input and output are controlled by the central processor. Note that failure to service a data transfer request within its specified limit causes an error interrupt and an attempt to read or write the data frame begins again after a large delay to backspace and erase the tape.

2.5.2.3 1004 Channel

The 1004 Channel interface adapter performs electrical interface functions for the 1004 card processor. Both input and output are controlled by the central processor. Note that this simplified model does not include any external function command other than input or output and that the interval between data requests if fixed.

2.5.2.4 CONSOLE Channel

Both input and output are controlled by the central processor (input by the interlock of the console keyboard). Note that this simplified model does not include any provision for keyboard initiated external interrupts and that the interval between data requests is fixed.

2.5.3 Clocks

The clocks are as follows:

1. a Real Time Clock (RCLOK) Channel, consisting of the real time clock located within the central processor;
2. a Day Clock (DCLOK) Channel, consisting of the day clock located within the central processor and the operator's console.

Note that these clocks are attached to dummy channels for computational purposes.

2.6 Decision Time Model

This portion of the algorithm specifies the choice of the action to be performed by the model, by determining which event will occur next. Simulated time is increased at this point by a variable time increment, so that simulated time will then be equal to the time at which the next event occurs in the real world.

3.0 A Sample Data Frame

To clarify the operation of the algorithm, let us postulate a sample data frame. SAMPLE is a ten character data frame occurring once a second. It appears on a ten bit HS CLT operating at 1 kb/sec. It is outputted to the RTCC as a ten character data frame on a ten bit 494/7094 adapter operating at 40.8 kb/sec. SAMPLE is to be processed by the

A segment called WORKER, utilizing the buffer group called TEN.

The input data frame queue contains SAMPLE, with its first data request occurring at T = 0. At some later time (T = .001 sec.), the multiplexer within the SCS Channel Model selects this data request and presents it to the computer 13 μs later. A later data scan within the I/O Hardware Model detects the data request, calculating the time at which the request is completed (T = .0015 sec.). The SCS Channel Model computes that the request required sixteen per cent of the character availability time (.009 sec.) to be completed and that the second data request will occur at T = .010 sec. Because this is the first input data request for SAMPLE, the buffer histogram for TEN is increased by one. The tenth data request is completed at T = .094 sec. Because this is the last character in the data frame, an external interrupt is presented to the multiplexer and the first data request of the succeeding data frame set to occur at T = 1 sec. After selection by a multiplexer data request evaluation, the external interrupt is presented to the computer 13 μs later. A later interrupt scan within the I/O Hardware Model detects the interrupt.

The processing being executed is examined by the Processing Model for suspendability and suspended. An α

segment in WORKER is selected, performing the interrupt recording and suspension operations. The next α segment selected from WORKER performs the interrupt analysis operation. The third α segment selected from WORKER is intended to perform the outputting of SAMPLE to the RTCC. However, during its execution an interrupt occurs and the α segment being processed is suspended. At some later time, the executive/switcher A segment called SWITCH selects the suspended WORKER for execution. Near the end of WORKER, an executive return requests the output of SAMPLE to the RTCC. SAMPLE is queued on the output data frame queue.

At some later time, the worker segment called RTCCOUT sends an external function command to the 494/7094 adapter within the S/S Channel Model, generating an output data request at T = .5 sec. The S/S later polls this 494/7094 adapter and selects the output data request, presenting it to the computer. A later data scan detects the data request, calculating the time at which the request is completed. The S/S Channel Model computes the character availability time used and the time at which the next data request occurs. After the completion of the eleventh data request, an output monitor interrupt is generated. Because this is the last output data request for SAMPLE, the buffer histogram for TEN is decreased by one. After detection of

the interrupt in the I/O Hardware Model and the interrupt recording and analysis operations in the Processing Model, the worker A segment called RTCCOUTMON returns to WORKER. At the completion of WORKER, an executive return signifies the end of the processing chain by returning to SWITCH, which then examines the existing queues to determine what a segment the Processing Model should execute.

4.0 Summary

The capabilities and limitations of the Communications Processor in an Apollo mission environment will be determined by the use of the UNIVAC 494 Algorithm. It may be used to determine the character transfer, processing and storage performance of the system.

The UNIVAC 494 Algorithm is designed as a flexible analysis tool for the evaluation of system performance. New peripheral device models may be added by examining their operation with respect to the nine peripherals modeled in the algorithm. New function tags may be added, reflecting additional system performance measurement points. Estimates for proposed new functions may be inserted to determine system capabilities with the addition of a new processing task. However, increased flexibility is only bought with added complexity. Each proposed change to the algorithm and its input data (especially processing entities) should be carefully examined to insure proper algorithm operation.

The UNIVAC 494 algorithm may be visualized as an array of variables, identified by both channel (j) and device (i). The variables consist of indicators, event occurrence times (T_{xx}), and incremental times (ΔT_{xx}). Data frame queues for the system traffic loading are maintained for every channel and device.

The algorithm derives the value of these variables, computing new indicators and event times as events occur in the simulated time stream. Events are chosen for servicing by the algorithm according to the time-dependent hardware/software priority and logic of the CCATS environment. The results obtained consist of event service times under a CCATS loading. The resource measured is time. Summation of the delays incurred in servicing events will measure incremental loading delay. The ratio of time consumed to time available for certain significant events will measure potential data loss.

R. M. Marella/cas

WH-4262-RMM-RP R. M. MARELLA

Attachments
Figure 1
Appendix A

Copy (with attachments) to
Messrs. J. J. Hibbert - Bellcomm
J. Z. Menard - Bellcomm (10)

P. V. Dimock - WH
Mrs. S. E. Miller - WH
Messrs. A. E. Peterson - WH
C. W. Schramm - WH
M. P. Wilson - WH

Initialization	I/O Hardware Model
Input Configuration Data: Hardware (Transfer Data), Software (Processing Data), Loading (Traffic Data).	Perform Interrupt Scan
	Perform Data Scan.
Initialize the algorithm with input data and preset variable values.	Choose and perform the next c action: 1) Data Transfer, 2) Program Subinstructi

	Processing Model	Channel Operation Model
.	If an interrupt has been selected by the interrupt scan, set up interrupt processing. (If the current program is suspendable, suspend it and proceed to interrupt analysis processing. If the current program is nonsuspendable, perform interrupt recording processing and return to the nonsuspendable program.) Select a program subinstruction for execution and update the processing housekeeping variables. Check the function tag of the program subinstruction and perform any special operations required.	Mask the channels.
omputer on.		

Block Diagram
UNIVAC 494 Algorithm
FIGURE 1

Peripheral Device Models		Decision Time Model
SCS: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	Choose the next decision time of the algorithm.
S/S: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
PBT: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
FH880: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
III C: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
RCLOK: Compute CAT. Set the next data request.	Set an interrupt.	
DCLOK: Compute CAT. Set the next data request.	Set an interrupt.	
1004: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
CONSOLE: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	

APPENDIX A

UNIVAC 494 Algorithm

Figure Number	Number of Pages	Subject
A1	47	Definitions
A2	1	Block Diagram
A3	1	Initialization
A4	2	Interrupt Scan
A5	3	Data Scan
A6	1	I/O Lockout
A7	36	Processing Housekeeping
A8	1	I/O Model Selection
A9	7	SCS Channel
A10	7	S/S Channel
A11	7	PBT Channel
A12	9	FH880 Channel
A13	9	IIIC Channel
A14	1	RCLOK
A15	1	DCLOK
A16	7	1004 Channel
A17	7	CONSOLE Channel
A18	2	Time Housekeeping
A19	1	Results Computation
A20	12	Error Housekeeping

T ≡ A running time variable used to calculate decision times.

T_{START} ≡ The running time value at which computation begins.

T_{STOP} ≡ The running time value at which computation ceases and results are determined.

j ≡ An index used to denote the I/O channel. Note that dummy channels are assigned to the Day Clock and Real Time Clock for computational purposes.

g ≡ An index used to denote the I/O channel group.

i ≡ An index used to denote the device connected to the I/O channel. Note that where a single physical device is full duplex, two indices are assigned to it, one for input and one for output. Note also that where there is but one device connected to the channel, the index is not required, but is retained as a dummy.

k ≡ An index used to denote the priority level of the program segment. There are five levels for computational purposes: INT,0,1,2,3. Level INT locks out I/O interrupts and is both nonsuspendable and noninterruptable. Level 0 is nonsuspendable and interruptable. Level 1, Level 2, and Level 3 are each suspendable and interruptable by Level INT and also by Level 0 in certain cases.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-1

h ≡ An index used to denote the type of program to which the housekeeping information pertains. There are four types defined by the algorithm: w, e, s, p. Type w is a worker program, type e is an executive program (excluding the switcher), type s is a switcher program, and type p is a predecessor program which may itself be of type w, e, or s.

d ≡ An index used to denote the identity of data frame information. There are three types defined by the algorithm: i, o, s. Type i is an input data frame, type o is an output data frame, and type s is a source data frame.

$T_{DS} \equiv T_{\text{Data Scan}}$ = The time at which a data scan is initiated.

$T_{LDS} \equiv T_{\text{Last Data Scan}}$ = The time at which the last data scan in the current subinstruction cycle is initiated.

DI ≡ Data Indicator = A data indicator which is set by the successful detection of a data request during a data scan. It is reset by the initiation of the data transfer.

$T_{DTA} \equiv T_{\text{Data Transfer Available}}$ = The time at which the data transfer requested by a successful data scan is initiated.

$T_{MF} \equiv T_{\text{Memory Free}}$ = The time at which memory is free during the current subinstruction cycle.

$T_{DRC} \equiv T_{\text{Data Request Complete}}$ = The time at which the data transfer has been completed with respect to the peripheral device.

$T_{SE} \equiv T_{\text{Subinstruction Execution}}$ = The time at which the next subinstruction cycle is initiated.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-2

IOS ≡ Input/Output Sequence = A data indicator which is set by the initiation of a data transfer and which prevents interrupt scans during the data transfer. It is reset by the selection of a subinstruction cycle for execution after an unsuccessful data scan.

DATAJ ≡ The value of j selected by a successful data scan.

DATAI ≡ The value of i selected by a successful data scan.

REQ ≡ Request = The type of data transfer operation requested of a peripheral device model by a successful data scan. The three types are: DATA, FN13, FN17. DATA signifies the completion of a data request generated by the peripheral device and causes it to generate the time at which the next request will occur. FN13 signifies a command to perform some action. Unless otherwise specified, the command is assumed to mean generate input data to the computer or accept output data from the computer. FN17 signifies the completion of an external interrupt request generated by the peripheral device at the end of a data frame.

DR_{FN17} ≡ $(\text{Data Request})_{FN17}$ = A data request indicator which is set by a SC function tag, so that the request may be recognized by the next data scan.

J_{FN17} ≡ The value of j corresponding to DR_{FN17}.

I_{FN17} ≡ The value of i corresponding to DR_{FN17}.

DR_{FN13} ≡ $(\text{Data Request})_{FN13}$ = A data request indicator which is set by an EF function tag, so that the request may be recognized by the next data scan.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-3

J_{FN13} ≡ The value of j corresponding to DR_{FN13}.

I_{FN13} ≡ The value of i corresponding to DR_{FN13}.

$TYPE_{FN13}$ ≡ An indicator used to determine when a command other than one to input or output data is sent to a peripheral device. The only command so used is the terminate command sent to the FH880.

$LATENCY_{FN13}$ ≡ An indicator used to determine the delay inserted in fulfilling a read or write command by an addressable peripheral device (FH880, IIIC). A minimum, average, or maximum latency delay may be inserted. If a specific address is furnished, the actual delay is computed and inserted.

$ADDRESS_{FN13}$ ≡ A variable used to specify the starting address for an input or output by an addressable peripheral device.

$ADDRESS_j$ ≡ A variable specifying the address for the peripheral device on the j^{th} channel (FH880, IIIC).

$CHAN_j$ ≡ $(Channel)_j$ = A variable specifying whether a channel is normal or compatible. A normal computer channel operates at the maximum transfer rate of the internal computer hardware (1.82 μs per transfer). A compatible channel operates at a transfer rate less than the maximum to accomodate slower peripheral devices (3.0 μs per input transfer; 4.8 μs per output transfer).

AO_j ≡ $(Active\ Output)_j$ = An indicator used to inhibit output data requests on the j^{th} channel after an Output Monitor Interrupt.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-4

$AI_j \equiv (\text{Active Input})_j$ = An indicator used to inhibit input data requests on the j^{th} channel after an Input Monitor Interrupt.

$T_{AO_j} \equiv (T_{\text{Active Output}})_j$ = The time at which output data requests are allowed on the j^{th} channel.

$T_{AI_j} \equiv (T_{\text{Active Input}})_j$ = The time at which input data requests are allowed on the j^{th} channel.

$\Delta T_{AO_j} \equiv (\Delta T_{\text{Active Output}})_j$ = The time required to set AO_j after the RC function tag is detected.

$\Delta T_{AI_j} \equiv (\Delta T_{\text{Active Input}})_j$ = The time required to set AI_j after the RC function tag is detected.

$T_{CMO_g} \equiv (T_{\text{Channel Mask Output}})_g$ = The time at which the I/O lockout allows an output channel within the g^{th} channel group to recognize a data request.

$T_{CMI_g} \equiv (T_{\text{Channel Mask Input}})_g$ = The time at which the I/O lockout allows an input channel within the g^{th} channel group to recognize a data request.

$\Delta T_{CMO_g} \equiv (\Delta T_{\text{Channel Mask Output}})_g$ = The output lockout mask time for the g^{th} channel group, measured from T_{DS} of the data transfer. This is equal to 3 μs for a compatible channel group; 0 for a normal channel group.

$\Delta T_{CMI_g} \equiv (\Delta T_{\text{Channel Mask Input}})_g$ = The input lockout mask time for the g^{th} channel group, measured from T_{DS} of the data transfer. This is equal to 1.5 μs for a compatible channel group; 0 for a normal channel group.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-5

$T_{DR_j} \equiv (T_{\text{Data Request}})_j$ = The time at which a data request on the j^{th} channel is presented to the I/O hardware.

$DR_j \equiv (\text{Data Request})_j$ = A data indicator which is set if an unmasked data request exists on the j^{th} channel (either input or output) and the channel is active. It is reset by the initiation of the data transfer or by a data scan which finds the request masked or the channel inactive.

$T_{DRE_j} \equiv (T_{\text{Data Request Evaluation}})_j$ = The time at which the j^{th} channel control unit examines its devices to present a request to the computer. If a control unit is totally under the control of the computer (FH880, IIIC, 1004, CONSOLE), then this variable is used to find the next data frame when interrupts are inhibited.

$\Delta T_{DRE_j} \equiv (\Delta T_{\text{Data Request Evaluation}})_j$ = The time required by a control unit to perform a data request evaluation. Note that SCS channel evaluations are made by first examining requests, then connecting a requesting device to the I/O channel. S/S and PBT channel evaluations are made by first connecting the next round-robin polling station, then evaluating the request.

For an SCS channel, this is 13 μsec.

For a S/S channel, this is 2 μsec (polling time).

For a PBT channel, this is 1.6 μsec (polling time).

For an FH880, IIIC, 1004, or CONSOLE channel, this is an arbitrary time used between the completion of the last data request and the initiation of the succeeding data frame (in the case where interrupts are inhibited).

Definitions
UNIVAC 494 Algorithm
FIGURE A1-6

$\Delta T_{CL_j} \equiv (\Delta T_{\text{Control Line}})_j$ = The control line time for the j^{th} channel, determined by signal rise times and cable length. This is equal to 1.5 μs for all channels.

$\Delta T_{D_j} \equiv (\Delta T_{\text{Downtime}})_j$ = The downtime required by the j^{th} channel control unit between successive data requests. (S/S and PBT.) Note that the minimum value for this quantity is determined by the U494 I/O hardware and is 200 ns. Note also that this time is absorbed into ΔT_{DRE_j} for SCS channels, into $\Delta T_{CU_{ji}}$ for FH880 and IIC channels, and into $\Delta T_{DR_{ji}}$ for 1004 and CONSOLE channels. For an S/S or PBT channel, this is 1 μs.

$TYPE_j \equiv$ A variable used to identify the type of peripheral device control unit connected to the j^{th} channel. The nine types are: SCS, S/S, PBT, FH880, IIIC, RCLOK, DCLOK, 1004, CONSOLE.

The SCS (Standard Communication System Channel) is a channel to which is attached a Communication Multiplexer and a number of Communication Line Terminals (CLTs).

The S/S (Scanner Selector Channel) is a channel to which is attached a Scanner Selector and CCU 494/7094 adapters. The Scanner Selector operates in a round-robin polling sequence.

The PBT (Polynomial Buffer Terminal Channel) is a channel to which is attached a Scanner Selector and Univac 33/22 Polynomial Buffer Terminals. The Scanner Selector operates in a round-robin polling sequence.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-7

The FH880 (FH880 Magnetic Drum Subsystem Channel) is a channel to which is attached a buffered FH880 Channel Synchronizer, an FH880 Drum Control Unit, and one FH880 Drum operating at an interlace of one.

The IIIC (IIIC Magnetic Tape Subsystem Channel) is a channel to which is attached a buffered IIIC Channel Synchronizer, a IIIC Tape Control Unit, and several IIIC Magnetic Tape Units operating at a speed of 556 frames/inch. Note that IVC Magnetic Tape Units are identical with the IIIC Magnetic Tape Units, except for the ability to operate at 800 frames/inch.

The RCLOK (Real Time Clock) is a dummy channel used to create the data requests and interrupts of the Real Time Clock located within the UNIVAC 494 hardware.

The DLOK (Day Clock) is a dummy channel used to create the data requests and interrupts of the Day Clock located within the UNIVAC 494 hardware and the UNIVAC 494 Operator's Console.

The 1004 (1004 Card Processor Subsystem Channel) is a channel to which is attached a Universal Interface Adapter, and a 1004 Card Processor.

The CONSOLE (494 Operator's Console Subsystem Channel) is a channel to which is attached a 494 Operator's Console.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-8

$T_{IS} \equiv T_{\text{Interrupt Scan}}$ = The time at which an interrupt scan is initiated.

II ≡ Interrupt Indicator = An interrupt indicator which is set by the successful detection of an interrupt during an interrupt scan. It is reset by the execution of the first succeeding attempt to select a subinstruction cycle.

INTJ ≡ The value of j selected by a successful interrupt scan.

INTI ≡ The value of i selected by a successful interrupt scan.

INT ≡ Interrupt = The type of interrupt selected by a successful interrupt scan. The seven types are: EXRN, INEXT, OUTEXT, INMON, OUTMON, RCLOK, DCLOK.

EXRNINT ≡ Executive Return Interrupt = An interrupt indicator which is set by the servicing of a subinstruction with an EXRN function tag. It is reset by the interrupt scan which detects it.

$T_{EI_j} \equiv (T_{\text{External Interrupt}})_j$ = The time at which an external interrupt on the j^{th} channel is presented to the I/O hardware. EI_j is set at this time.

$EI_j \equiv (\text{External Interrupt})_j$ = An interrupt indicator for the j^{th} channel which is set by the coincidence of running time and T_{EI_j} $(T \geq T_{EI_j})$. It is reset by the interrupt scan which detects it.

$OMI_j \equiv (\text{Output Monitor Interrupt})_j$ = An interrupt indicator for the j^{th} output channel which is set by servicing of the last data request of the data frame. It is reset by the interrupt scan which detects it.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-9

IMI_j ≡ (Input Monitor Interrupt)$_j$ = An interrupt indicator for the j^{th} input channel which is set by the servicing of the last data request of the data frame. It is reset by the interrupt scan which detects it.

RCLOKINT ≡ Real Time Clock Interrupt = An interrupt indicator which is set by the servicing of the last data request of the Real Time Clock. It is reset by the interrupt scan which detects it.

DCLOKINT ≡ Day Clock Interrupt = An interrupt indicator which is set by the servicing of the last data request of the Day Clock. It is reset by the interrupt scan which detects it.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-10

$T_{DR_{ji}} \equiv (T_{\text{Data Request}})_{ji}$ = The time at which the i^{th} device on the j^{th} channel presents a data request to its control unit.

$\Delta T_{DR_{ji}} \equiv (\Delta T_{\text{Data Request}})_{ji}$ = The time between data requests by the i^{th} device on the j^{th} channel.

For HS and WB devices on SCS channels and for S/S channels, this is

$$\left(\frac{\text{number of bits}}{\text{character}}\right) \times \left(\frac{\text{line time}}{\text{bit}}\right).$$

For LS devices on SCS channels, this is

$$\left(\frac{\text{line time}}{\text{character}}\right).$$

For the FH880, this is

$$\left(\frac{\text{drum rotation time}}{\text{angular address}}\right) \times \text{(interlace factor)}.$$

For the IIIC, this is

$$\left(\frac{\text{tape movement time}}{\text{inch}}\right) \times \left(\frac{5 \text{ frames}}{\text{number of frames/inch}}\right).$$

For the RCLOK, this is 200 μs.

For the DCLOK, this is 600 ms.

For the 1004, this is $\left(\frac{\text{line time}}{\text{word}}\right)$.

For the CONSOLE, this is $\left(\frac{\text{line time}}{\text{character}}\right)$.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-11

$\Delta T_{DR_{jiCC_{ji}}} \equiv (\Delta T_{\text{Data Request}})_{jiCC_{ji}}$ = The time between data requests by the i^{th} device on the j^{th} PBT channel. This varies between 10 and 20 μsec for a PBT, dependent upon format, word position, and polynomial code. If a constant is to be assumed, replace $\Delta T_{DR_{jiCC_{ji}}}$ by $\Delta T_{DR_{ji}}$. Note that CC_{ji} ranges from one to DF_{ji} and hence its maximum value is 20.

$\Delta T_{B_{ji}} \equiv (\Delta T_{\text{Bit Time}})_{ji}$ = The communications line time per bit for the i^{th} device on the j^{th} channel (SCS, S/S, PBT).

$\Delta T_{FDR_{ji}} \equiv \Delta T_{\text{First Data Request}_{ji}}$ = The time between the first and second output data requests for the i^{th} device on the j^{th} S/S channel.

For a 494/7094 adapter operating at 40.8 kb/s, this is 2.33 μs.

$T_{FDR_{ji}} \equiv (T_{\text{First Data Request}})_{ji}$ ≡ The time at which the first data request by the i^{th} device on the j^{th} channel (PBT) is presented to its control unit.

$FDR_{ji} \equiv (\text{First Data Request})_{ji}$ = An indicator used to indicate a special condition for the i^{th} output device on the j^{th} channel (SCS, S/S, FH880, or IIIC).

For an HS or WB device on an SCS channel, or for an S/S channel device, this indicator is used to indicate the first output data request.

For an FH880 or IIIC channel, it is used to indicate the first two output data requests.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-12

CC_{ji} ≡ (Character Count)$_{ji}$ = The number of the data transfer within the data frame for the i^{th} device on the j^{th} channel.

DF_{ji} ≡ (Data Frame)$_{ji}$ = The number of data transfers in a data frame for the i^{th} device on the j^{th} channel.

Note that for a PBT channel, this is 20 since the channel will operate in a 30-bit transfer mode with 600-bit data blocks.

DIR_{ji} ≡ (Direction)$_{ji}$ = The direction of data flow through the i^{th} device on the j^{th} channel. The two directions are IN, OUT. IN means input to the UNIVAC 494 central processor. OUT means output from the UNIVAC 494 central processor.

$TYPE_{ji}$ ≡ A variable used to identify the type of i^{th} peripheral device connected to the j^{th} SCS or S/S channel. There are six types of peripheral devices for an SCS channel: WBIN, WBOUT, HSIN, HSOUT, LSIN, LSOUT.

WBIN (Wide Band Synchronous Input CLT) is a synchronous input CLT used for wide band data traffic (40.8 kilobits/second). Framing is provided by the recognition of synchronization characters by the CLT and by the central processor's worker program.

WBOUT (Wide Band Synchronous Output CLT) is a synchronous output CLT used for wide band data traffic (40.8 kilobits/second). Framing is provided by the central processor's worker program.

HSIN (High Speed Synchronous Input CLT) is a synchronous input CLT used for high speed data traffic (1, 2, 2.4, 4.8 kilobits/second). Framing is provided by an externally controlled interface.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-13

HSOUT (High Speed Synchronous Output CLT) is a synchronous output CLT used for high speed data traffic (1, 2, 2.4, 4.8 kilobits/second). Framing is provided by the central processor's worker program.

LSIN (Low Speed Asynchronous Input CLT) is an asynchronous input CLT used for teletype traffic (75, 100 words/minute). Framing is provided by recognition of teletype character by the CLT and by the central processor's worker program.

LSOUT (Low Speed Asynchronous Output CLT) is an asynchronous output CLT used for teletype traffic (75, 100 words/minute). Framing is provided by the central processor's worker program.

There are four types of peripheral devices for an S/S channel: RTCCIN, RTCCOUT, CIMIN, DDDOUT.

RTCCIN (Real Time Computer Complex Input) is a synchronous CCU 494/7094 adapter used for wide band data traffic (40.8 kilobits/second). Framing is provided by an externally controlled interface and by the central processor's worker program.

RTCCOUT (Real Time Computer Complex Input) is a synchronous CCU 494/7094 adapter used for wide band data traffic (40.8 kilobits/second). Framing is provided by an externally controlled interface and by the central processor's worker program.

CIMIN (Computer Input Multiplexer Input) is a synchronous CCU 494/7094 adapter used for high speed data traffic (2.4 kilobits/second). Framing is provided by an externally controlled interface and by the central processor's worker program.

DDDOUT (Digital Display Driver Output) is a synchronous CCU 494/7094 adapter used for wide band data traffic (40.8 kilobits/second). Framing is provided by an externally controlled interface and by the central processor's worker program.

$DFINFO_{ji}$ ≡ $(\text{Data Frame Information})_{ji}$ = The set of current values of the information necessary to specify each data frame for the i^{th} device on the j^{th} channel. Generically this consists of information pertaining to an input data frame, an output data frame, and a source data frame. If this were an input data frame, we would be concerned with the subsets pertaining to the input data frame and the source data frame (for new data frames entering the communications processor, these generally would be identical). If this were an output data frame, we would be concerned with the subsets pertaining to the output data frame and the source data frame. The set which is used at any one time to perform input or output data transfers is $\{J_{ji}, I_{ji}, SERIAL_{ji}, MSG_{ji}, SEG_{ji}, DF_{ji}, A_{ji}, BGRP_{ji}, T_{FS_{ji}}\}$. See the definition of DF^{*}_{k}.

$T_{REF_{ji}}$ ≡ $(T_{Reference})_{ji}$ = The time to which all frame starts are referenced for the i^{th} device on the j^{th} channel. This is a convenient method of shifting the entire data frame sequence in time. Note that for the FH880, this is also the time at which address coincidence occurs for the reference drum address (Angular address 0, Angular sector 0) for the i^{th} device on the j^{th} FH880 channel.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-15

$Q_{DF_{ji}} \equiv (\text{Queue}_{\text{Data Frame}})_{ji}$ = The queue of data frames for the i^{th} device on the j^{th} channel. Each entry consists of a set of values for $DFINFO_{ji}$. The current data frame entry on this queue is called QDF*. Note that $(J,I) \epsilon \{Q_{DF_{ji}}\}$ should retain the appropriate values if not specified when an entry is made to this queue.

$INTINFO_{ji} \equiv (\text{Interrupt Information})_{ji}$ = The set of values corresponding to the previous data frame. This is a temporary set which contains the set of values for $DFINFO_{ji}$ after the interrupt has occurred. This set is later transferred to the processing housekeeping when the interrupt has been detected and accepted for processing.

$IE_{ji} \equiv (\text{Interrupt Enable})_{ji}$ = An indicator which is used to allow the algorithm to ignore interrupt actions which are required by the i^{th} device on the j^{th} channel. If this indicator is set, interrupts are allowed to occur and the full interrupt record cycle is necessary to allow data transfer to resume.

$\Delta T_{EI_{ji}} \equiv (\Delta T_{\text{External Interrupt}})_{ji}$ = The time between the recognition of an acknowledge signal for the last data transfer by the i^{th} device on the j^{th} channel, and the presentation of an external interrupt to its control unit.

$T_{EI_{ji}} \equiv (T_{\text{External Interrupt}})_{ji}$ = The time at which the i^{th} device on the j^{th} channel presents an external interrupt to its control unit.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-16

$T_{BAR_{ji}} \equiv (T_{\text{Buffer Action Request}})_{ji}$ = The time at which an output polynomial buffer in the i^{th} device on the j^{th} PBT channel is emptied and causes an external interrupt. If interrupts are not allowed, this variable is used to set up the succeeding data frame.

$PBC_{ji} \equiv (\text{Polynomial Buffer Count})_{ji}$ = An indicator used to indicate the number of polynomial buffers in use by the i^{th} device on the j^{th} PBT channel.

$CAT_{ji} \equiv (\text{Character Availability Time Ratio})_{ji}$ = The ratio of character availability time used during a data transfer for the i^{th} device on the j^{th} channel to the time available for the data transfer.

Note that for a PBT channel, this actually refers to the transfer of a data block (20 transfers).

$C_{ji} \equiv (\text{Character Availability Time})_{ji}$ = The time available for a character transfer for the i^{th} device on the j^{th} channel (SCS, S/S, RCLOK, DCLOK, 1004, CONSOLE).

For a WBIN, WBOUT, HSIN, HSOUT device on an SCS channel, or for a S/S channel, this is $(\Delta T_{DR_{ji}} - \Delta T_{B_{ji}})$.

For an LSIN, LSOUT device on an SCS channel, this is (stop bit time + $\frac{1}{2}$ last information bit time).

For an RCLOK, DCLOK, 1004, or CONSOLE channel, this is assumed to be $\Delta T_{DR_{ji}}$.

For an FH880 or IIIC channel, this is the actual time used to service a data request.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-17

$D_{ji} \equiv (\text{Delay})_{ji}$ = The time delay inserted into the data stream for the i^{th} device on the j^{th} S/S channel if $CAT_{ji} > 1$. The special nature of this interface ensures that exceeding the quoted Character Availability Time does not cause loss of data, but does cause a delay to be inserted in the bit stream.

$CN_{ji} \equiv (\text{Nominal Current Character Availability Time})_{ji}$ = The nominal time available for the current character transfer for the i^{th} device on the j^{th} FH880 or IIIC channel.

$CM_{ji} \equiv (\text{Maximum Current Character Availability Time})_{ji}$ = The maximum time available for the current character transfer for the i^{th} device on the j^{th} FH880 or IIIC channel.

$CNOM_{ji} \equiv (\text{Nominal Character Availability Time})_{ji}$ = The maximum value of CN_{ji} under specified initial conditions. Note that the value differs for input and output transfers.

$CMAX_{ji} \equiv (\text{Maximum Character Availability Time})_{ji}$ = The maximum value of CM_{ji} under specified initial conditions. Note that the value differs for input and output transfers.

$T_{AC_{ji}} \equiv (T_{\text{Address Coincidence}})_{ji}$ = The time at which address coincidence occurs for the first data request in the present data frame for the i^{th} device on the j^{th} FH880 or IIIC channel. Note that the IIIC is not actually addressable, but we may create $T_{AC_{ji}}$ to account for tape movement.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-18

$\Delta T_{LATENCY_{ji}}$ ≡ The maximum time required to reach the desired address by drum rotation (FH880) or tape movement (IIIC) for the i^{th} device on the j^{th} channel. For the FH880, this is 34 ms. For the IIIC, this is an inserted estimate.

$\Delta T_{DA_{ji}}$ ≡ $(\Delta T_{\text{Dead Address}})_{ji}$ ≡ The drum rotation time required between angular address 2047 and angular address 0 for the i^{th} device on the j^{th} FH880 channel. This time is required to switch tracks for succeeding data transfers and is equal to $8 \cdot (\Delta T_{DR_{ji}})$ at an interlace of one or 129.36 μs.

$T_{FH_{ji}}$ ≡ $(T_{\text{Function Hold}})_{ji}$ = The time at which transfers are allowed between device and control unit for the i^{th} device on the j^{th} FH880 or IIIC channel.

$\Delta T_{FH_{ji}}$ ≡ $(\Delta T_{\text{Function Hold}})_{ji}$ = The time between initiation of a function and the first data transfers between device and control unit for the i^{th} device on the j^{th} FH880 or IIIC Channel. For an FH880 channel, this is 110 μs.

$\Delta T_{CU_{ji}}$ ≡ $(\Delta T_{\text{Control Unit}})_{ji}$ = The minimum downtime required by the j^{th} FH880 or IIIC channel control unit between successive data requests for the i^{th} device. Note that the value of this quantity includes allowance for control line time and is dependent upon direction (input or output).

Definitions
UNIVAC 494 Algorithm
FIGURE A1-19

$BGRP_{ji}$ ≡ $(\text{Buffer Group})_{ji}$ = The name of the buffer group used by the i^{th} device on the j^{th} channel. This variable is a member of $DFINFO_{ji}$.

$BCTR_{BGRP_{ji}}$ ≡ $(\text{Buffer Counter})_{BGRP_{ji}}$ = The number of buffers currently allocated to use by $BGRP_{ji}$ by worker programs. Note that the maximum value of the subscript is equal to the total number of distinct buffer groups.

$BHIST_{BGRP_{ji}}$ ≡ $(\text{Buffer Histogram})_{BGRP_{ji}}$ = The number of buffers actually in use by $BGRP_{ji}$. The endpoints of information flow (the first data transfer of an input data frame, the last date transfer of an output data frame) are measurement points for this variable. Additional measurement points are generated by worker programs. Note that the toggling of a buffer is a change in $BCTR_{BGRP_{ji}}$ and must be correlated with the later increase in $BHIST_{BGRP_{ji}}$ to determine that the toggling operation was successful.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-20

β Segment ≡ A list of subinstructions defining an instruction sequence.

α Segment ≡ A list of β segments and the number of times each is to be executed. Each α segment operates at a single priority level and is a predetermined processing entity with all branching removed.

A Segment ≡ A list of α segments and the variables necessary to produce the correct flow path through them. Each A segment of the worker type specifies the chain of processing to be executed for a specific type of data frame.

P^*_{hk} ≡ $(\text{Program}^*)_{hk}$ = The set of processing housekeeping variables required to define the type h program operating at level k. The set is made up of the following members defined below,

$\{H_{hk}, K_{hk}, X_{hk}, A_{hk}, \alpha CTR_{hk}, \alpha MAX_{hk}, \alpha_{hk}, \beta CTR_{hk}, \beta MAX_{hk}, \beta_{hk}, LCTR_{hk}, LMAX_{hk}, SCTR_{hk}, SMAX_{hk}, HSKP_{hk}\}$.

H_{hk} ≡ The type of program for α_{hk}.

K_{hk} ≡ The program priority level for α_{hk}.

X_{hk} ≡ The flow path to be followed if A_{hk} is of the executive or switcher type; DUMMY if A_{hk} is of the worker type.

A_{hk} ≡ $(\text{A Name})_{hk}$ = The name of the A segment in which the next α segment will be found. See further explanation of the contents of A_{hk} below.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-21

$\alpha CTR_{hk} \equiv (\alpha \text{ Counter})_{hk}$ = The entry counter for A_{hk} which determines the next α segment entry to be picked up.

$\alpha MAX_{hk} \equiv (\alpha \text{ Maximum})_{hk}$ = The maximum number of α segment entries in A_{hk}.

$\alpha_{hk} \equiv (\alpha \text{ Name})_{hk}$ = The name of the α segment. See further explanation of the contents of α_{hk} below.

$\beta CTR_{hk} \equiv (\beta \text{ Counter})_{hk}$ = The entry counter for α_{hk} which determines the next β segment entry to be picked up.

$\beta MAX_{hk} \equiv (\beta \text{ Maximum})_{hk}$ = The maximum number of β segment entries in α_{hk}.

$\beta_{hk} \equiv (\beta \text{ Name})_{hk}$ = The name of the β segment. See further explanation of the contents of β_{hk} below.

$LCTR_{hk} \equiv (\text{Loop Counter})_{hk}$ = The loop counter which determines the current number of excursions thru β_{hk}.

$LMAX_{hk} \equiv (\text{Loop Maximum})_{hk}$ = The maximum number of excursions thru β_{hk}.

$SCTR_{hk} \equiv (\text{Subinstruction Counter})_{hk}$ = The entry counter for β_{hk} which determines the next subinstruction entry to be picked up.

$SMAX_{hk} \equiv (\text{Subinstruction Maximum})_{hk}$ = The maximum number of subinstruction entries in β_{hk}.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-22

$HSKP_{hk}$ ≡ $(\text{Housekeeping})_{hk}$ = The housekeeping mode to be observed when α_{hk} is exhausted if h is equal to w. This variable takes on the values RJP, JP, and DUMMY to allow the return to a predecessor α segment after the execution of the worker α segment indicated by the RJP function tag.

PID_{hk} ≡ $(\text{Program Identification})_{hk}$ = The subset of P^*_{hk} used to identify a program. The subset is made up of the following members defined above, $\{H_{hk}, K_{hk}, A_{hk}, \alpha CTR_{hk}, \alpha_{hk}, \beta_{hk}\}$.

$A_{hk}\big|_{h=w}$ ≡ $(\text{A Name})_{w,k}$ = The name of the A segment containing the set of processing housekeeping variables for the α segment entry defined by αCTR_{hk}. The set is made up of the following members defined below, {Pα, PFT, WK, WH, WX, WA, WαCTR, Wα, EA, EαCTR}.

Pα ≡ Predecessor α = The name of the α segment we should have been executing to pick up this α segment entry.

PFT ≡ Predecessor Function Tag = The name of the function tag we should have executed to pick up this α segment entry. The allowable function tags are EXRN, RJP, JP.

WK ≡ Worker K = The priority level of Wα.

WH ≡ Worker H = The type of program of Wα.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-23

WX ≡ Worker X = The flow path of Wα if WA is of the executive or switcher type.

WA ≡ Worker A Name = The name of the A segment in which the next α segment will be found for Wα.

WαCTR ≡ Worker α Counter = The entry counter for WA which determines the next α segment entry to be picked up.

Wα ≡ Worker α = The name of the worker α segment to be executed if PFT is RJP or JP. The name of the α segment to be queued if PFT is EXRN. (If Wα is DUMMY, then no queue operation will take place and the remaining five worker variables are also DUMMY.)

EA ≡ Executive A Name = The name of the A segment in which the next α segment will be found if PFT is EXRN; DUMMY if PFT is not EXRN.

EαCTR ≡ Executive α Counter = The entry counter for EA which determines the next α segment to be picked up if PFT is EXRN; DUMMY if PFT is not EXRN.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-24

$A_{hk}\,|_{h=e,s}$ ≡ $(\text{A Name})_{(e,s),k}$ = The name of the A segment containing the set of processing housekeeping variables for the α segment entry defined by αCTR_{hk}. The set is made up of the following members defined below,

{Q, YK, YH, YX, YA, YαCTR, Yα, NK, NH, NX, NA, NαCTR, Nα}.

Q ≡ Queue Name = The name of the queue if the flow path must be determined by determining the value of a queue related variable. This variable is DUMMY if no such determination is necessary.

YK ≡ Yes K = The priority level of Yα.

YH ≡ Yes H = The type of program of Yα.

YX ≡ Yes X = The flow path of Yα if YA is of the executive or switcher type.

YA ≡ Yes A Name = The name of the A segment in which the next α segment will be found for Yα.

YαCTR ≡ Yes α Counter = The entry counter for YA which determines the next α segment entry to be picked up.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-25

Yα ≡ Yes α = The name of the α segment to be executed if the YES flow path was chosen. If Yα is DUMMY, we immediately take the next α segment entry, asking the appropriate question determined by YX to find the flow path to be followed.

NK ≡ No K = The priority level of Nα.

NH ≡ No H = The type of program of Nα.

NX ≡ No X = The flow path of Nα if NA is of the executive or switcher type.

NA ≡ No A = The name of the A segment in which the next α segment will be found for Nα.

NαCTR ≡ No α Counter = The entry counter for NA which determines the next α segment entry to be picked up.

Nα ≡ No α = The name of the α Segment to be executed if the NO flow path was chosen. If Nα is DUMMY, we immediately take the next α segment entry, asking the appropriate question determined by NX to find the flow path to be followed.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-26

$\alpha_{hk} \equiv (\alpha \text{ Name})_{hk}$ = The name of the α segment containing the set of processing housekeeping variables for the β segment entry defined by βCTR_{hk}. The set is made up of the following members defined below, $\{\beta, LMAX\}$.

$\beta \equiv \beta$ Name = The name of the β segment.

LMAX ≡ Loop Maximum = The maximum number of excursions through β.

$\beta_{hk} \equiv (\beta \text{ Name})_{hk}$ = The name of the β segment containing the set of processing housekeeping variables for the subinstruction entry defined by $SCTR_{hk}$. The set is made up of the following members defined below, {FT,SILO,CCMF,CCLDS}.

FT ≡ Function Tag = A variable used to define the subinstruction which is to be executed and to perform any operation necessary for algorithm housekeeping.

A NORMAL function tag causes the algorithm to execute a subinstruction.

An EXRN function tag causes the algorithm to execute a subinstruction and generate an executive return interrupt.

An RJP function tag causes the algorithm to set up processing housekeeping to execute an α segment found in A_{hk} and return to the α segment containing the RJP function tag.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-27

A JP function tag causes the algorithm to set up processing housekeeping to execute an α segment found in A_{hk}.

An EF function tag causes the algorithm to generate a command to a peripheral device.

An EFTYPE function tag causes the algorithm to generate the additional variables necessary for a command to an addressable peripheral device.

An SC function tag causes the algorithm to generate a store channel command to a peripheral device.

An RC function tag causes the algorithm to reinitiate a channel.

A SET:SERIAL function tag causes the algorithm to generate a value of SERIAL for data frame information.

A SET:MSG function tag causes the algorithm to generate a value of MSG for data frame information.

A SET:SEG function tag causes the algorithm to generate a value of SEG for data frame information.

A SET:DF function tag causes the algorithm to generate a value of DF for data frame information.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-28

A SET:A function tag causes the algorithm to generate a value of A for data frame information.

A SET:BGRP function tag causes the algorithm to generate a value of BGRP for data frame information.

A SET:FS function tag causes the algorithm to generate a value of T_{FS} for data frame information.

A SET:BCTR function tag causes the algorithm to generate a value of $BCTR_{BGRP}$.

A SET:BHIST function tag causes the algorithm to generate a value of $BHIST_{BGRP}$.

A SET:BUSY function tag causes the algorithm to generate a value of $QBUSY_Q$, $QGRPBUSY_Q$, or $QSGRPBUSY_Q$.

A SET:QDF function tag causes the algorithm to add or delete an entry from a data frame queue.

A SET:QP function tag causes the algorithm to add or delete an entry from a processing queue. See FIGURES A1-37,38,39,40,41.

SILO ≡ Subinstruction Interrupt Lockout = A variable used to define whether interrupts are locked out by the hardware if FT is NORMAL or EXRN. (1 inhibits interrupts; 0 allows interrupts.) See FIGURES A1-37,38,39,40,41.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-29

CCMF ≡ Clock Cycles - Memory Free = A variable used to define the number of clock cycles in this subinstruction until memory is free, if FT is NORMAL or EXRN. (7 if memory is used; 0 if memory is not used.) See FIGURES A1-37,38,39,40,41.

CCLDS ≡ Clock Cycles - Last Data Scan = A variable used to define the number of clock cycles in this subinstruction until the last data scan is initiated. (Total number of clock cycles in this subinstruction minus 3.) See FIGURES A1-37,38,39,40,41.

$MAX\alpha_{A_{hk}}$ ≡ $(\text{Maximum } \alpha)_{A_{hk}}$ = The maximum number of α segment entries in the A segment specified by A_{hk}. There is one variable associated with each A segment.

$MAX\beta_{\alpha_{hk}}$ ≡ $(\text{Maximum } \beta)_{\alpha_{hk}}$ = The maximum number of β segment entries in the α segment specified by α_{hk}. There is one variable associated with each α segment.

$MAXS_{\beta_{hk}}$ ≡ $(\text{Maximum Subinstruction})_{\beta_{hk}}$ = The maximum number of subinstruction entries in the β segment specified by β_{hk}. There is one variable associated with each β segment.

FT ≡ Function Tag = A temporary variable used to hold the value of FT ϵ $\{\beta_{hk}\}$.

SILO ≡ Subinstruction Interrupt Lockout = A temporary variable used to hold the value of SILO ϵ $\{\beta_{hk}\}$.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-30

CCMF ≡ Clock Cycles - Memory Free = A temporary variable used to hold the value of CCMF ϵ $\{\beta_{hk}\}$.

CCLDS ≡ Clock Cycles - Last Data Scan = A temporary variable used to hold the value of CCLDS ϵ $\{\beta_{hk}\}$.

YK ≡ Yes K = A temporary variable used to hold the value of YK ϵ $\{A_{hk}\}$, NK ϵ $\{A_{hk}\}$, or WK ϵ $\{A_{hk}\}$.

YH ≡ Yes H = A temporary variable used to hold the value of YH ϵ $\{A_{hk}\}$, NH ϵ $\{A_{hk}\}$, or WH ϵ $\{A_{hk}\}$.

YX ≡ Yes X = A temporary variable used to hold the value of YX ϵ $\{A_{hk}\}$, NX ϵ $\{A_{hk}\}$, or WX ϵ $\{A_{hk}\}$.

YA ≡ Yes A Name = A temporary variable used to hold the value of YA ϵ $\{A_{hk}\}$, NA ϵ $\{A_{hk}\}$, or WA ϵ $\{A_{hk}\}$.

YαCTR ≡ Yes α Counter = A temporary variable used to hold the value of YαCTR ϵ $\{A_{hk}\}$, NαCTR ϵ $\{A_{hk}\}$, or WαCTR ϵ $\{A_{hk}\}$.

Yα ≡ Yes α = A temporary variable used to hold the value of Yα ϵ $\{A_{hk}\}$, Nα ϵ $\{A_{hk}\}$, or Wα ϵ $\{A_{hk}\}$.

X ≡ A temporary variable used to hold the value of X_{hk}.

BGRP ≡ Buffer Group Name = A temporary variable used to hold the name of the buffer group.

Q ≡ Queue Name = A temporary variable used to hold the name of the queue. There are two types of queues used by the algorithm: data frame queues and processing queues. A data frame queue entry consists of a value for DF^*_k; the current entry is denoted by QDF*. A processing queue entry consists of values for P^*_{hk}, $P^*_{p,k}$, and DF^*_k; the current entry is denoted by QP^*_w, QP^*_p, and QDF*.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-31

QGRP ≡ Queue Group Name = A temporary variable used to hold the name of the queue group.

QSGRP ≡ Queue Super Group Name = A temporary variable used to hold the name of the queue super group.

$QBUSY_Q$ ≡ $(\text{Queue Busy})_Q$ = An indicator used to denote whether a queue is busy. This indicator is not expected to be used.

$QCTR_Q$ ≡ $(\text{Queue Counter})_Q$ = An entry counter used to denote the number of entries in Q.

$QGRP_Q$ = $(\text{Queue Group Name})_Q$ = The name of the queue group to which Q belongs.

$QGRPBUSY_{QGRP}$ ≡ $(\text{Queue Group Busy})_{QGRP}$ = An indicator used to denote whether a queue group is busy.

$QGRPCTR_{QGRP}$ ≡ $(\text{Queue Group Counter})_{QGRP}$ = An entry counter used to denote the number of entries in queues which are members of QGRP.

$QSGRP_{QGRP}$ ≡ $(\text{Queue Super Group Name})_{QGRP}$ = The name of the queue super group to which QGRP belongs.

$QSGRPBUSY_{QSGRP}$ ≡ $(\text{Queue Super Group Busy})_{QSGRP}$ = An indicator used to denote whether a queue super group is busy. This indicator is not expected to be used.

$QSGRPCTR_{QSGRP}$ ≡ $(\text{Queue Super Group Counter})_{QSGRP}$ = An entry counter used to denote the number of entries in queue groups which are members of QSGRP.

QPTGS1 ≡ Queue Place To Go Suspend Level 1 = The name of the queue used for suspension of a program running at priority level 1.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-32

QPTGS2 ≡ Queue Place To Go Suspend Level 2 = The name of the queue used for suspension of a program running at priority level 2.

QPTGS3 ≡ Queue Place To Go Suspend Level 3 = The name of the queue used for suspension of a program running at priority level 3.

$QINEXT_j$ ≡ (Queue Input External Interrupt)$_j$ = The name of the queue used for input external interrupts on the j^{th} channel.

$QINMON_j$ ≡ (Queue Input Monitor Interrupt)$_j$ ≡ The name of the queue used for input monitor interrupts on the j^{th} channel.

$QOUTMON_j$ ≡ (Queue Output Monitor Interrupt)$_j$ = The name of the queue used for output monitor interrupts on the j^{th} channel.

$QOUTEXT_j$ ≡ (Queue Output External Interrupt)$_j$ = The name of the queue used for output external interrupts on the j^{th} channel.

QRCLOK ≡ Queue Real Time Clock Interrupt = The name of the queue used for real time clock interrupts.

QDCLOK ≡ Queue Day Clock Interrupt = The name of the queue used for day clock interrupts.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-33

$DF^*_k \equiv (\text{Data Frame}^*)_k$ = The set of data frame variables defining the data frame associated with P^*_{hk}. The set is made up of the following subsets defined below, $\{DF^*_{i,k}, DF^*_{o,k}, DF^*_{s,k}\}$. Each of the three subsets may be identified as $DF^*_{d,k}$, where d takes on the values i,o,s. The subset $DF^*_{d,k}$ is made up of the following members defined below,
{J,I,SERIAL,MSG,SEG,DF,A,GBRP,T_{FS}}.

$DF^*_{i,k} \equiv (\text{Data Frame}^*)_{i,k}$ = The set of data frame variables defining the input data frame associated with P^*_{hk}.

$DF^*_{o,k} \equiv (\text{Data Frame}^*)_{o,k}$ = The set of data frame variables defining the output data frame associated with P^*_{hk}.

$DF^*_{s,k} \equiv (\text{Data Frame}^*)_{s,k}$ = The set of data frame variables defining the source data frame associated with P^*_{hk}.

J ≡ The channel associated with the data frame.

I ≡ The device associated with the data frame.

SERIAL ≡ Serial Number = The sequential serial number associated with the data frame.

MSG ≡ Message Number = The sequential message number associated with the data frame.

SEG ≡ Segment Number = The sequential segment number associated with the data frame.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-34

DF ≡ Data Frame = The number of characters in the data frame. Note that for an output data frame using output monitor interrupts this is one greater than the actual number of characters in the data frame to insure proper timing of the interrupt.

A ≡ A Name = The name of the A segment associated with the data frame. The α segment to be executed following an interrupt and the succeeding flow path is determined by A.

BGRP ≡ Buffer Group Name = The name of the buffer group associated with the data frame.

T_{FS} ≡ $T_{\text{Frame Start}}$ = The time at which the data frame was initiated with respect to $T_{REF_{JI}}$.

$DFID_k$ ≡ $(\text{Data Frame Identification})_k$ = The subset of DF^*_k used to identify a data frame. The subset is made up of the following members defined above, $\{(J,I,SERIAL) \in DF^*_{i,k}, (J,I,SERIAL) \in DF^*_{o,k}, (J,I,SERIAL) \in DF^*_{s,k}\}$.

Definitions
UNIVAC 494 Algorithm
FIGURE A1-35

k	$P^*_{w,k}$ (h=w)	$P^*_{e,k}$ (h=e)	$P^*_{s,k}$ (h=s)	$P^*_{p,k}$ (h=p)	DF^*_k: $DF^*_{i,k}$ (d=i)	$DF^*_{o,k}$ (d=o)	$DF^*_{s,k}$ (d=s)
INT							
0							
1							
2							
3							

$$P^*_{hk} \equiv \left\{H_{hk}, K_{hk}, X_{hk}, A_{hk}, \alpha CTR_{hk}, \alpha MAX_{hk}, \alpha_{hk}, \beta CTR_{hk}, \beta MAX_{hk}, \beta_{hk}, LCTR_{hk}, LMAX_{hk}, SCTR_{hk}, SMAX_{hk}, HSKP_{hk}\right\}$$

$$DF^*_k \equiv \left\{DF^*_{i,k}, DF^*_{o,k}, DF^*_{s,k}\right\}$$

$$DF^*_{d,k} \equiv \left\{J, I, SERIAL, MSG, SEG, DF, A, BGRP, T_{FS}\right\}$$

$$A_{hk}\Big|_{h=w} \equiv \left\{P\alpha, PFT, WK, WH, WX, WA, W\alpha CTR, W\alpha, EA, E\alpha CTR\right\}$$

$$A_{hk}\Big|_{h=e,s} \equiv \left\{Q, YK, YH, YX, YA, Y\alpha CTR, Y\alpha, NK, NH, NX, NA, N\alpha CTR, N\alpha\right\}$$

$$\alpha_{hk} \equiv \left\{\beta, LMAX\right\}$$

$$\beta_{hk} \equiv \left\{FT, SILO, CCMF, CCLDS\right\}$$

Definitions: Program Housekeeping Master List
UNIVAC 494 Algorithm
FIGURE A1-36

Function Tag (FT)	Subinstruction Interrupt Lockout (SILO)	Clock
NORMAL	Interrupt lockout for this subinstruction. (1 inhibits interrupts; 0 allows interrupts)	Number of cl instruction (7 if memory is not used)
EXRN	Interrupt lockout for this subinstruction. (1 inhibits interrupts; 0 allows interrupts)	Number of cl instruction (7 if memory is not used)
RJP	DUMMY	DUMMY
JP	DUMMY	DUMMY
EF	1	i
	1	o
	DUMMY	The value o

Cycles-Memory Free (CCMF)	Clock Cycles-Last Data Scan (CCLDS)	Operation
ock cycles in this sub- until memory is free. is used; 0 if memory	Number of clock cycles in this sub-instruction until the last data scan is initiated. (Total number of clock cycles in this subinstruction minus 3)	Set up execution information for a NORMAL subinstruction.
ock cycles in this sub- until memory is free. is used; 0 if memory	Number of clock cycles in this sub-instruction until the last data scan is initiated. (Total number of clock cycles in this subinstruction minus 3)	Set up execution information for an EXRN subinstruction and cause an EXRN interrupt.
	DUMMY	Set up housekeeping for a RJP and pick up the first subinstruction of the next α. Execution information for the RJP subinstruction is found in the previous NORMAL subinstruction.
	DUMMY	Set up housekeeping for a JP and pick up the first subinstruction of the next α. Execution information for the JP subinstruction is found in the previous NORMAL subinstruction.
	DUMMY	Create a FN13 data request for the peripheral device specified by the input data frame information.
	DUMMY	Create a FN13 data request for the peripheral device specified by the output data frame information.
? i.	The value of j.	Create a FN13 data request for the peripheral device specified by CCMF and CCLDS. Execution information for the EF subinstruction is found in the succeeding NORMAL subinstruction.

Definitions: Function Tag
UNIVAC 494 Algorithm
FIGURE A1-37

Function Tag (FT)	Subinstruction Interrupt Lockout (SILO)	Clock C
EF TYPE	The value of $TYPE_{FN13}$ for the previous EF subinstruction. (1 for input or output; 2 for a drum termination command)	The value of vious EF subi specific late CCLDS; 1 for average laten latency)
SC	1	i
	1	o
	DUMMY	The value of
RC	1	i
	1	o
	DUMMY	i

ycles-Memory Free (CCMF)	Clock Cycles-Last Data Scan (CCLDS)	Operation
$LATENCY_{FN13}$ for the pre-nstruction. (0 for a ncy to be determined by minimum latency; 2 for cy; 3 for maximum	The value of $ADDRESS_{FN13}$ for the previous EF subinstruction if $LATENCY_{FN13}$ is 0.	Provide the variables necessary for an addressable peripheral device to determine the time at which it begins to transfer data. Execution information for the preceeding EF subinstruction is found in the succeeding NORMAL subinstruction.
	DUMMY	Create a FN17 data request for the peripheral device specified by the input data frame information.
	DUMMY	Create a FN17 data request for the peripheral device specified by the output data frame information.
i.	The value of j.	Create a FN17 data request for the peripheral device specified by CCMF and CCLDS. Execution information for the SC subinstruction is found in the succeeding NORMAL subinstruction.
	DUMMY	Reinitiate the input channel for the peripheral device specified by the input data frame information.
	DUMMY	Reinitiate the output channel for the peripheral device specified by the output data frame information.
	The value of j.	Reinitiate the input channel specified by CCLDS.

Definitions: Function Tag
UNIVAC 494 Algorithm
FIGURE A1-38

Function Tag (FT)	Subinstruction Interrupt Lockout (SILO)	Clock Cy
	DUMMY	o
SET:SERIAL	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	The value of the data fram used as a ref data frame; o s for source no reference
SET:MSG	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	The value of the data fram used as a ref data frame; o frame; s for DUMMY for no
SET:SEG	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	The value of the data fram used as a ref put data fram frame; s for DUMMY for no
SET:DF	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	The value of the data fram used as a ref data frame; o s for source no reference
SET:A	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	DUMMY

ycles-Memory Free (CCMF)	Clock Cycles-Last Data Scan (CCLDS)	Operation
	The value of j.	Reinitiate the output channel specified by CCLDS. Execution information for the RC subinstruction is found in the succeeding NORMAL subinstruction.
direction specifying e information to be erence. (i for input for output data frame; data frame; DUMMY for data frame)	The value (plus or minus) to be added to the value of SERIAL contained in the data frame information specified by CCMF. The new value is inserted in the data frame information specified by SILO.	Provide a value of SERIAL for data frame information (input, output, or source). There is no execution information associated with this subinstruction.
direction specifying e information to be erence. (i for input for output data source data frame; reference data frame)	The value (plus or minus) to be added to the value of MSG contained in the data frame information specified by CCMF. The new value is inserted in the data frame information specified by SILO.	Provide a value of MSG for data frame information (input, output, or source). There is no execution information associated with this subinstruction.
direction specifying e information to be erence. (i for in- e; o for output data source data frame; reference data frame)	The value (plus or minus) to be added to the value of SEG contained in the data frame information specified by CCMF. The new value is inserted in the data frame information specified by SILO.	Provide a value of SEG for data frame information (input, output, or source). There is no execution information associated with this subinstruction.
direction specifying e information to be erence. (i for input for output data frame; data frame; DUMMY for data frame)	The value (plus or minus) to be added to the value of DF contained in the data frame information specified by CCMF. The new value is inserted in the data frame information specified by SILO.	Provide a value of DF for data frame information (input, output, or source). There is no execution information associated with this subinstruction.
	The value of A.	Provide a value of A for data frame information (input, output, or source). There is no execution information associated with this subinstruction.

Definitions: Function Tag
UNIVAC 494 Algorithm
FIGURE A1-39

Function Tag (FT)	Subinstruction Interrupt Lockout (SILO)	Clock Cyc
SET:BGRP	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	The value of d data frame inf as a reference frame; o for o for source dat reference data
SET:FS	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	The value of d data frame inf a reference. o for output d data frame; DU data frame)
SET:BCTR	The value of BGRP if the value of CCMF is DUMMY. DUMMY if the value of CCMF is not DUMMY.	The value of d the data frame as a reference frame; o for o for source dat reference data
SET:BHIST	The value of BGRP if the value of CCMF is DUMMY. DUMMY if the value of CCMF is not DUMMY.	The value of d the data frame used as a refe data frame; o s for source d no reference d
SET:BUSY	The value of Q.	1
	The value of QGRP.	2
	The value of QSGRP.	DUMMY
SET:QDF	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	1
	The value of direction specifying the data frame information to be used. (i for input data frame; o for output data frame; s for source data frame)	1

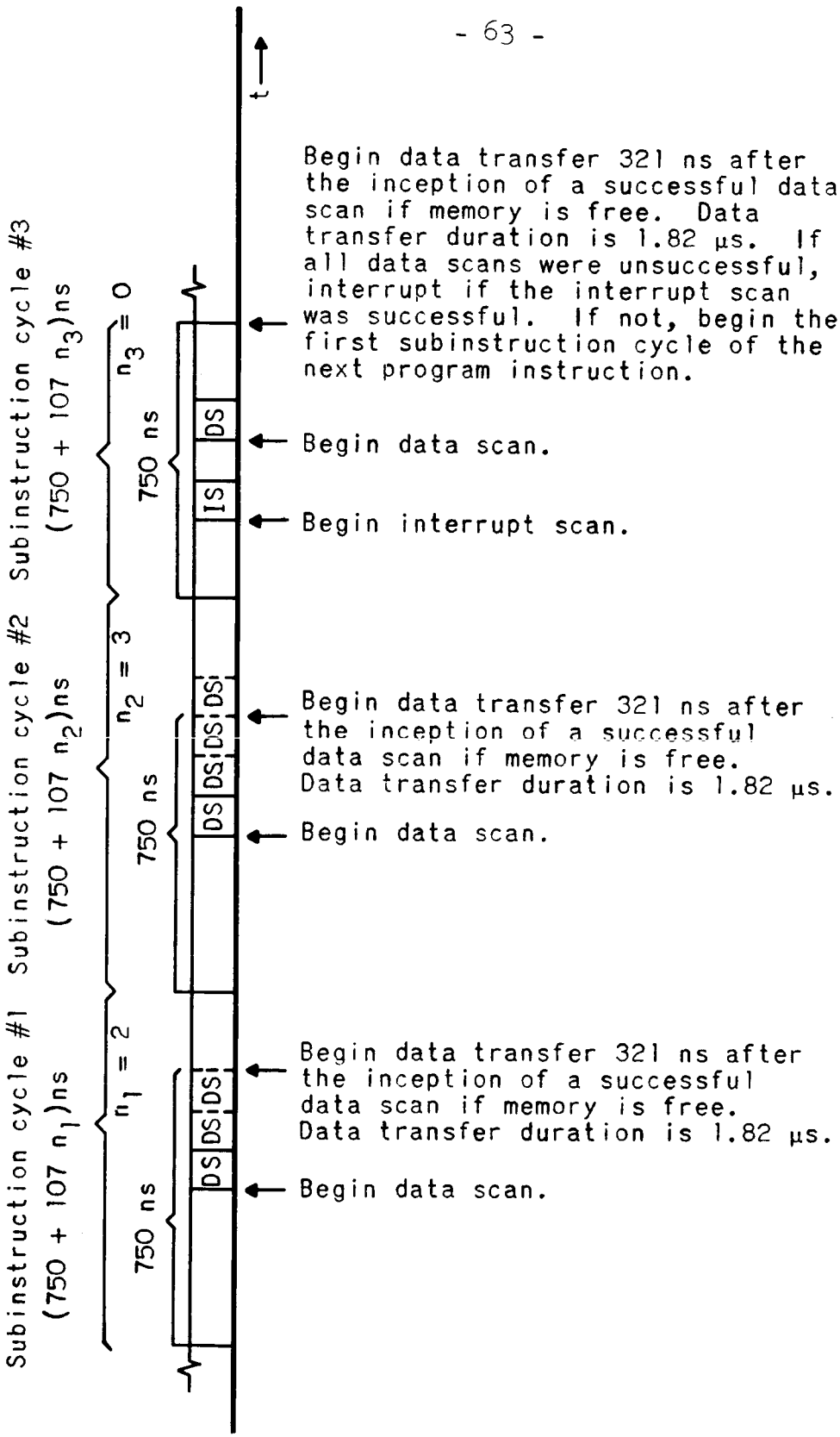
.es-Memory Free (CCMF)	Clock Cycles-Last Data Scan (CCLDS)	Operation
.rection specifying the ɔrmation to be used . (i for input data ıtput data frame; s ı frame; DUMMY for no frame)	The value of BGRP to be inserted in the data frame information specified by SILO if the value of CCMF is DUMMY. DUMMY if the value of CCMF is not DUMMY.	Provide a value of BGRP for data frame information (input, output, or source). There is no execution information associated with this subinstruction.
.rection specifying the ɔrmation to be used as (i for input data frame; ıta frame; s for source MMY for no reference	DUMMY	Provide a value of T_{FS} for data frame information (input, output, or source). There is no execution information associated with this subinstruction.
.rection specifying information to be used . (i for input data ıtput data frame; s ı frame; DUMMY for no frame)	The value (plus or minus) to be added to the value of $BCTR_{BGRP}$.	Provide a value of $BCTR_{BGRP}$. There is no execution information associated with this subinstruction.
.rection specifying information to be rence. (i for input for output data frame; ıta frame; DUMMY for ıta frame)	The value (plus or minus) to be added to the value of $BHIST_{BGRP}$.	Provide a value of $BHIST_{BGRP}$. There is no execution information associated with this subinstruction.
	The value of $QBUSY_{Q}$.	Provide a value for $QBUSY_{Q}$.
	The value of $QGRPBUSY_{QGRP}$.	Provide a value for $QGRPBUSY_{QGRP}$.
	The value of $QSGRPBUSY_{QSGRP}$.	Provide a value for $QSGRPBUSY_{QSGRP}$. There is no execution information associated with this subinstruction.
	1	Enter the data frame queue specified by SILO with the data frame information used by the current processing information.
	DUMMY	Remove an entry from the data frame queue specified by SILO.

Definitions: Function Tag
UNIVAC 494 Algorithm
FIGURE A1-40

Function Tag (FT)	Subinstruction Interrupt Lockout (SILO)	Clock Cy
	The value of Q.	DUMMY
	The value of Q.	DUMMY
SET:QP	The value of Q.	1
	The value of Q.	2
	The value of Q.	2
	The value of Q.	DUMMY

cles-Memory Free (CCMF)	Clock Cycles-Last Data Scan (CCLDS)	Operation
	1	Enter the data frame queue specified by SILO with the data frame information used by the current processing information.
	DUMMY	Remove an entry from the data frame queue specified by SILO.
	DUMMY	Enter the processing queue specified by SILO with the current processing and data frame information.
	DUMMY	Replace the current processing and data frame information with the first entry from the processing queue specified by SILO.
	1	Replace the current processing and data frame information with the first entry from the processing queue specified by SILO and remove the first entry from the queue.
	1	Remove the first entry from the queue specified by SILO.

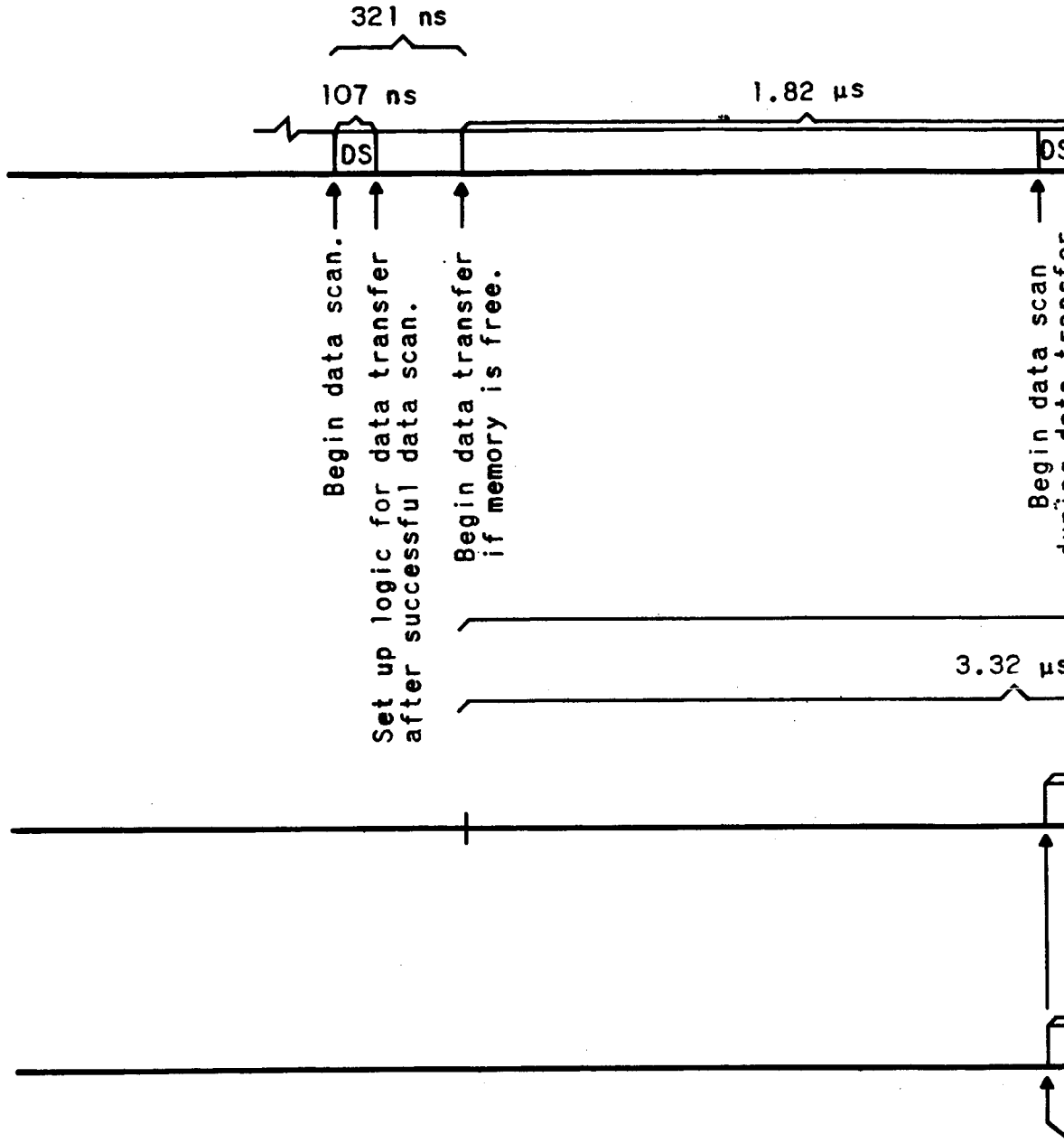
Definitions: Function Tag
UNIVAC 494 Algorithm
FIGURE A1-41



Definitions: Three subinstruction cycle instruction with scan cycles and no memory overlap.

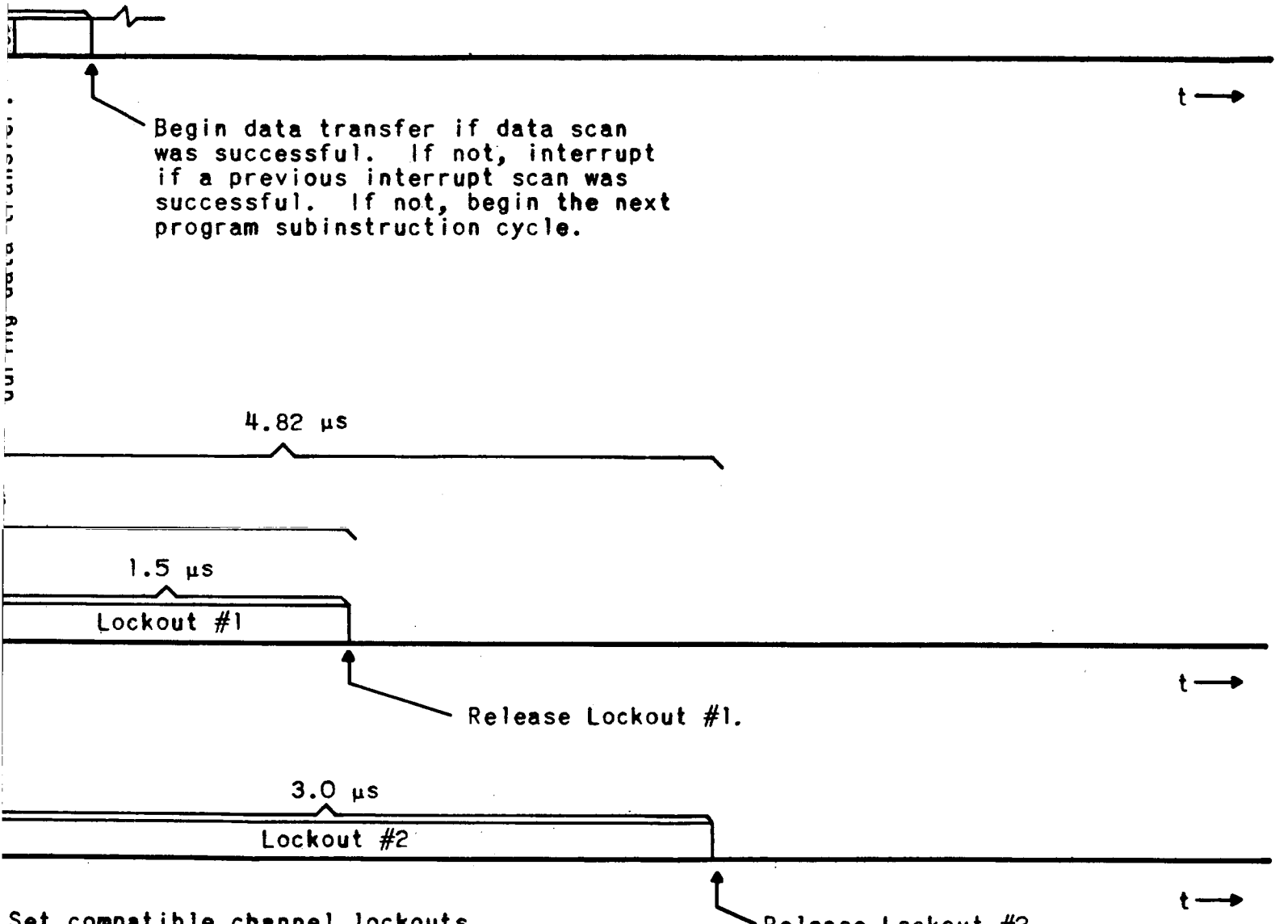
UNIVAC 494 Algorithm

FIGURE A1-42



Definitions: Data tra

UNI



Set compatible channel lockouts.
If the current transfer is an input, set all compatible channel group input and output lockouts to Lockout #1.
If the current transfer is an output, set all compatible channel group input lockouts to Lockout #1, this compatible channel group output lockout to Lockout #2, and all other compatible channel group output lockouts to Lockout #1.

nsfer with compatible channel lockout.

AC 494 Algorithm

FIGURE A1-43

Fault Interrupts

- Parity Error, Bank 0
- Parity Error, Bank 1
- Parity Error, Bank 2
- Parity Error, Bank 3
- Guard Mode
- Executive Return and Test & Set
- Floating point overflow/underflow
- Illegal Instruction

I/O Interrupts

- BCW Parity Error
- I/O Data Parity Error
- Power Loss
- External Interrupt (ESI)
- Input Monitor (ESI)
- Output Monitor (ESI)
- External Interrupt (ISI)
- Input Monitor (ISI)
- Output Monitor (ISI)
- Day Clock Interrupt
- Real Time Clock Interrupt
- External Sync #1
- External Sync #2

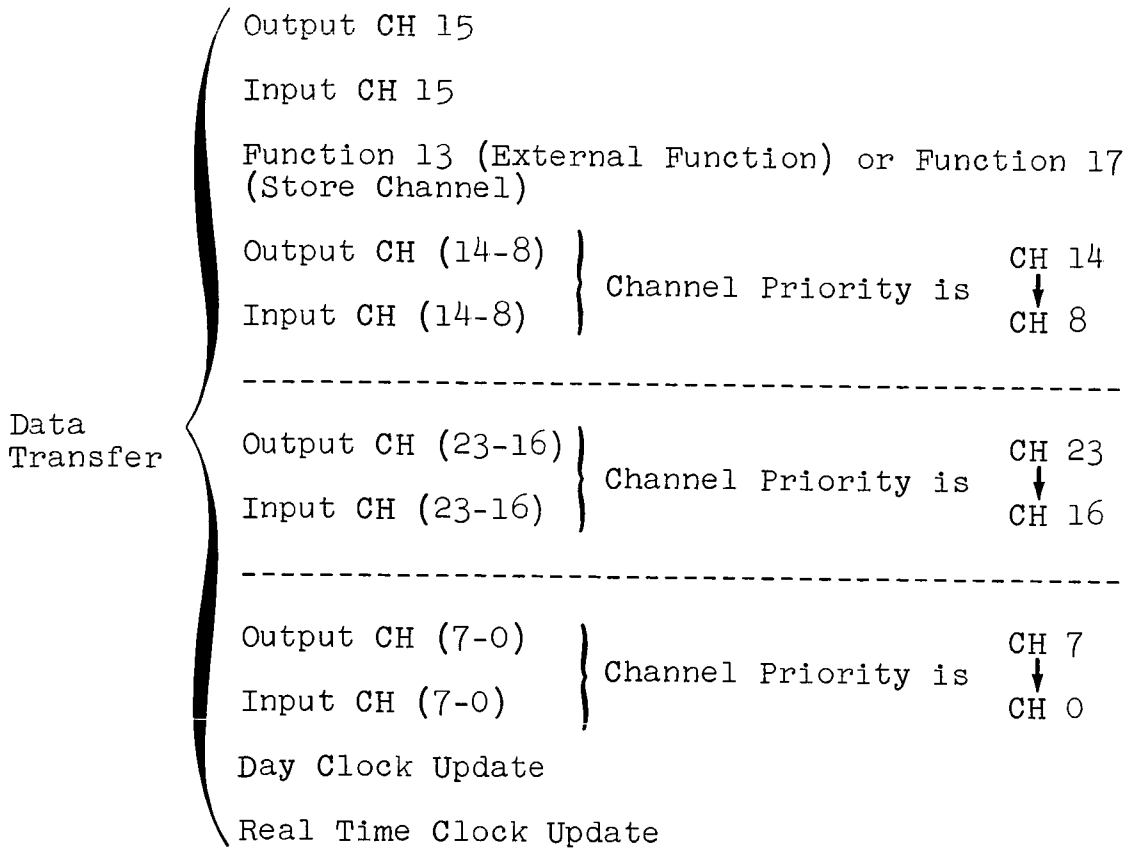
For these interrupts, the associated channel priority is as follows:

CH 15 ↓ CH 8

CH 23 ↓ CH 16

CH 7 ↓ CH 0

Definitions: Interrupt Scan Priority
UNIVAC 494 Algorithm
FIGURE A1-44



Note that in the case of a normal channel group with multiple data requests, we will service a transfer type opposite to that being completed. However, if an opposite transfer type does not exist, we will next service the same transfer type as that being completed. Channel 15 is not affected by this rule. If a hiatus appears in the request stream, we revert to the priority given above.

For a compatible channel group, input/output alternation is accomplished by input/output lockout times. (These lockout times are analogous to the mask times used in this algorithm.)

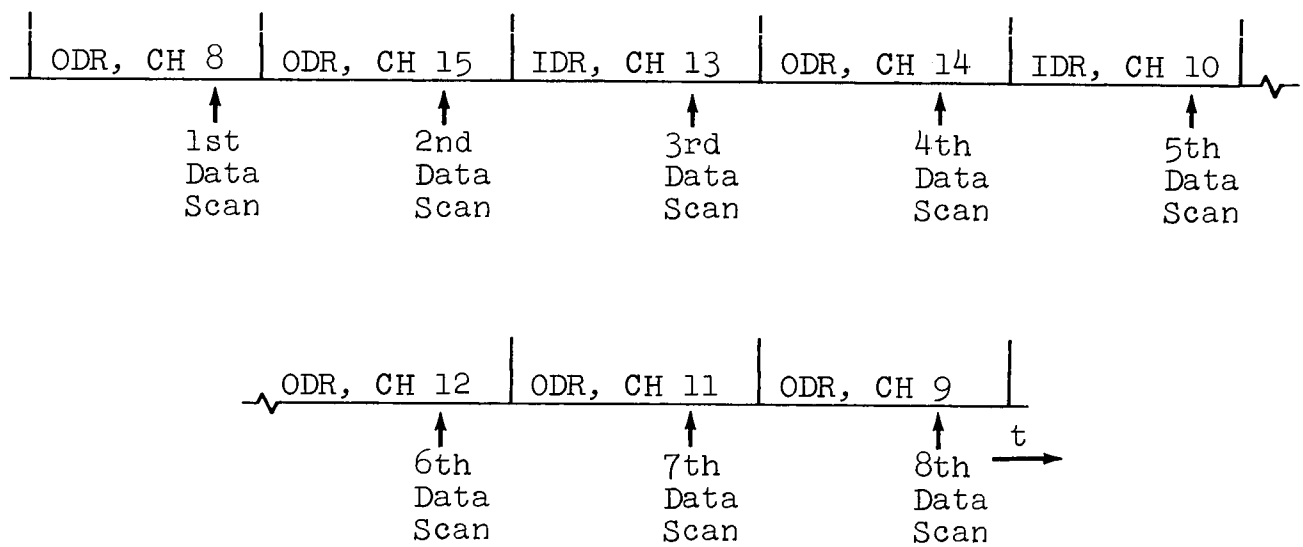
Definitions: Data Scan Priority
UNIVAC 494 Algorithm
FIGURE A1-45

Example: Normal Channel Group

Requests at first data scan:

ODR, CH 15
ODR, CH 14
IDR, CH 13
ODR, CH 12
ODR, CH 11
IDR, CH 10
ODR, CH 9

Service order by the UNIVAC 494:



Definitions: Data Scan Example
UNIVAC 494 Algorithm
FIGURE A1-46

SCS Channel:

The control unit performs a data request evaluation by examining all connected devices in a matrix fashion. Each CLT is assigned a strictly numerical priority dependent upon its position on the multiplexer. The priority scheme is CLT64 → CLT1. If a service request is found, it is presented to the computer. If no request exists, the matrix selection procedure is repeated. Note that the control unit does not distinguish between a data request and an external interrupt in its selection of a request for service.

S/S Channel:

The control unit performs a data request evaluation by moving one station in a round-robin polling scheme. If the 494/7094 adapter has a service request at this station, the control unit locks on and presents the request to the computer. If no service request exists, the control unit again steps one polling station, asking the next 494/7094 adapter if a request exists. Note that the priority selection of a request at a port is performed by the 494/7094 adapter (i.e., Input Data Request, End of Input External Interrupt, End of Output Time-out Interrupt, Output Data Request).

PBT Channel:

The control unit performs a data request evaluation by moving one station in a round-robin polling scheme. If the PBT has a service request at this station, the control unit locks on and presents the request to the computer. If no service request exists, the control unit again steps one polling station, asking the next PBT if a request exists. Note that the priority selection of a request at a port is performed by the PBT input request having priority over an output request.

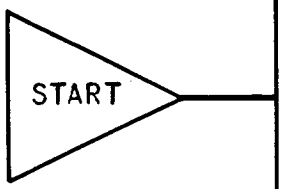
Definitions: Communications Peripheral Device Priority
UNIVAC 494 Algorithm
FIGURE A1-47

Initialization	I/O Hardware Model
Input Configuration Data: Hardware (Transfer Data), Software (Processing Data), Loading (Traffic Data).	Perform Interrupt Sca
	Perform Data Scan.
Initialize the algorithm with input data and preset variable values.	Choose and perform the next action: 1) Data Transfer, 2) Program Subinstruct

	Processing Model	Channel Operation Mode
. omputer on.	If an interrupt has been selected by the interrupt scan, set up interrupt processing. (If the current program is suspendable, suspend it and proceed to interrupt analysis processing. If the current program is nonsuspendable, perform interrupt recording processing and return to the nonsuspendable program.) Select a program subinstruction for execution and update the processing housekeeping variables. Check the function tag of the program subinstruction and perform any special operations required.	Mask the channels.

Block Diagram
UNIVAC 494 Algorithm
FIGURE A2-1

Peripheral Device Models		Decision Time Model
SCS: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	Choose the next decision time of the algorithm.
S/S: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
PBT: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
FH880: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
III C: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
RCLOK: Compute CAT. Set the next data request.	Set an interrupt.	
DCLOK: Compute CAT. Set the next data request.	Set an interrupt.	
1004: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	
CONSOLE: Compute CAT. Set the next data request. Perform a data request evaluation.	Set an interrupt.	



Set

All time variables to ∞.

All other variables to zero

Insert

T_{START}, T_{STOP}, h, k, P^*_{hk}, DF^*_k

Set for all applicable j, i, g

$T_{SE} = T_{DRC} = T_{AO_j} = T_{AI_j} = T_{CMO_g}$, $T_{CMI_g} =$

$T_{DRE_j} = T_{AC_{ji}} = T_{FH_{ji}} = T = T_{START}$

$AO_j = AI_j = CC_{ji} = 1$

Insert for all applicable j, i, g

$CHAN_j$, ΔT_{AO_j}, ΔT_{AI_j}, ΔT_{CMO_g}, ΔT_{CMI_g}, $\Delta T_{}$

ΔT_{CL_j}, ΔT_{D_j}, $TYPE_j$, $TYPE_{ji}$, DIR_{ji}, $T_{REF_{ji}}$,

IE_{ji}, $CNOM_{ji}$, $CMAX_{ji}$, $\Delta T_{DR_{ji}}$, $\Delta T_{DR_{ji}CC_{ji}}$,

$\Delta T_{B_{ji}}$, $\Delta T_{FDR_{ji}}$, $\Delta T_{LATENCY_{ji}}$, $\Delta T_{DA_{ji}}$, Δ

$\Delta T_{CU_{ji}}$, $\Delta T_{EI_{ji}}$

Insert for all applicable A segments, α segments,

$\{A_{hk}\}$, $MAX_\alpha A_{hk}$, $\{\alpha_{hk}\}$, $MAX_\beta \alpha_{hk}$, $\{\beta_{hk}\}$,

Initializat

10-1

Insert buffer group names for all applicable buffer groups.

Insert queue names for all applicable queues, including

QPTG0, QPTGS1, QPTG1, QPTGS2, QPTG2, QPTGS3, QPTG3,

$Q_{DF_{ji}}$, $QINEXT_j$, $QINMON_j$, $QOUTMON_j$, $QOUTEXT_j$, QRCLOK, QDCLOK

Insert for all applicable queues, queue groups, queue super groups

$QGRP_Q$, $QSGRP_{QGRP}$

Insert data frame information in $Q_{DF_{ji}}$ for all applicable j, i.

Set for all applicable j, i

$DFINFO_{ji}$ = QDF* from $Q_{DF_{ji}}$

RE_{ji} ' Remove the entry from $Q_{DF_{ji}}$

Set for all applicable j, i

$T_{DR_{ji}} = T_{FS_{ji}} + T_{REF_{ji}}$

$T_{FDR_{ji}} = T_{DR_{ji}}$ (PBT channel)

FH_{ji}' $PBC_{ji} = 1$ (PBT channel)

$T_{BAR_{ji}} = T_{DR_{ji}} + 600\ \Delta T_{B_{ji}}$ (PBT channel output device)

β segments $T_{BAR_{ji}} = T_{DR_{ji}} + 1200\ \Delta T_{B_{ji}}$ (PBT Channel input device)

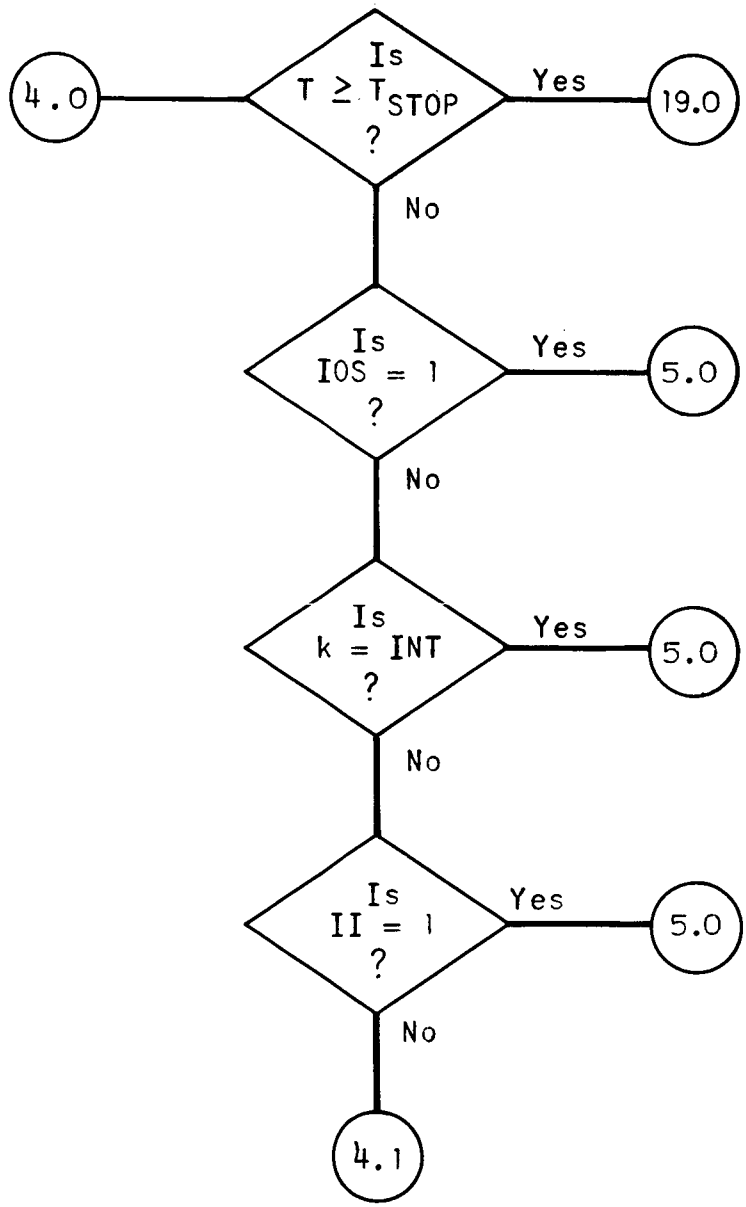
MAX $S_{\beta_{hk}}$ $FDR_{ji} = 1$ (output device except LSOUT)

7.0

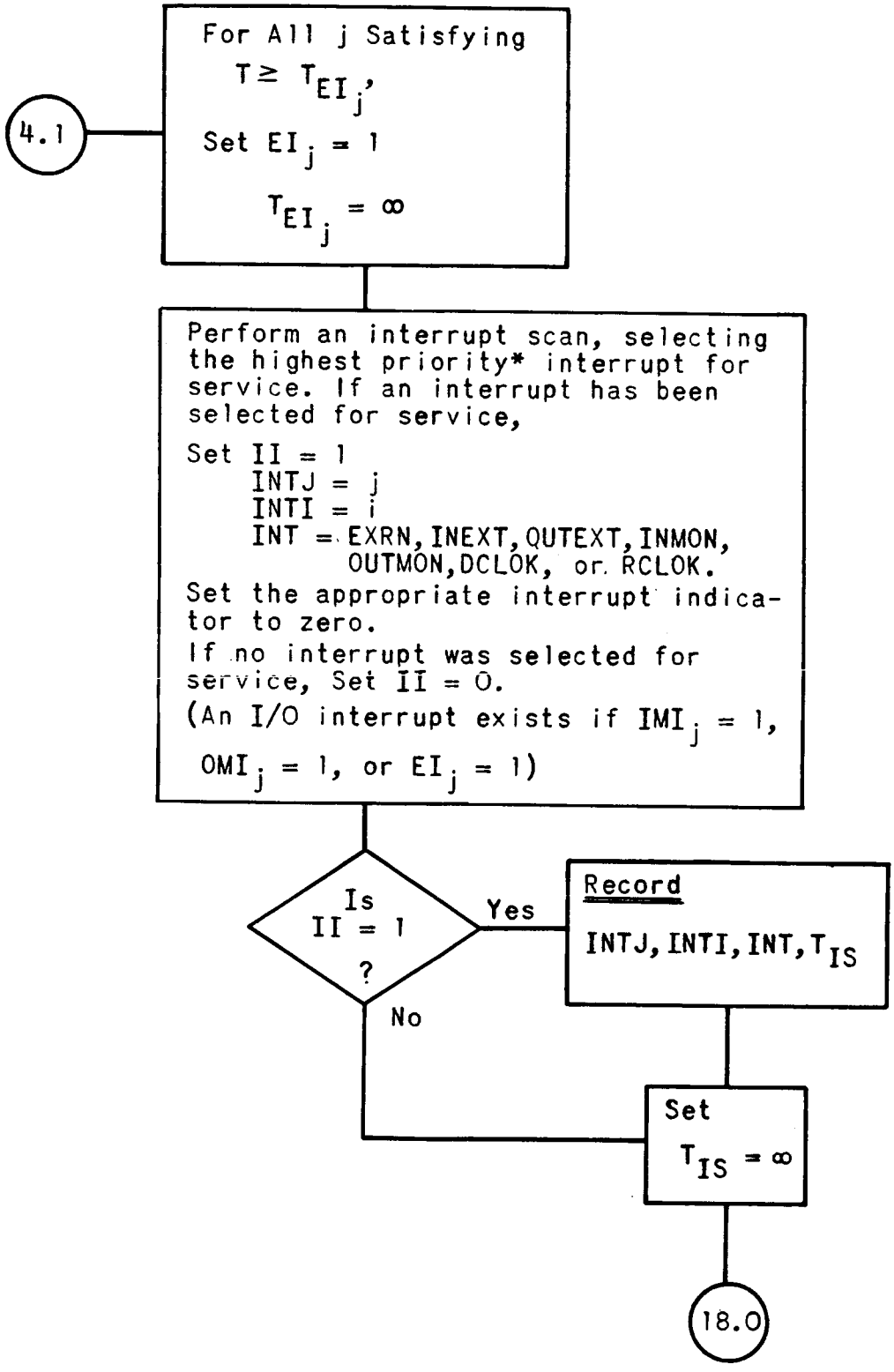
ion: Input the required data and initialize.

UNIVAC 494 Algorithm

FIGURE A3-1

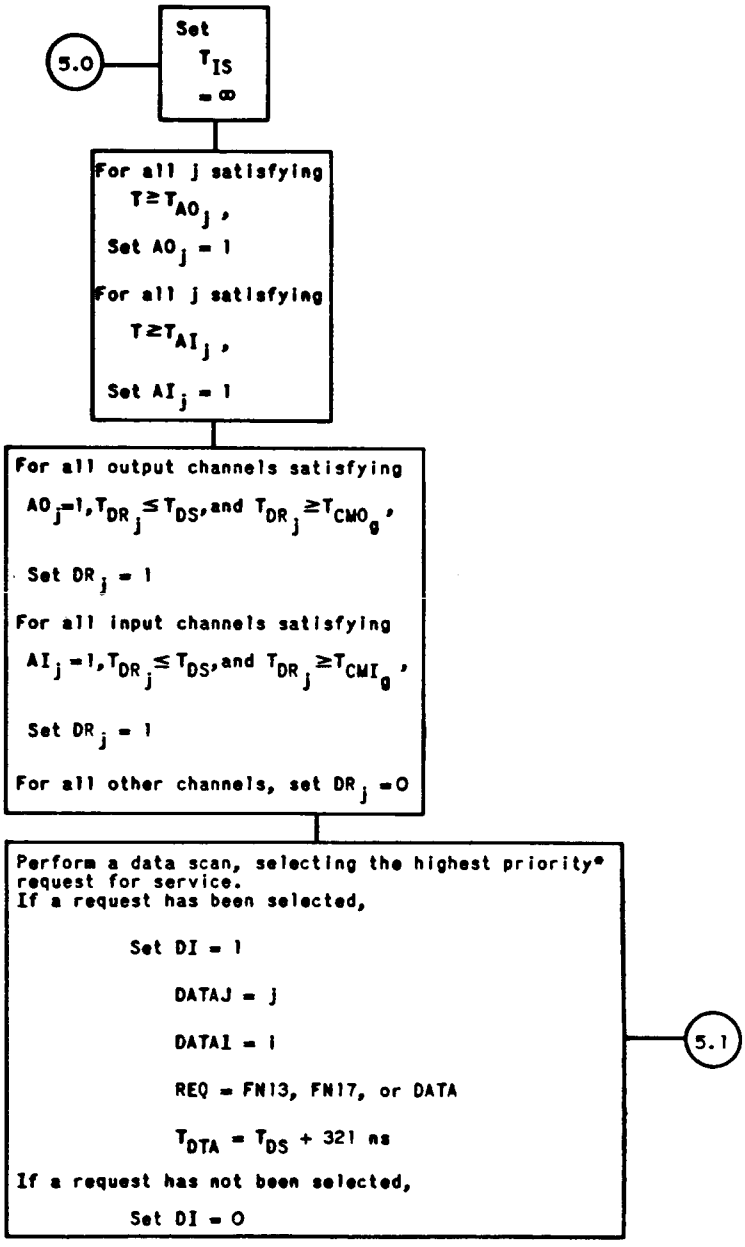


Interrupt Scan: Determine if an interrupt scan is allowed.
UNIVAC 494 Algorithm
FIGURE A4-1



Interrupt Scan: Perform an Interrupt Scan.
UNIVAC 494 Algorithm
FIGURE A4-2

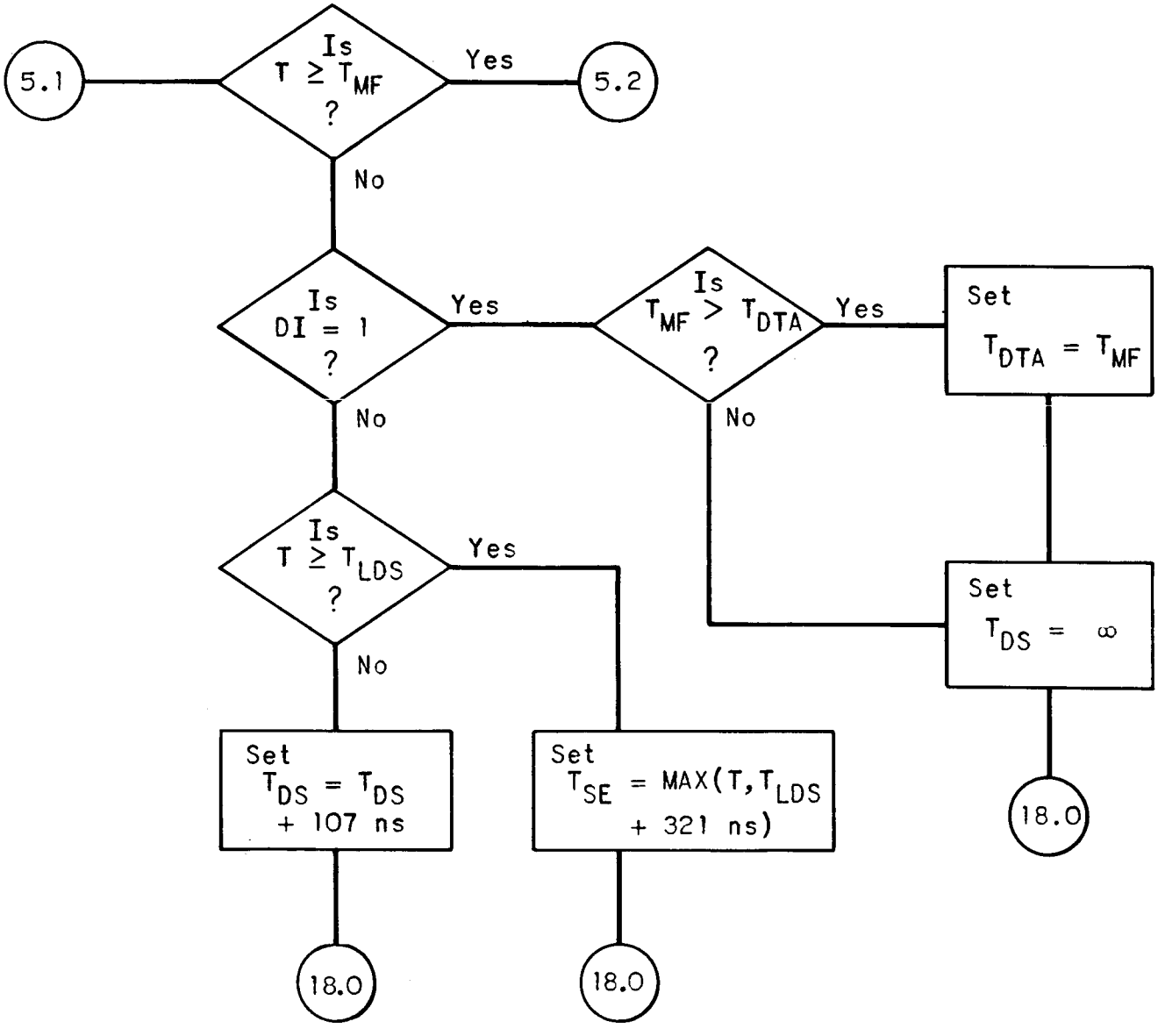
*See FIGURE A1-44



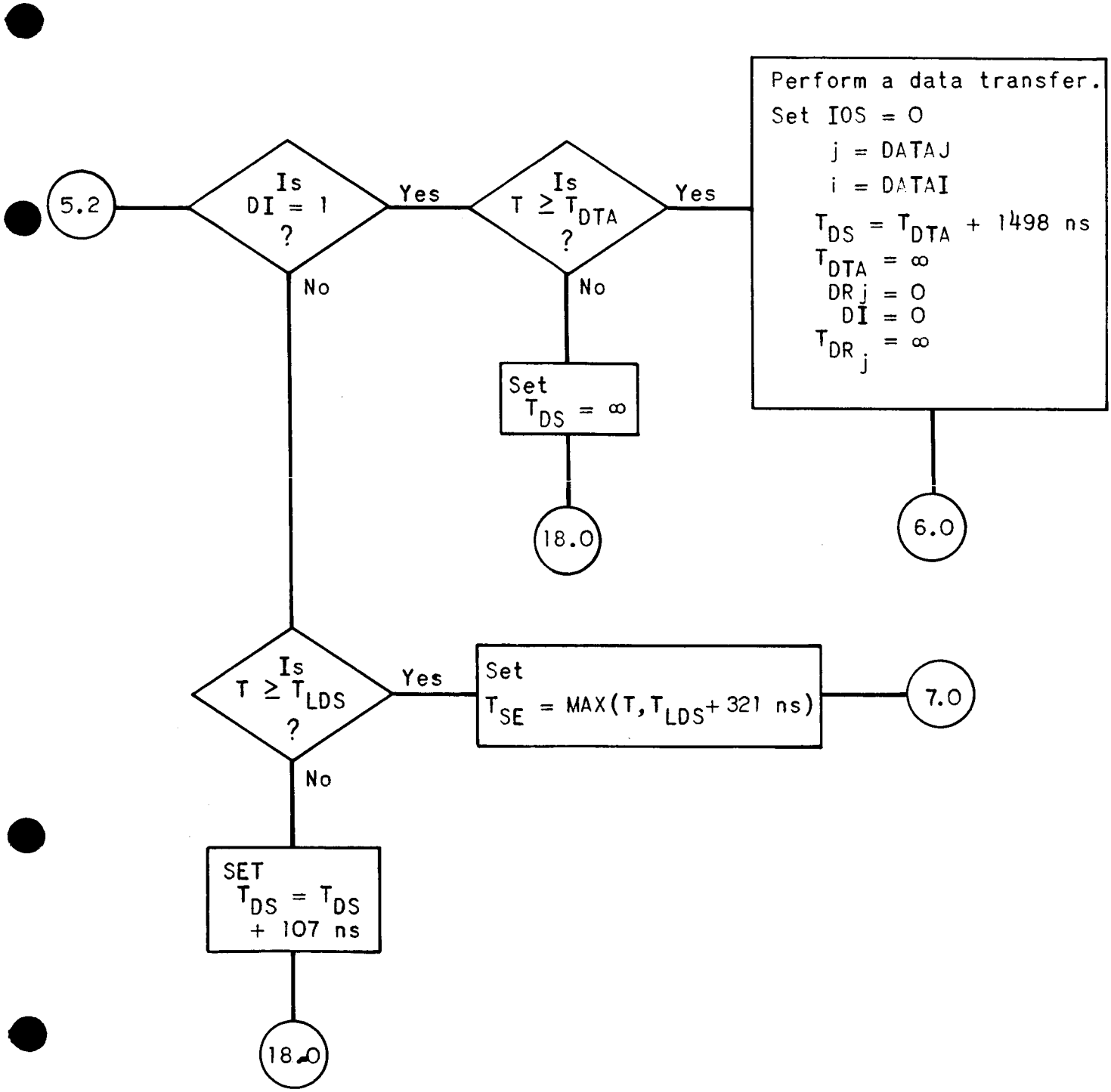
Data Scan: Perform a Data Scan.
UNIVAC 494 Algorithm

*See FIGURE A1-45

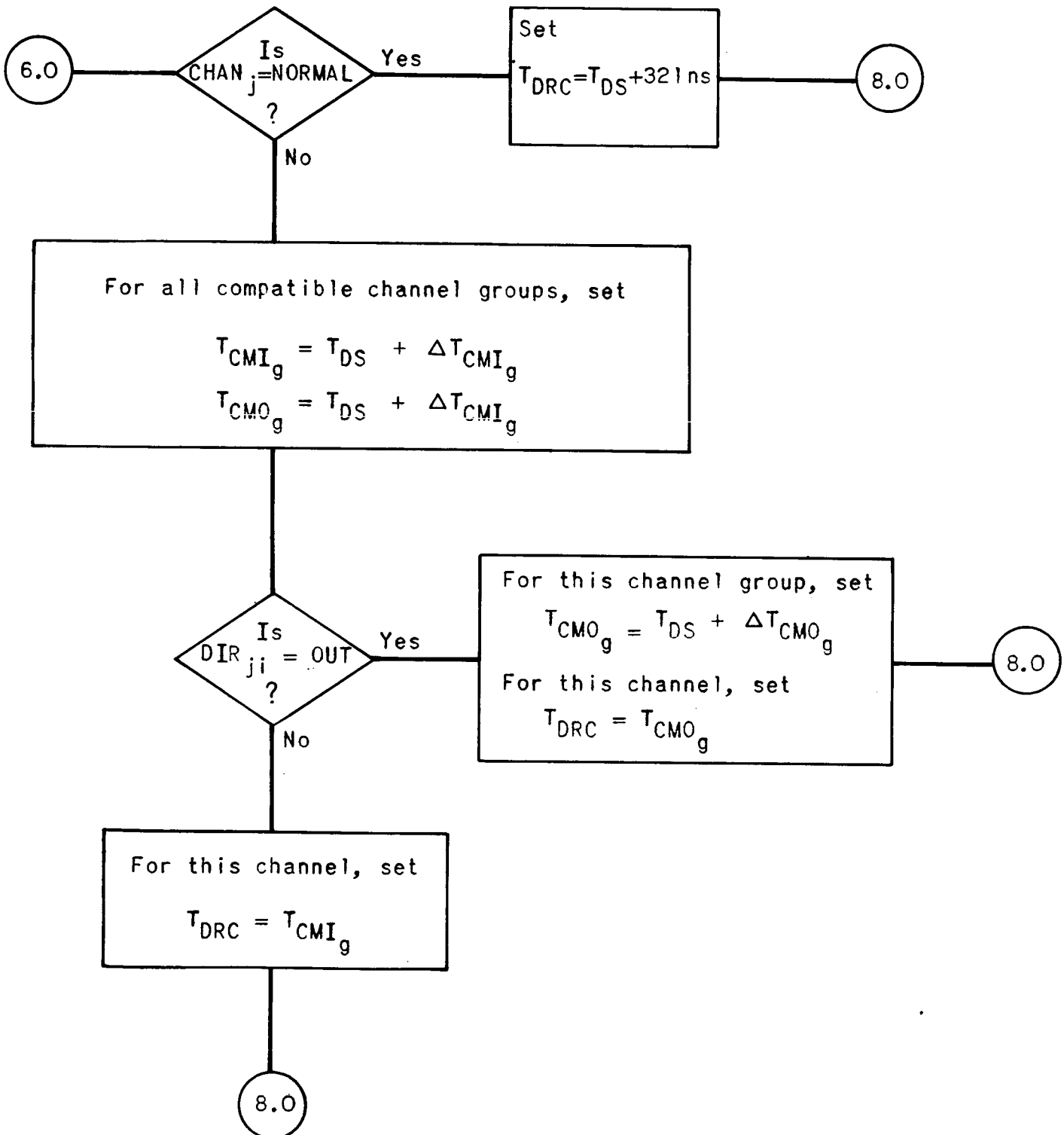
FIGURE A5-1



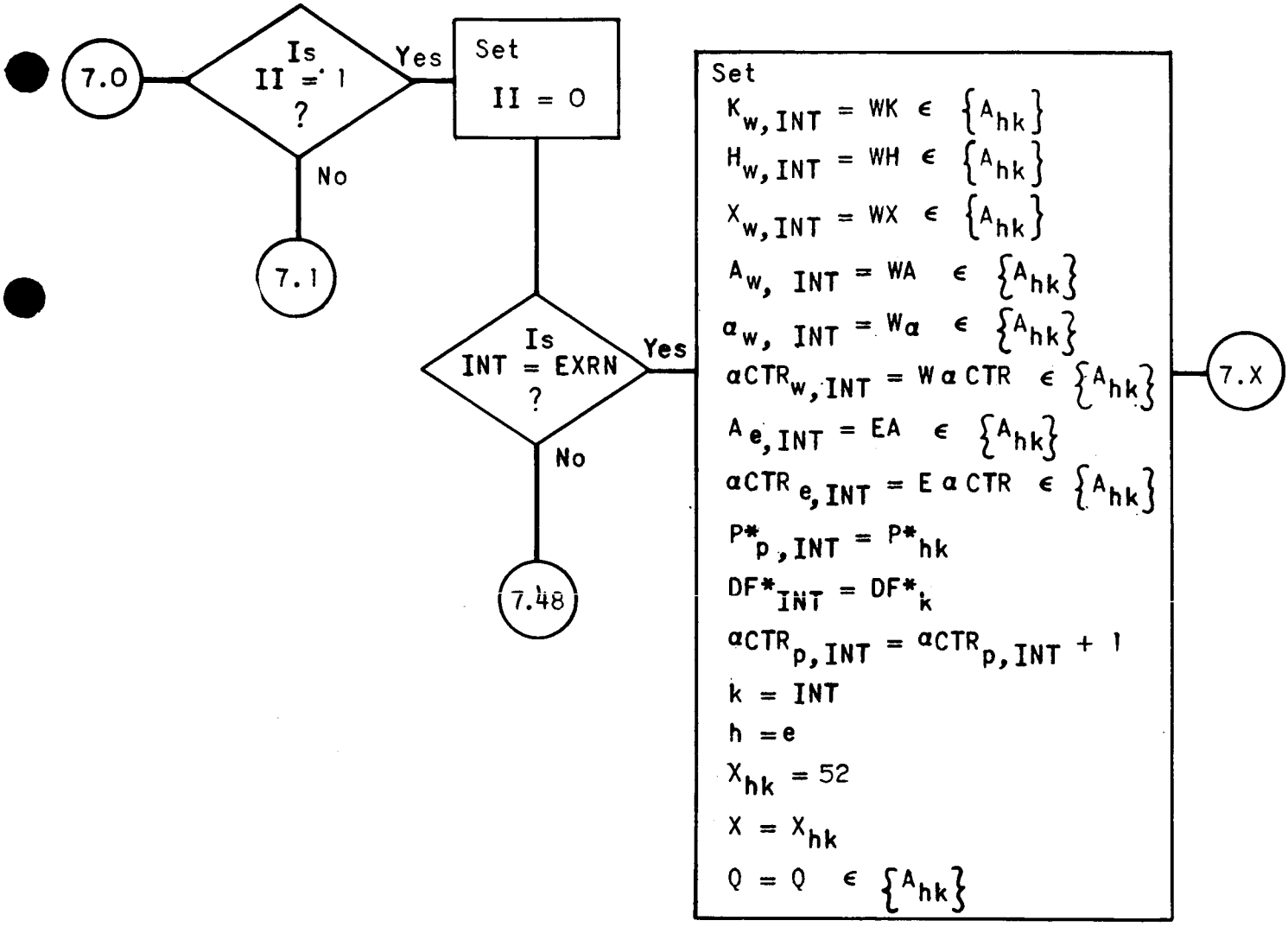
Data Scan: Set a Data Scan.
UNIVAC 494 Algorithm
FIGURE A5-2



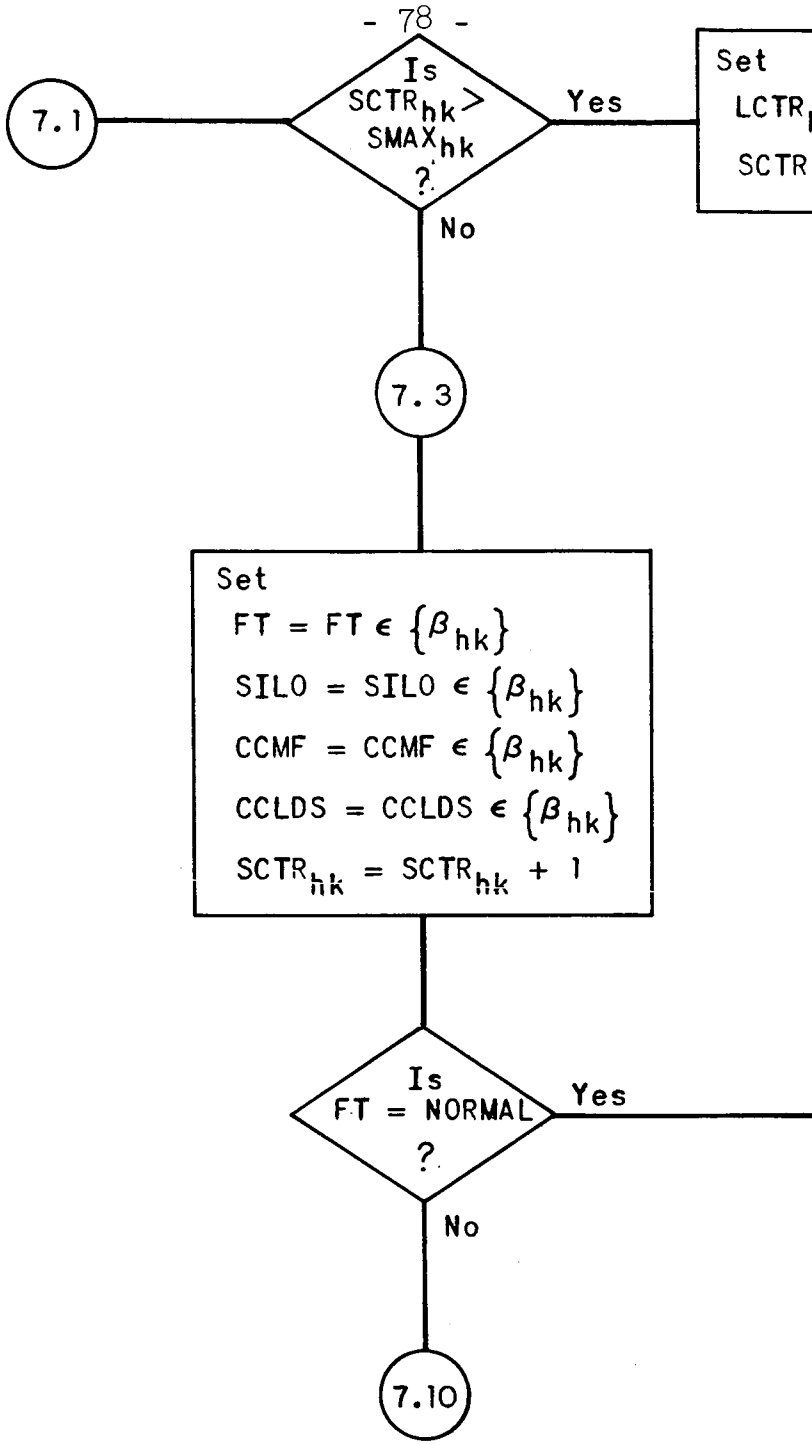
Data Scan: Perform a data transfer.
UNIVAC 494 Algorithm
FIGURE A5-3



I/O Lockout: Mask the Channels.
UNIVAC 494 Algorithm
FIGURE A6-1

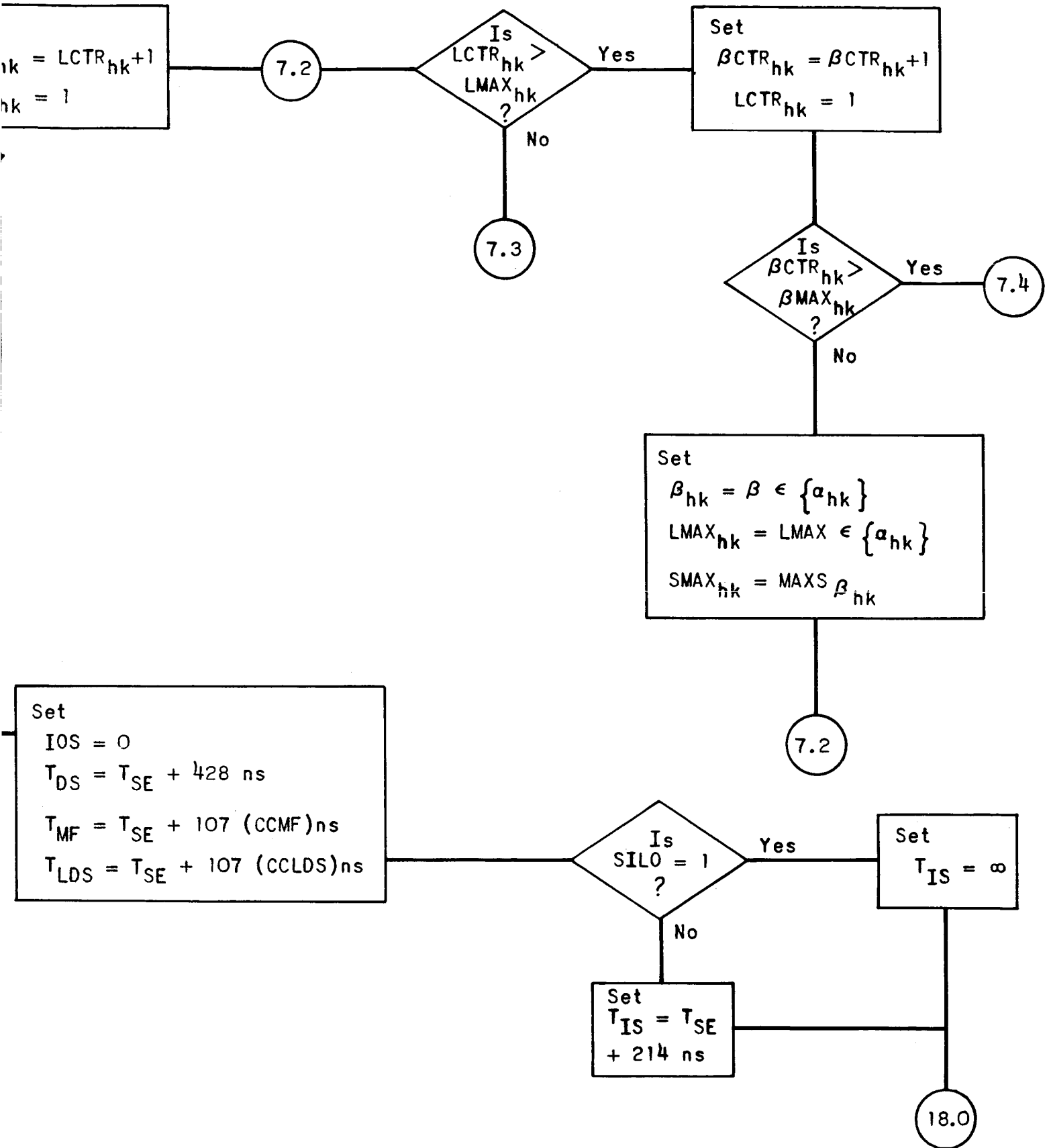


Processing Housekeeping:
Set up the processing for EXRN interrupt.
UNIVAC 494 Algorithm
FIGURE A7-1


7.1
Is SCTR$_{hk}$ > SMAX$_{hk}$?
Yes
Set
LCTR
SCTR
No
7.3
Set
FT = FT ϵ $\{\beta_{hk}\}$
SILO = SILO ϵ $\{\beta_{hk}\}$
CCMF = CCMF ϵ $\{\beta_{hk}\}$
CCLDS = CCLDS ϵ $\{\beta_{hk}\}$
SCTR$_{hk}$ = SCTR$_{hk}$ + 1
Is FT = NORMAL ?
Yes
No
7.10

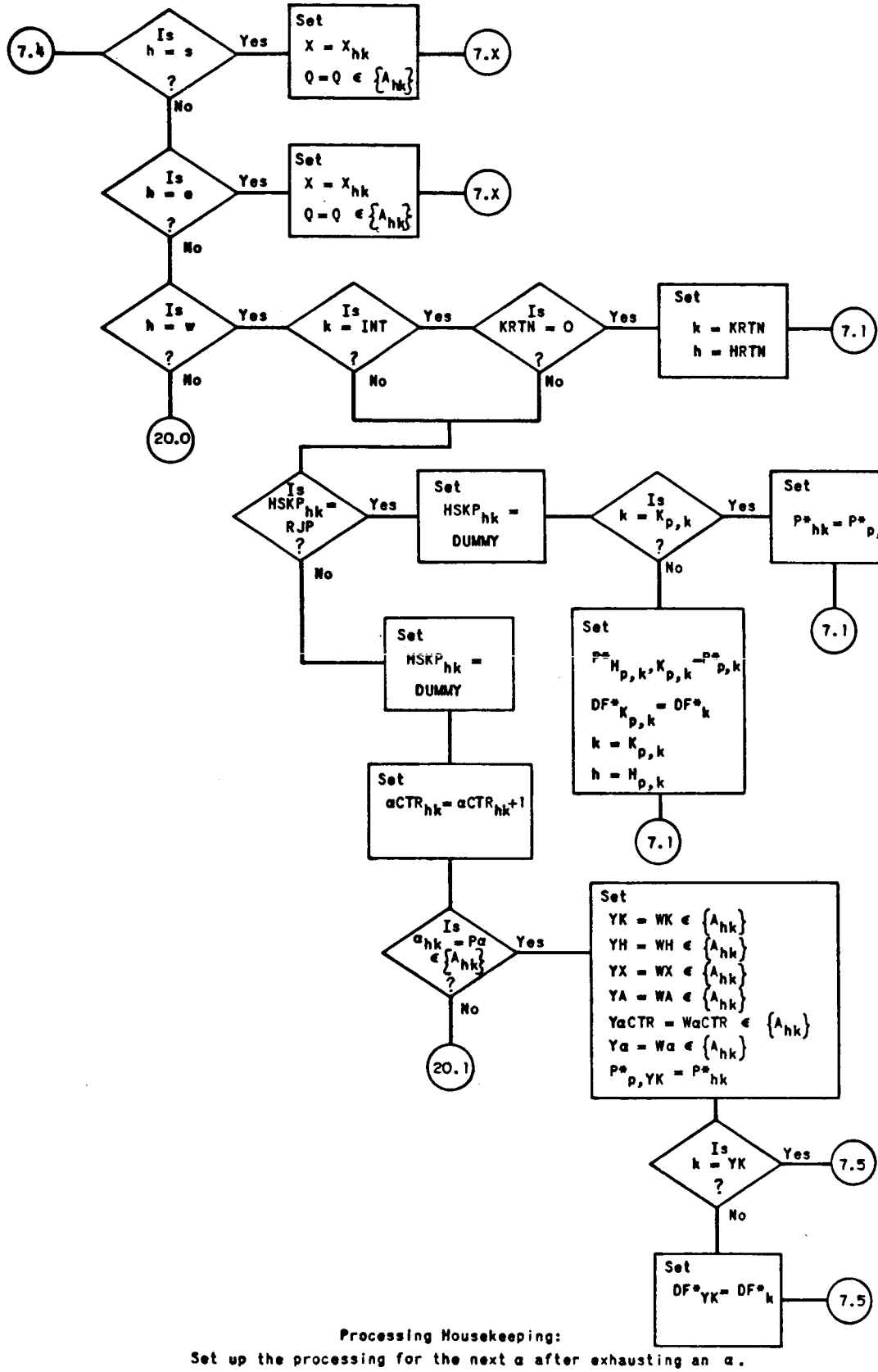
Proce

78-1

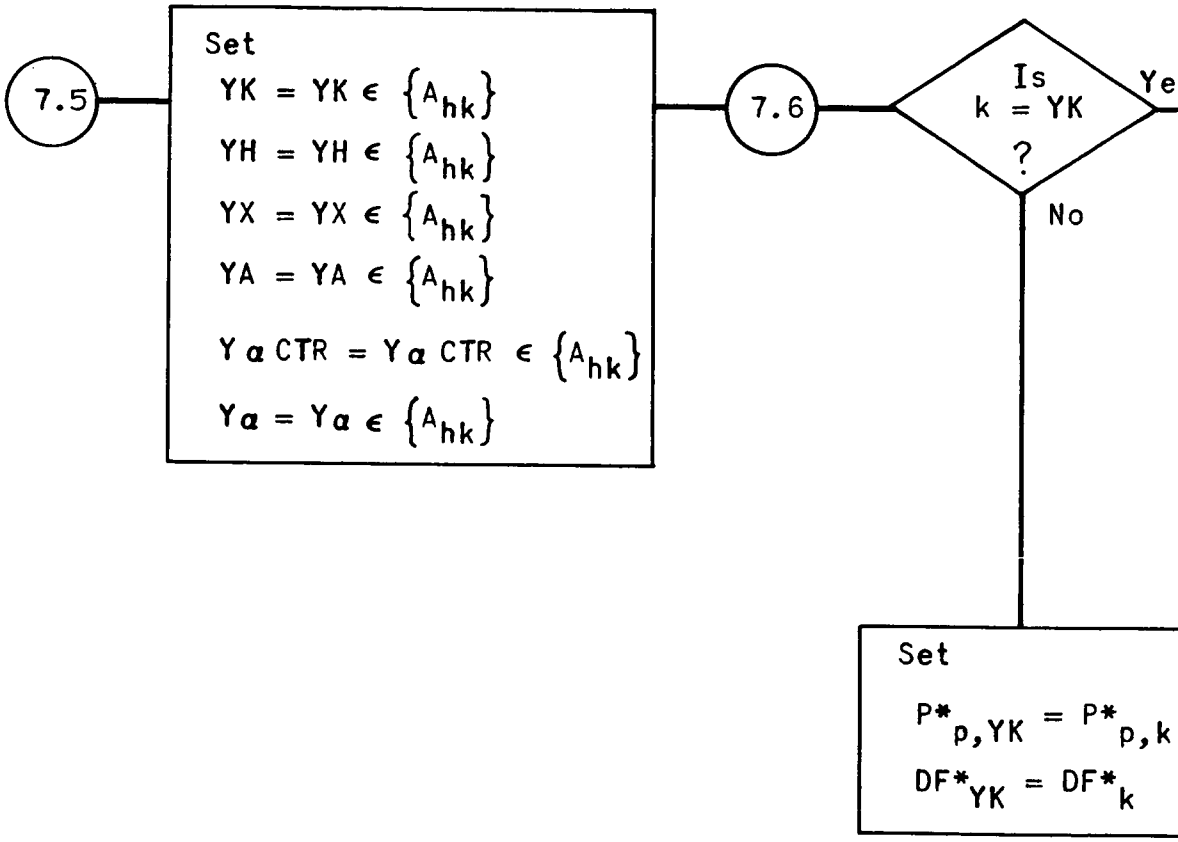


ssing Housekeeping: Pick the next subinstruction.

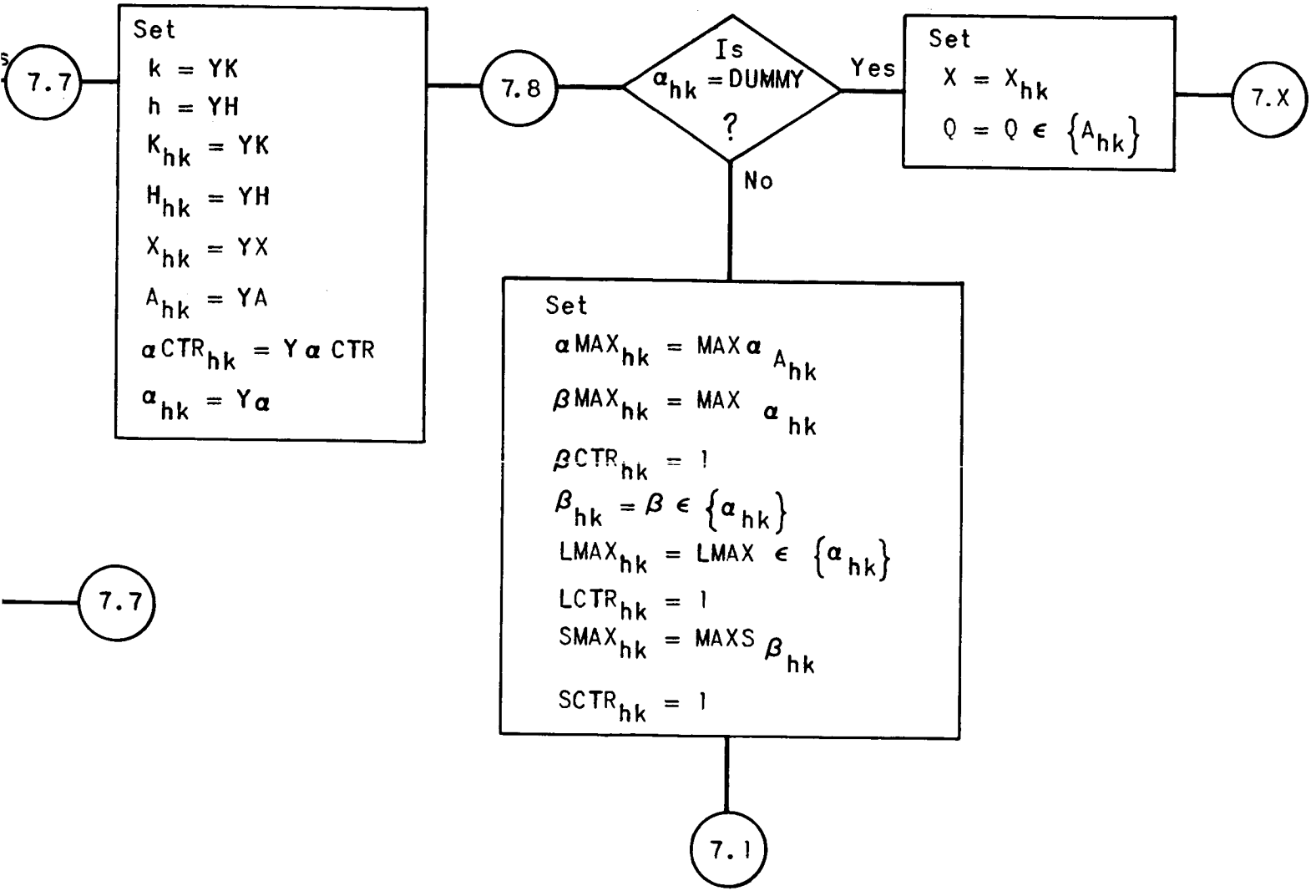
UNIVAC 494 Algorithm
FIGURE A7-2



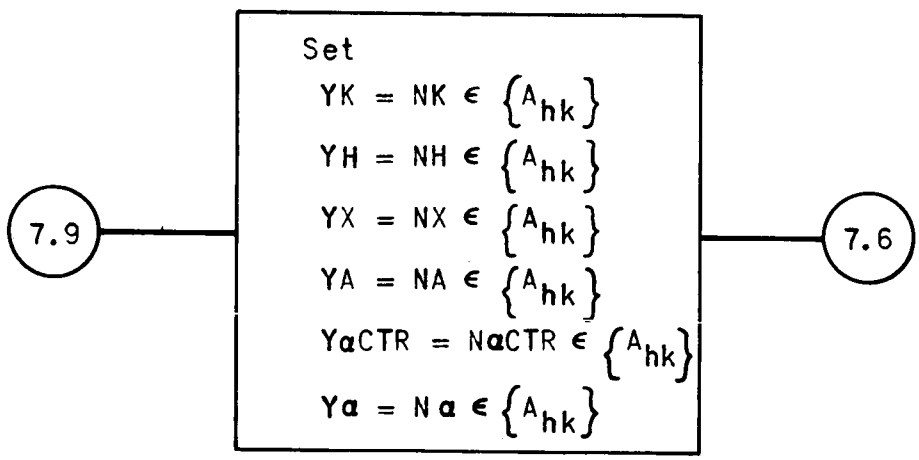
Processing Housekeeping:
Set up the processing for the next α after exhausting an α.
UNIVAC 494 Algorithm
FIGURE A7-3



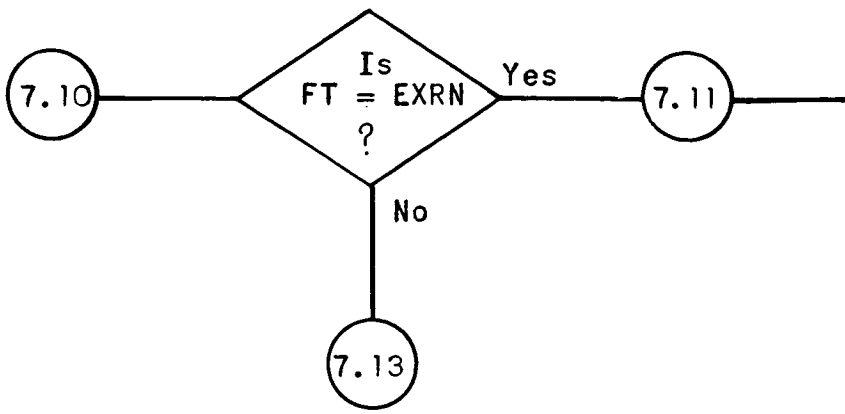
Processing Housekeeping: Set up ho
UNIVA
F

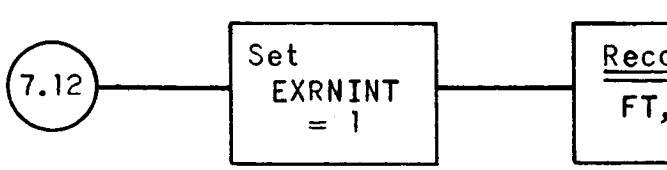


usekeeping variables after selection of an α.
: 494 Algorithm
IGURE A7-4

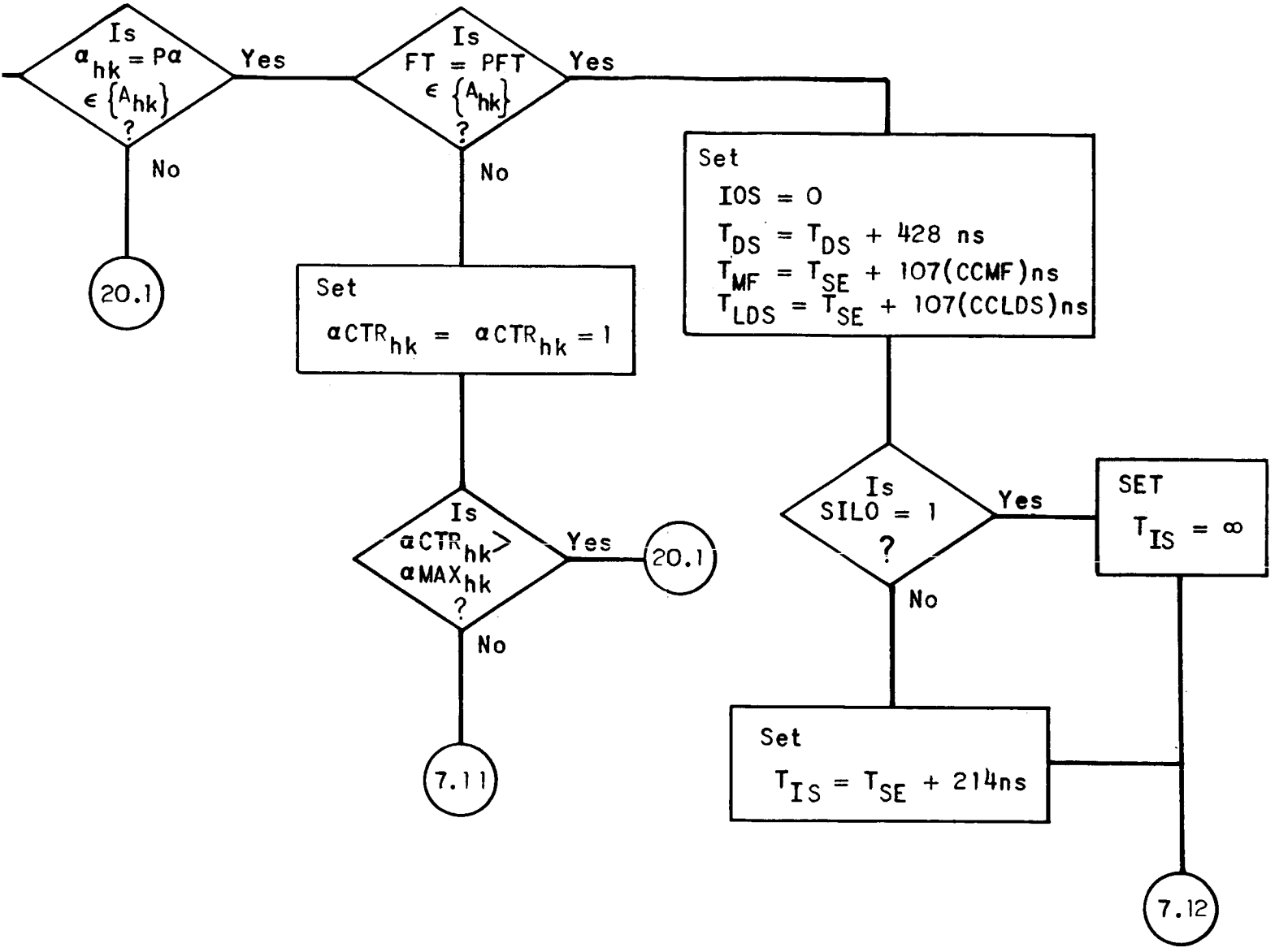


Processing Housekeeping:
Prepare to set up housekeeping variables after selection of an α.
UNIVAC 494 Algorithm
FIGURE A7-5





Process



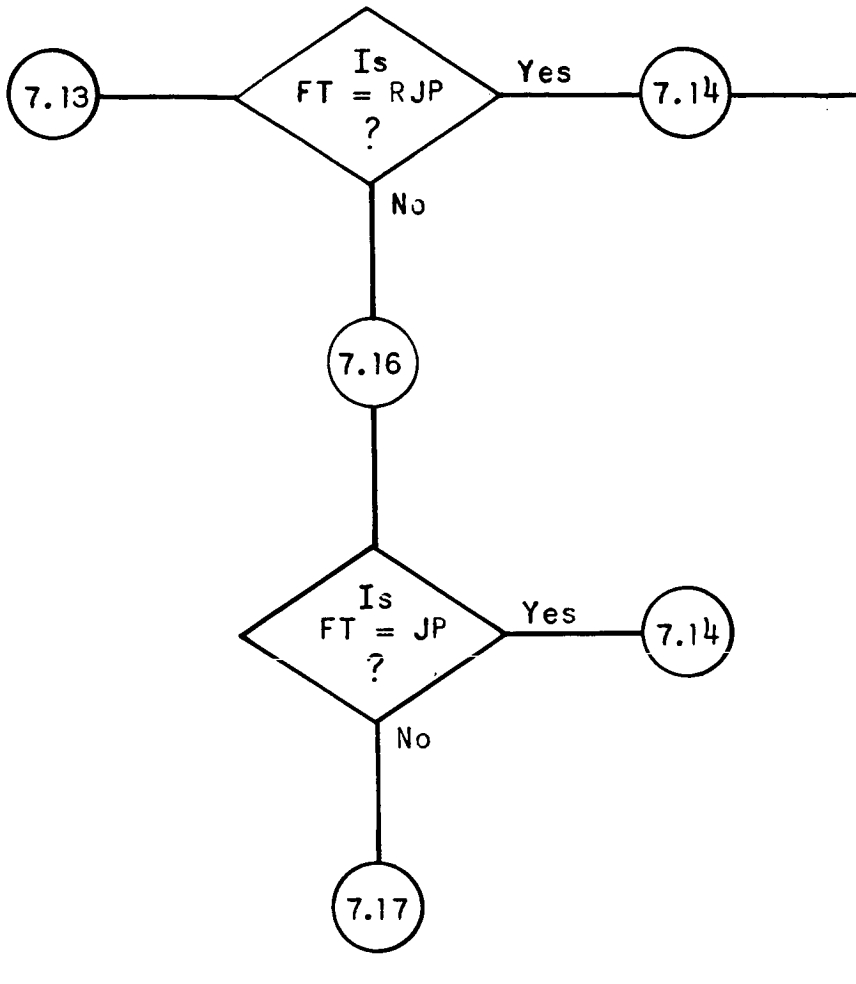
rd
T_{SE} , PID_{hk} , $DFID_k$

18.0

ng Housekeeping: Set an EXRN interrupt.

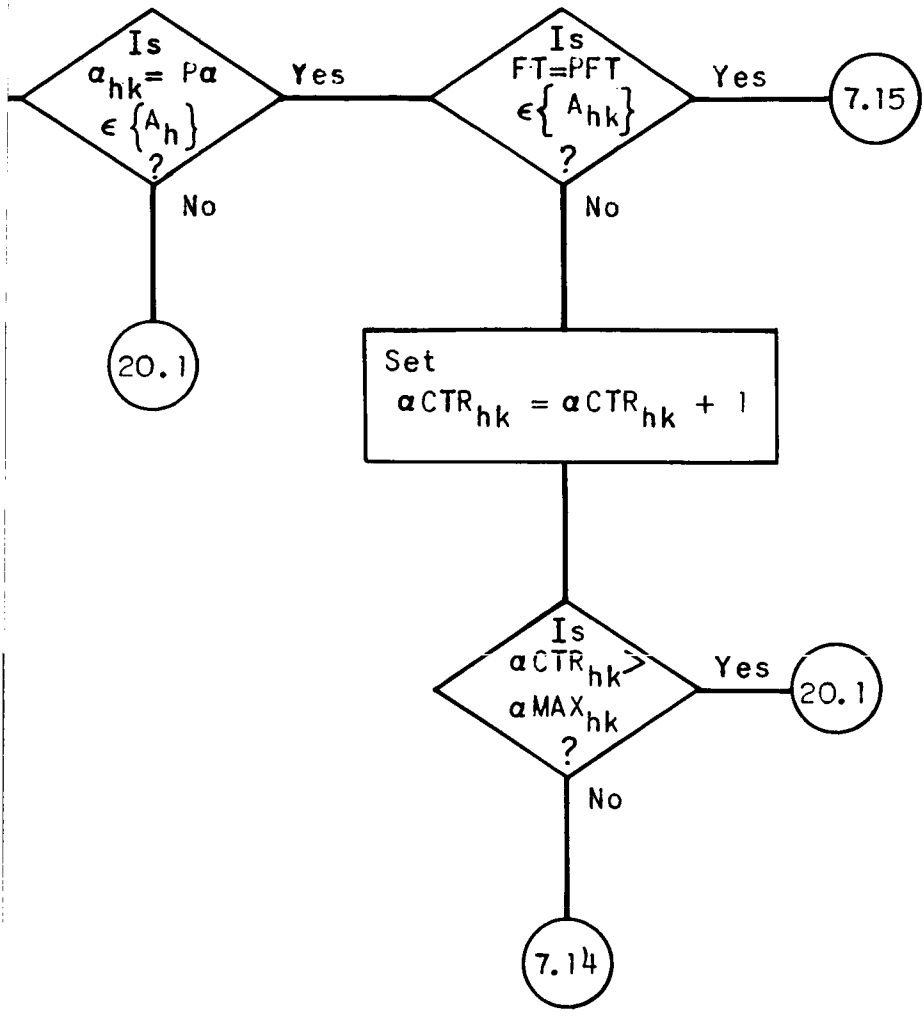
UNIVAC 494 Algorithm

FIGURE A7-6



Proc

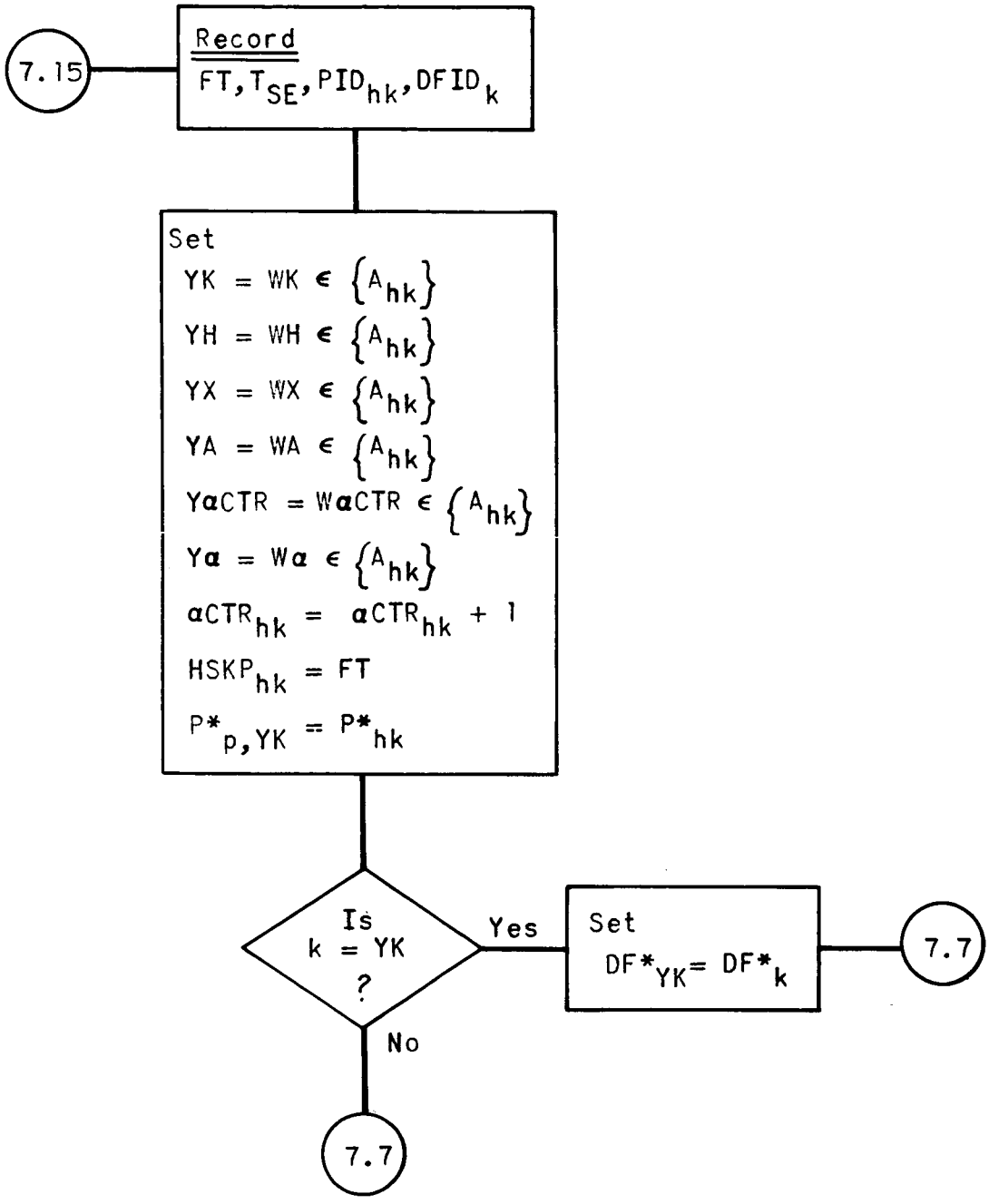
83-1



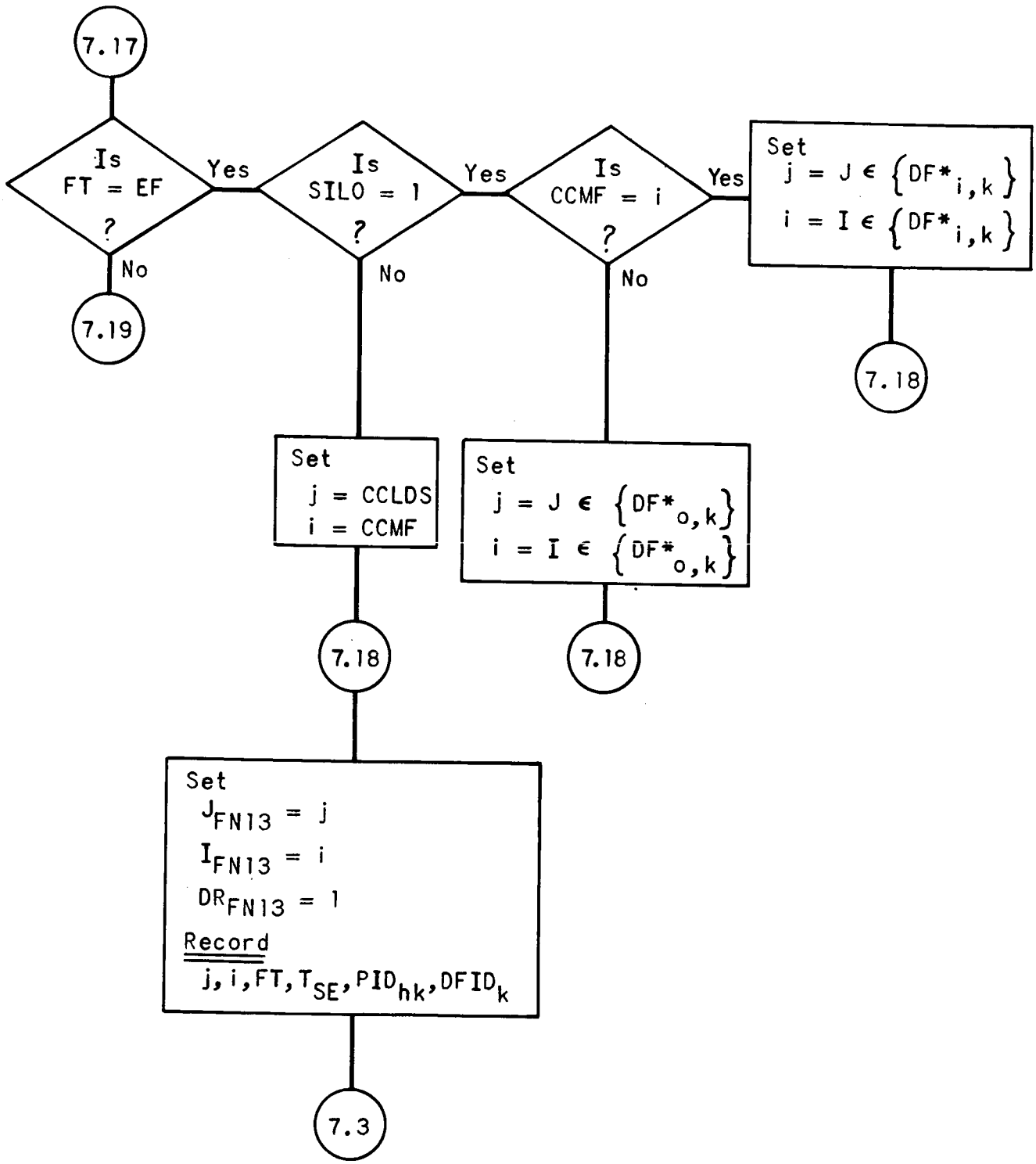
:essing Housekeeping: Set up a JP or RJP to an α.

UNIVAC 494 Algorithm

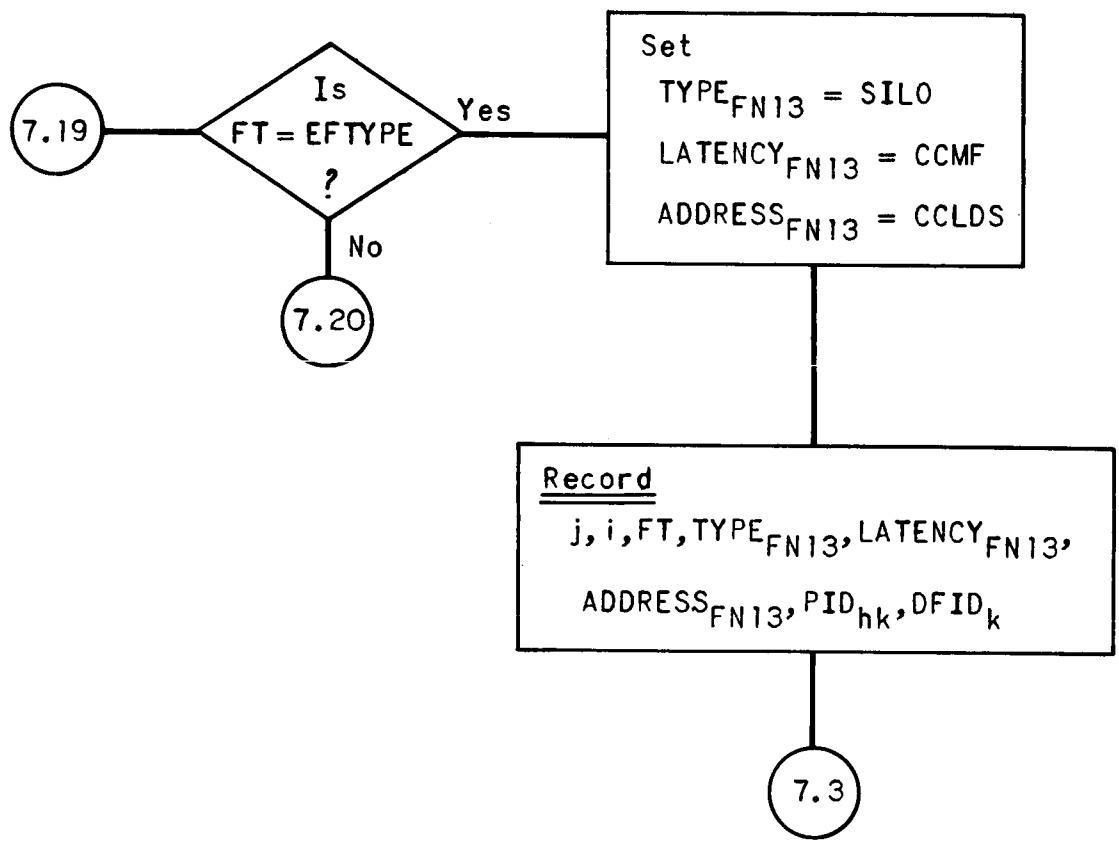
FIGURE A7-7



Processing Housekeeping:
Prepare to set up housekeeping variables
after selection of an α by a jump or return jump.
UNIVAC 494 Algorithm
FIGURE A7-8



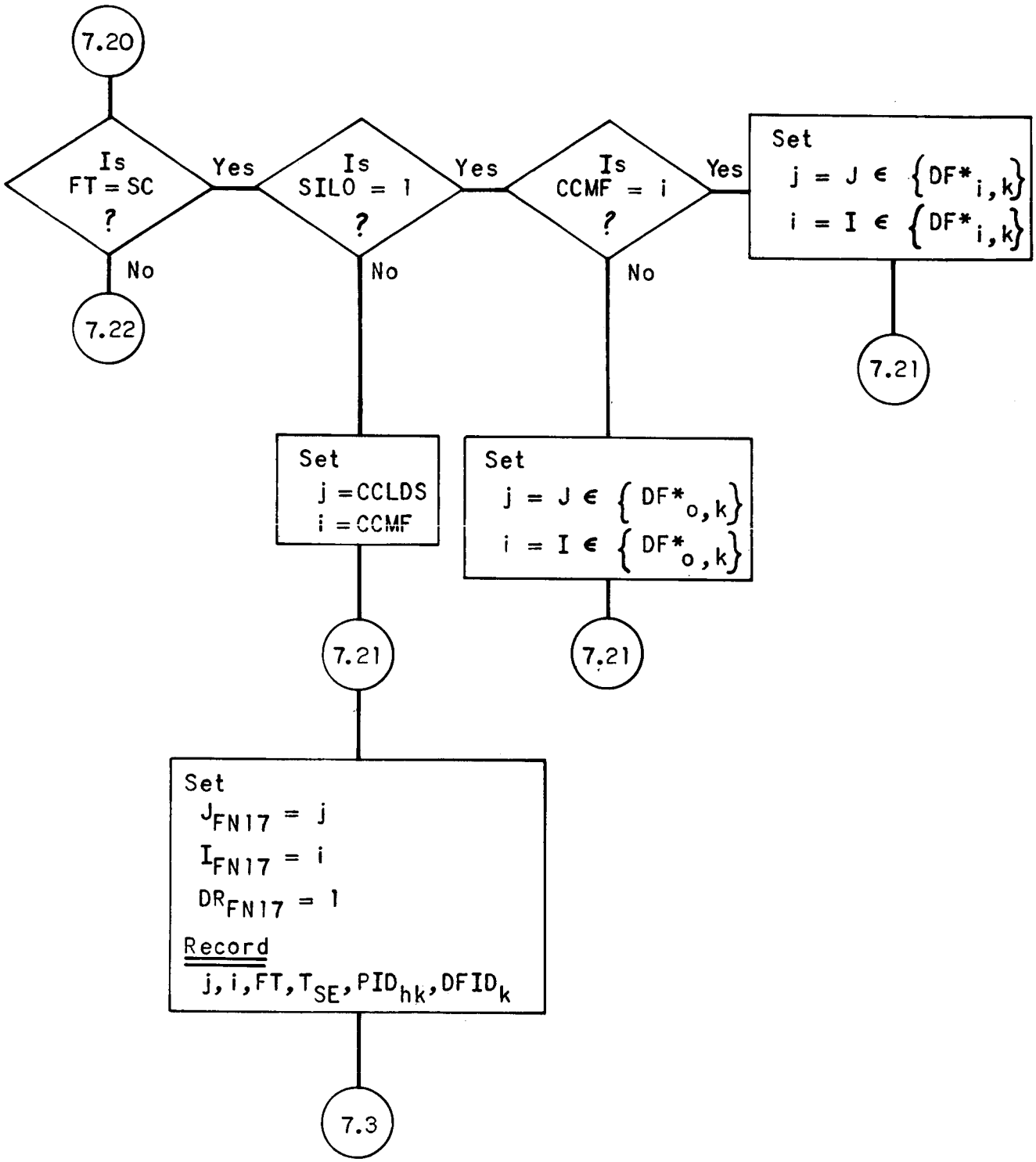
Processing Housekeeping: Set a FN13 data request.
UNIVAC 494 Algorithm
FIGURE A7-9



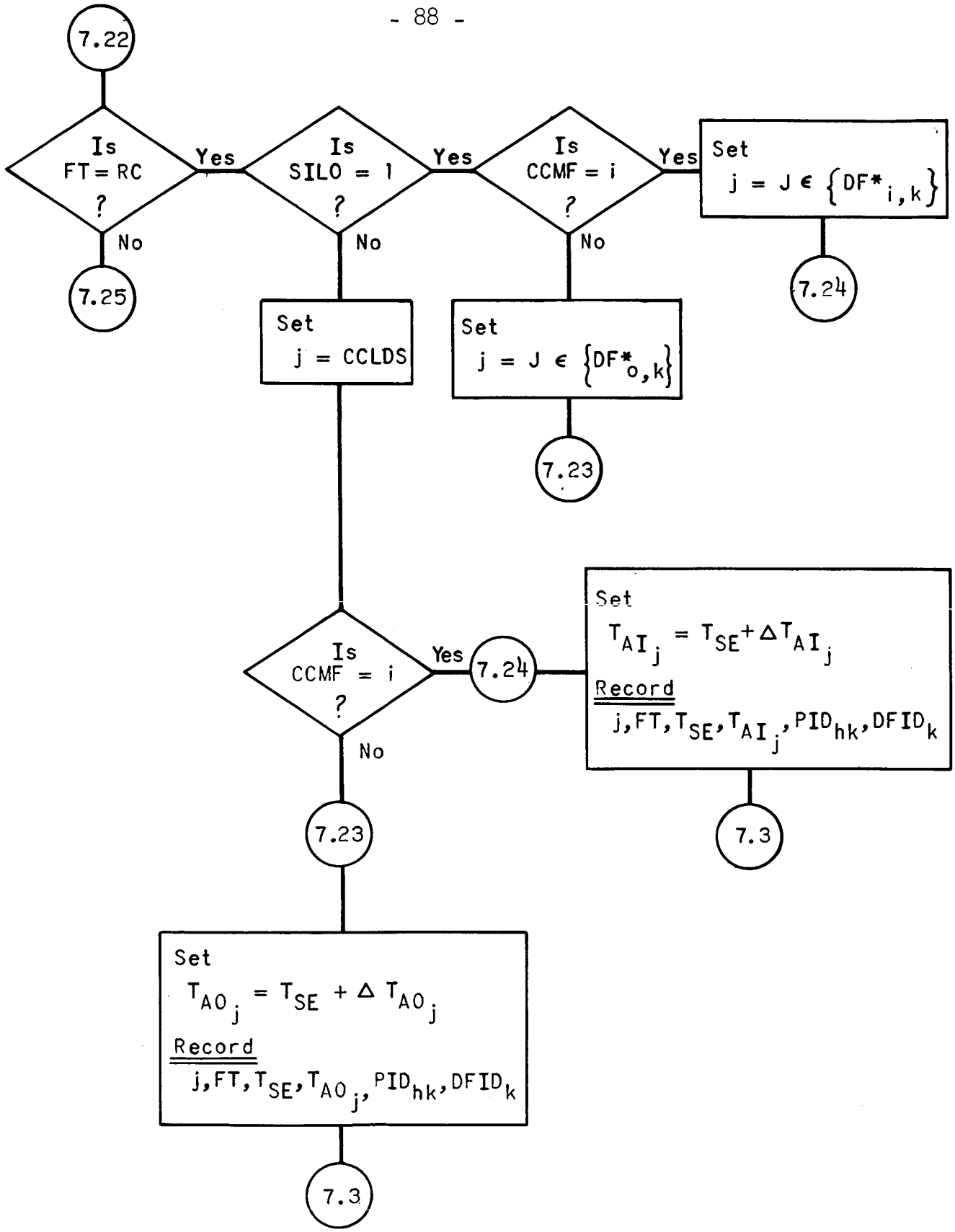
Processing Housekeeping:
Set additional variables for a FN13 action
by an addressable peripheral device.

UNIVAC 494 Algorithm

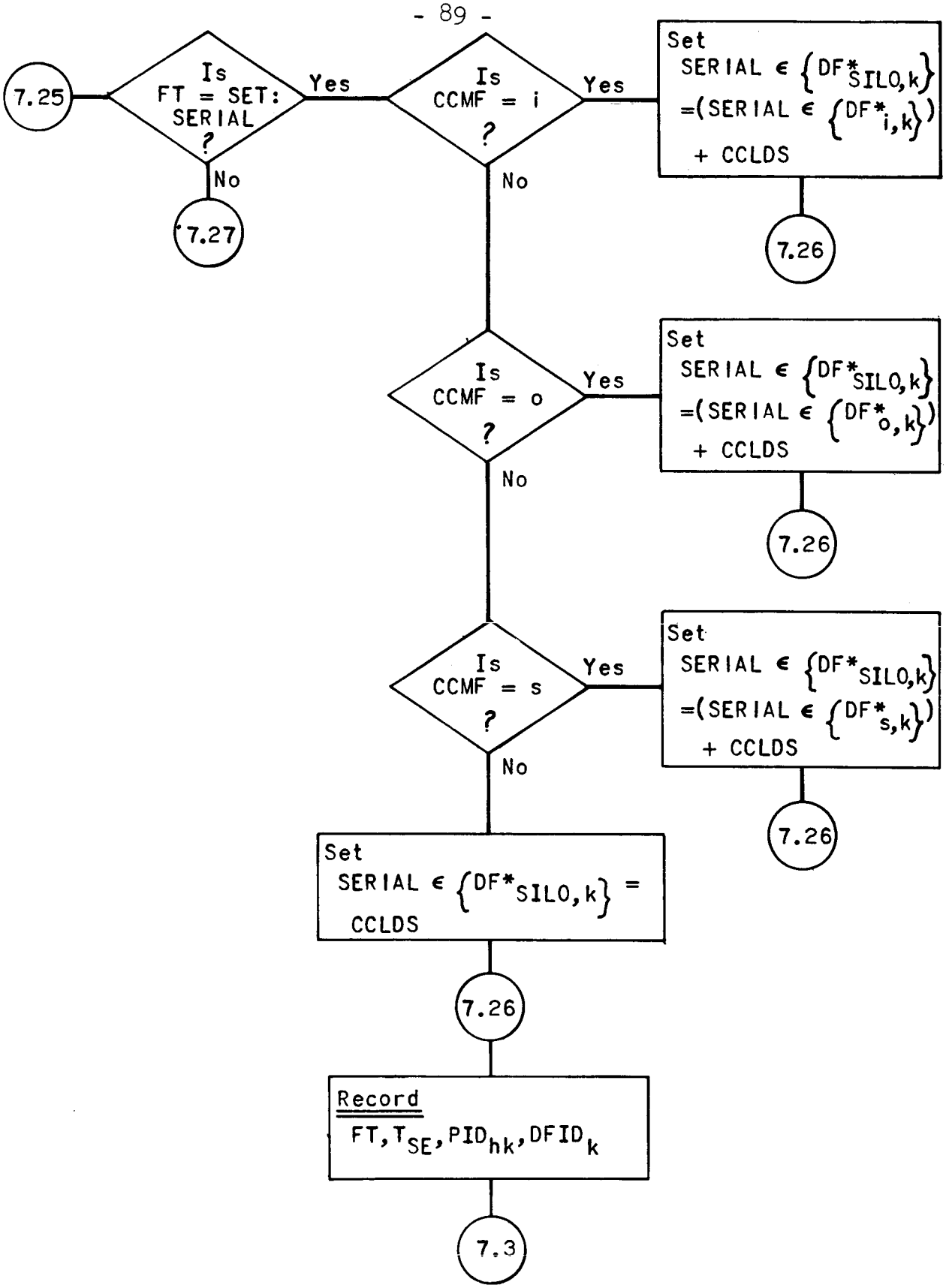
FIGURE A7-10



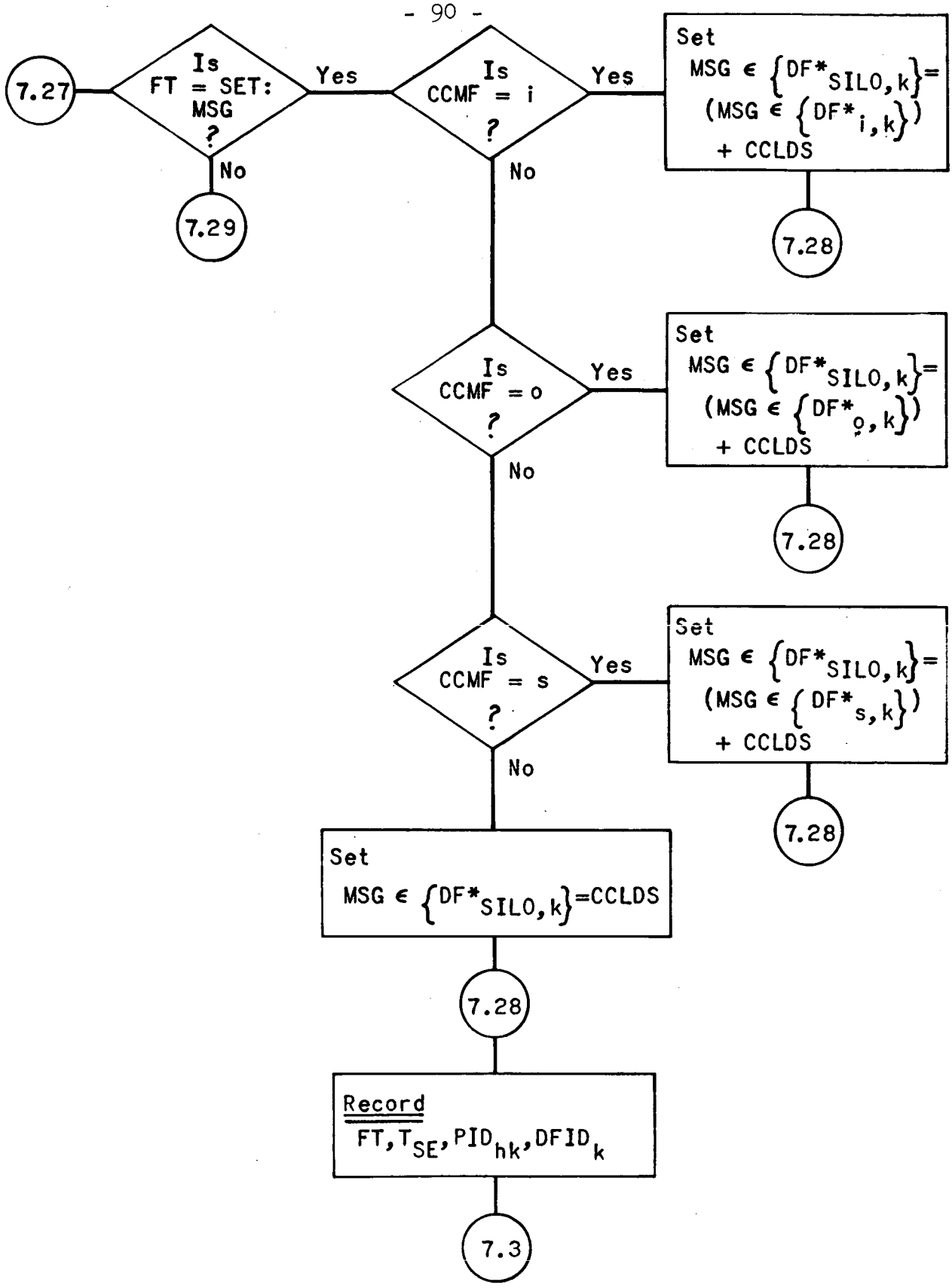
Processing Housekeeping: Set a FN17 data request.
UNIVAC 494 Algorithm
FIGURE A7-11



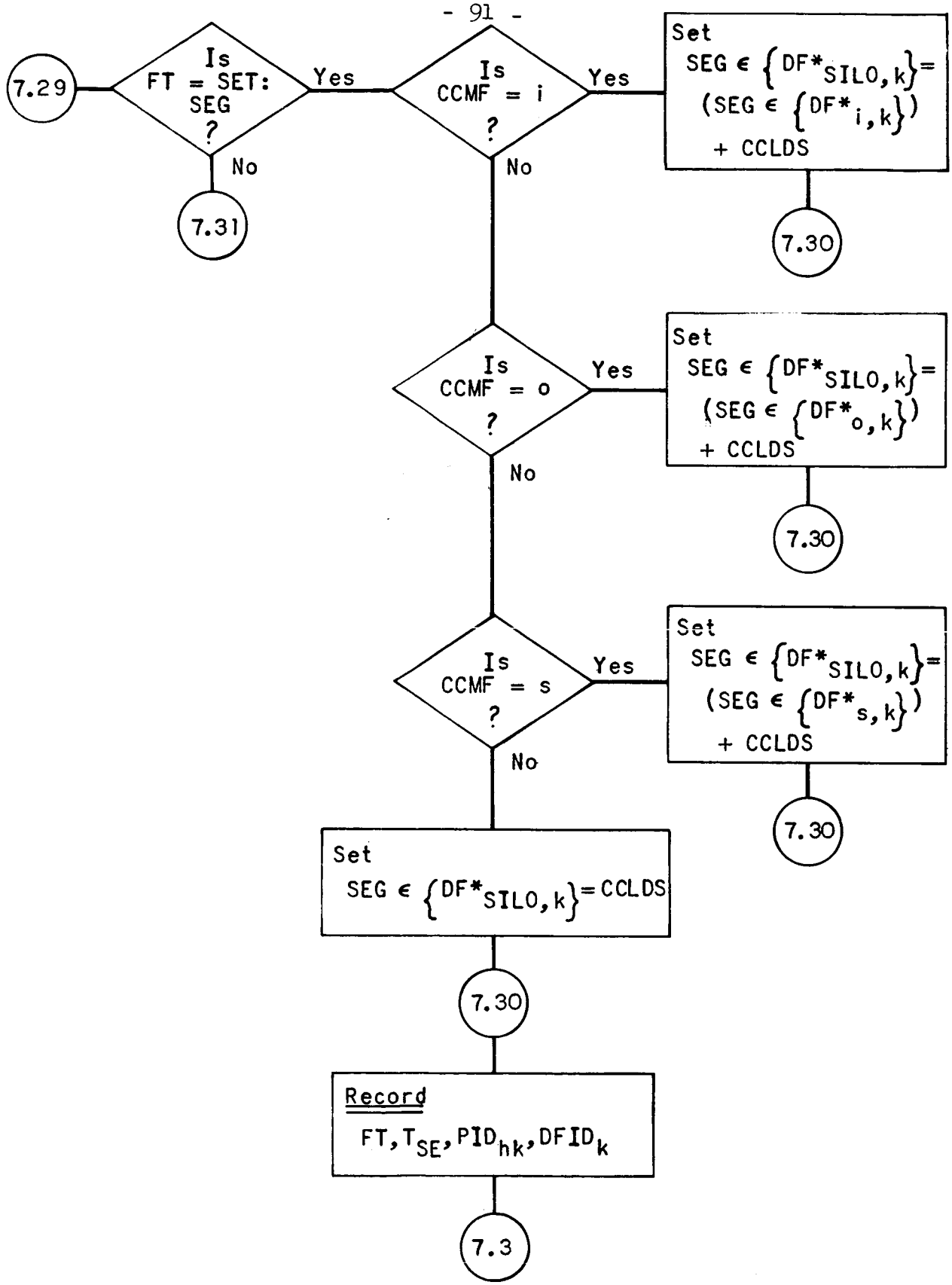
Processing Housekeeping: Reinitiate an I/O channel.
UNIVAC 494 Algorithm
FIGURE A7-12



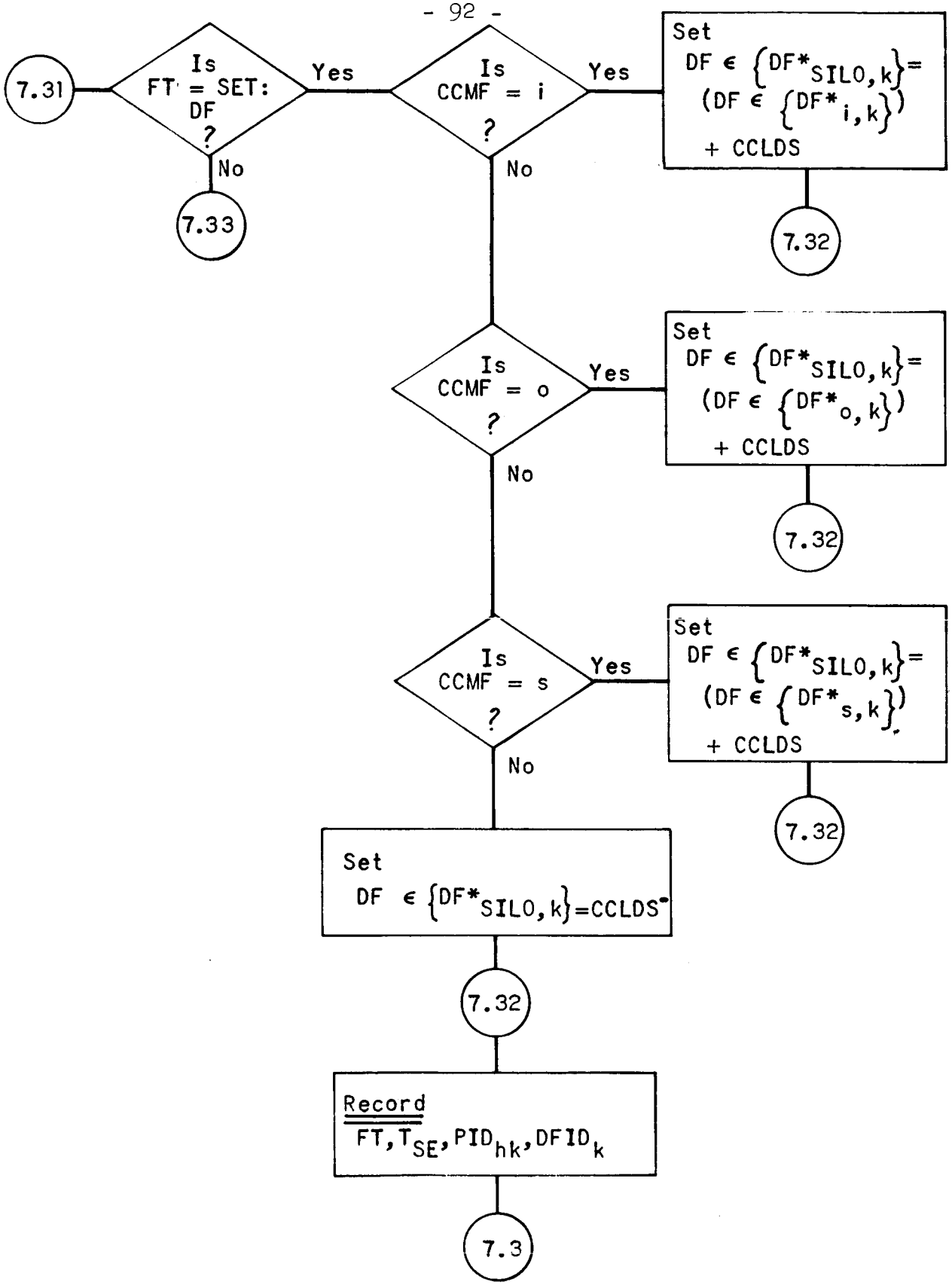
Processing Housekeeping:
Set a serial number for a data frame.
UNIVAC 494 Algorithm
FIGURE A7-13 .



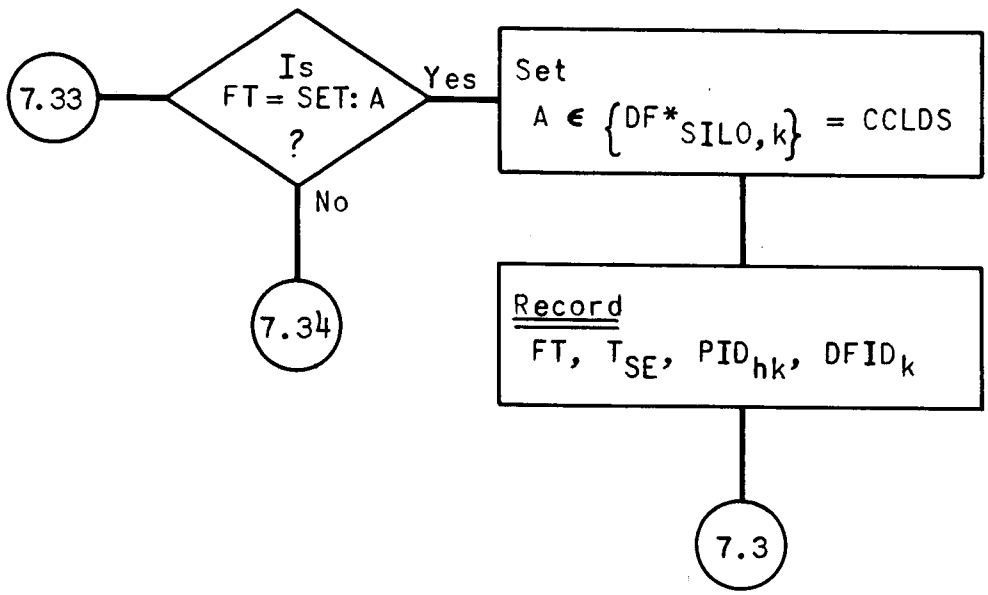
Processing Housekeeping:
Set a message number for a data frame.
UNIVAC 494 Algorithm
FIGURE A7-14



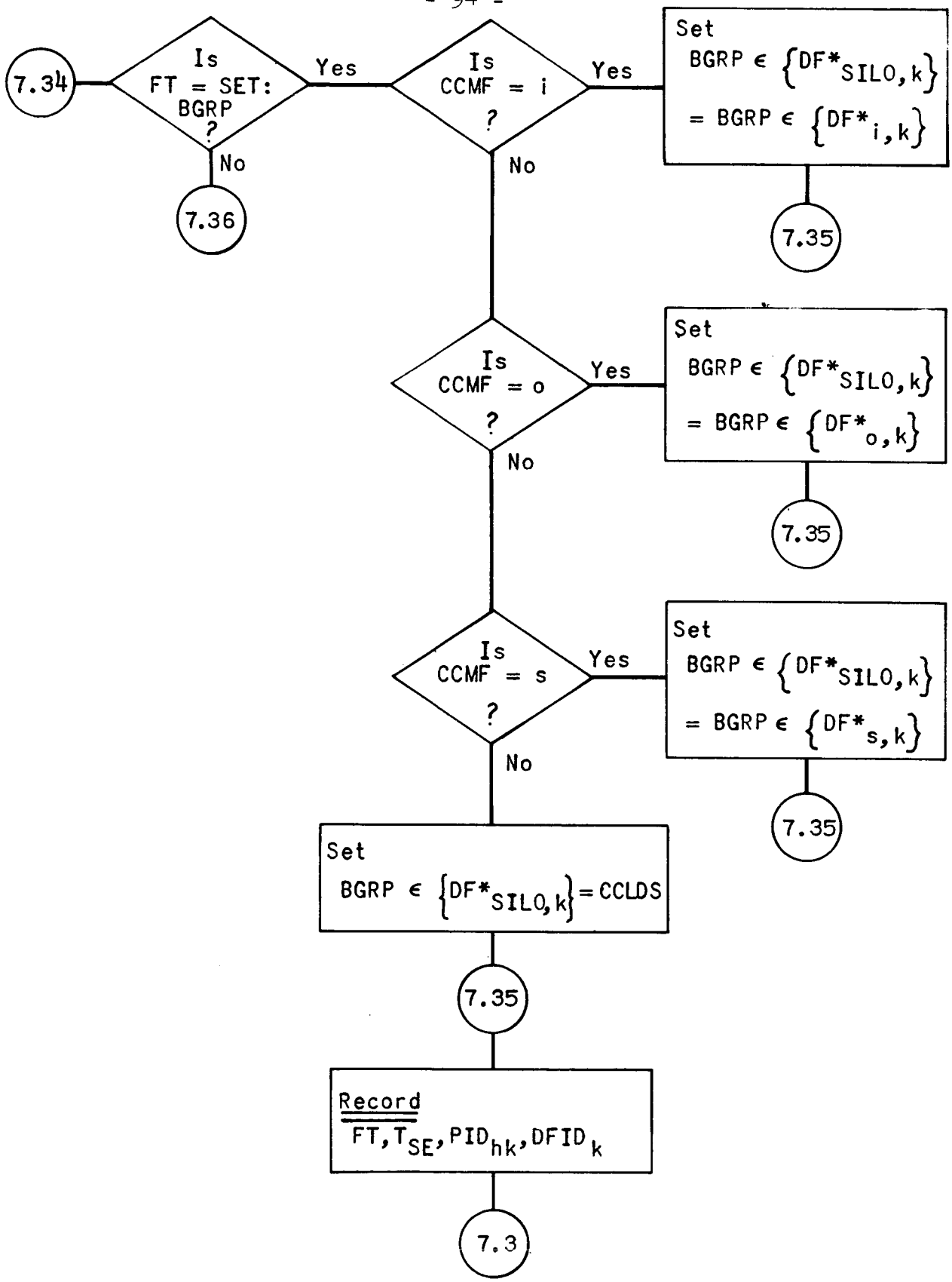
Processing Housekeeping:
Set a segment number for a data frame.
UNIVAC 494 Algorithm
FIGURE A7-15



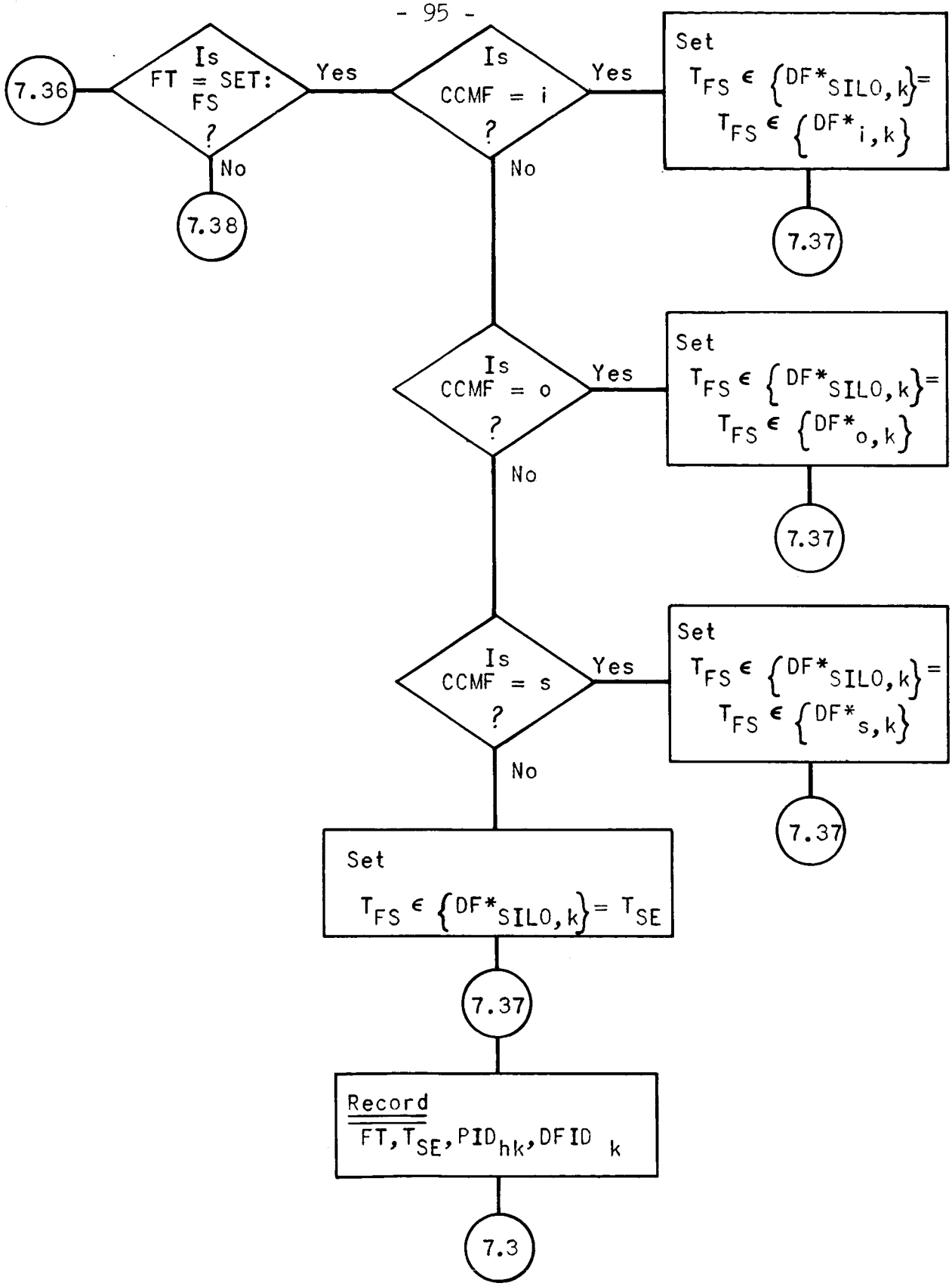
Processing Housekeeping:
Set the number of characters in a data frame.
UNIVAC 494 Algorithm
FIGURE A7-16



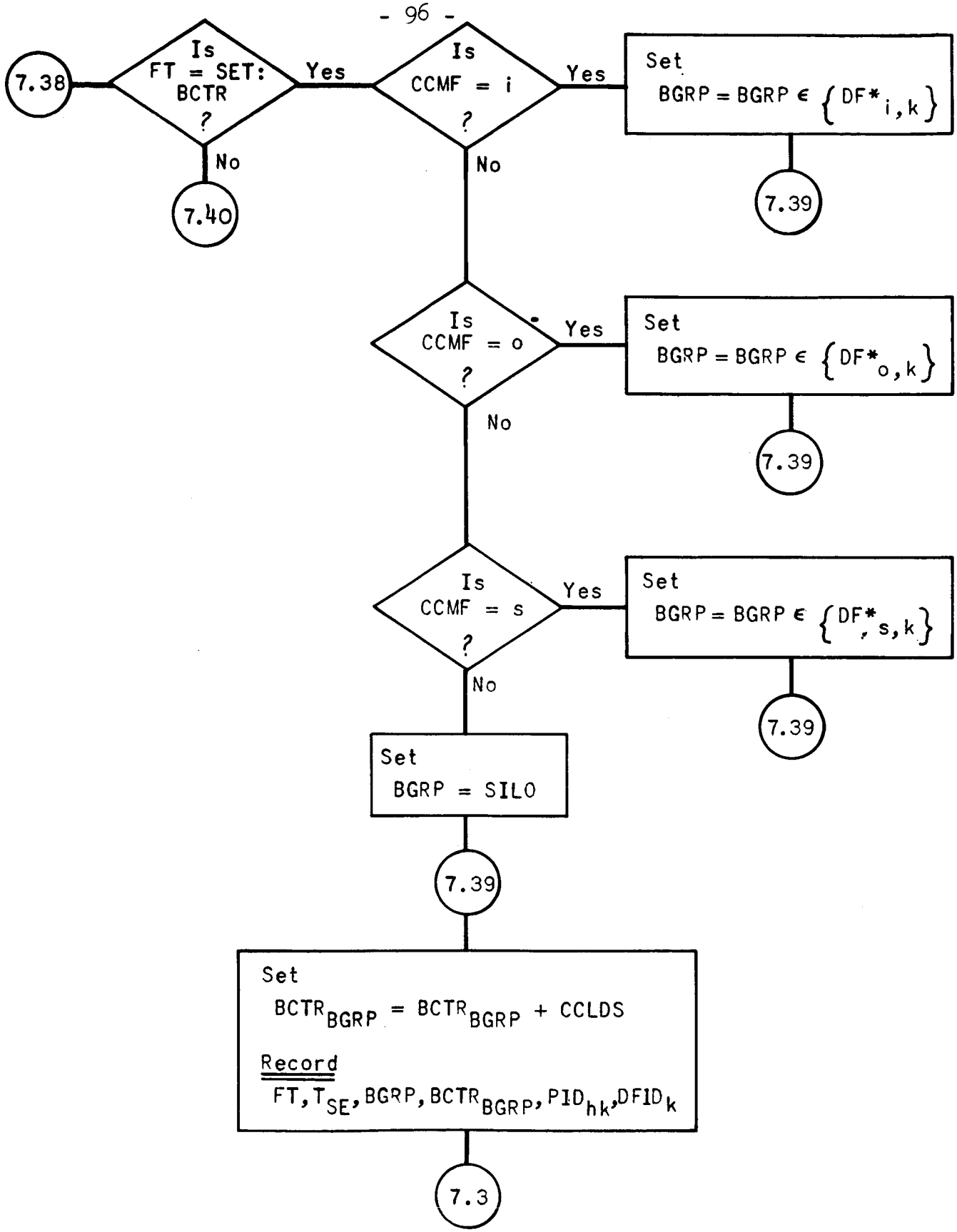
Processing Housekeeping:
Set the processing chain for a data frame.
UNIVAC 494 Algorithm
FIGURE A7-17



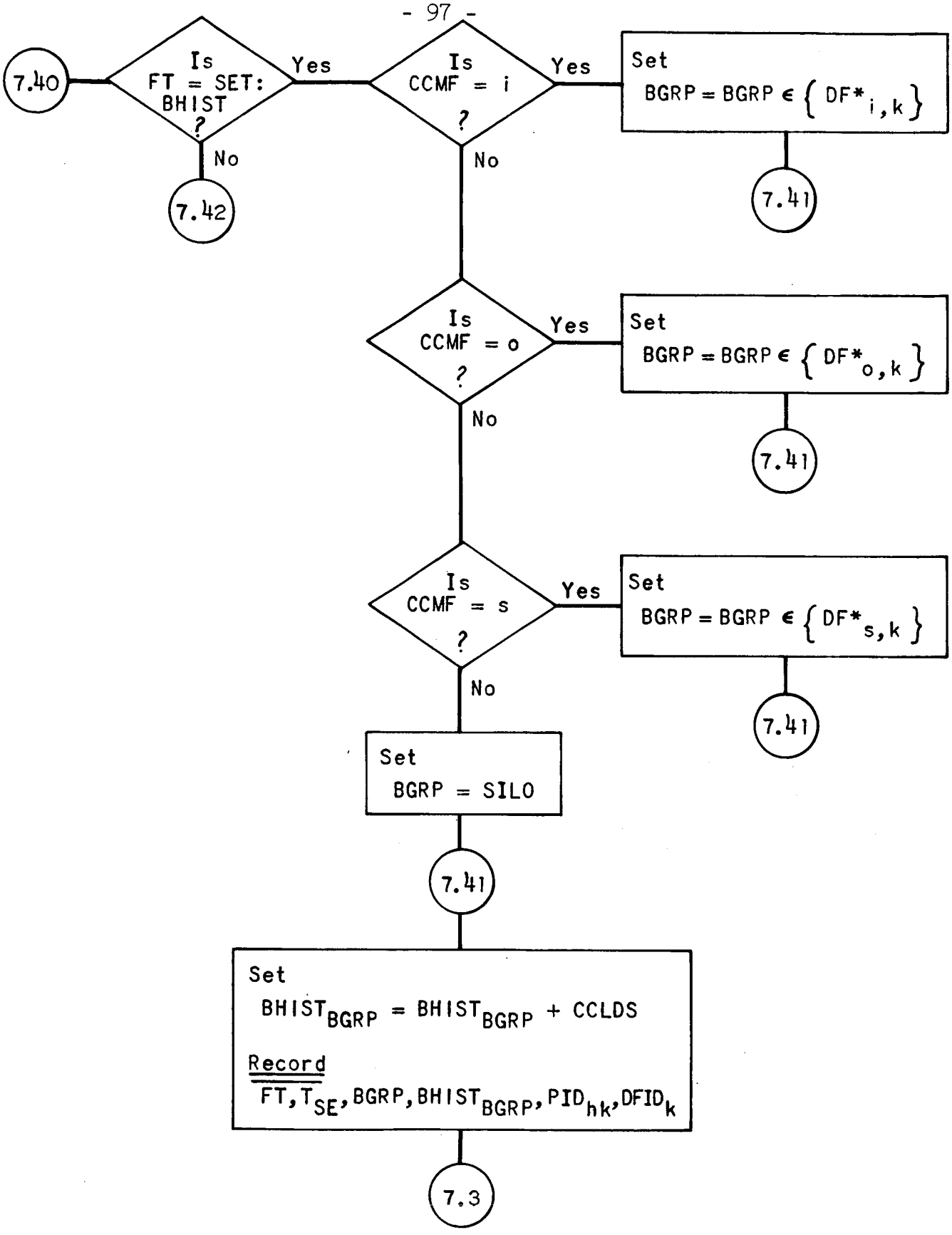
Processing Housekeeping:
Set the buffer group used by a data frame.
UNIVAC 494 Algorithm
FIGURE A7-18



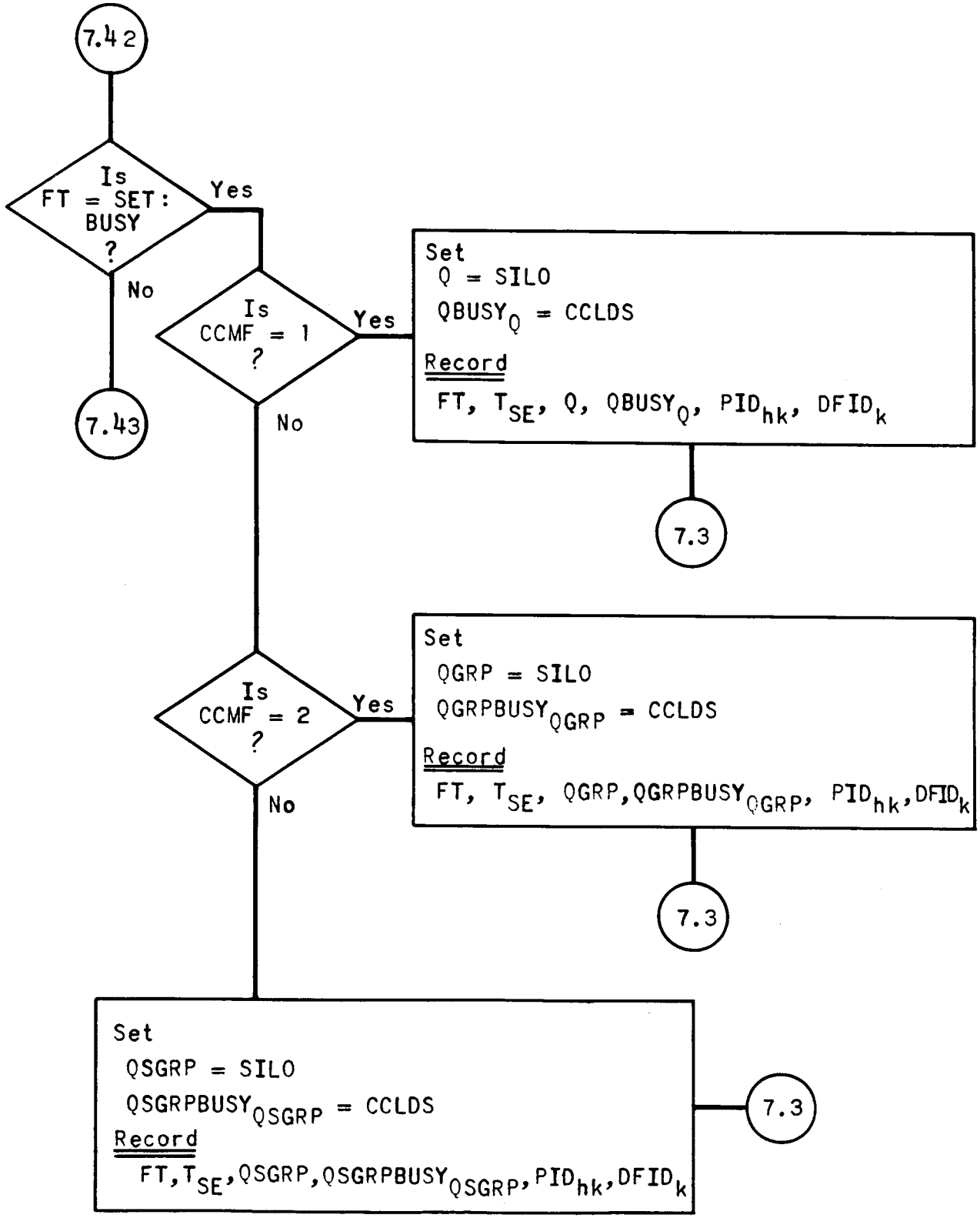
Processing Housekeeping: Set the start time of a data frame.
UNIVAC 494 Algorithm
FIGURE A7-19



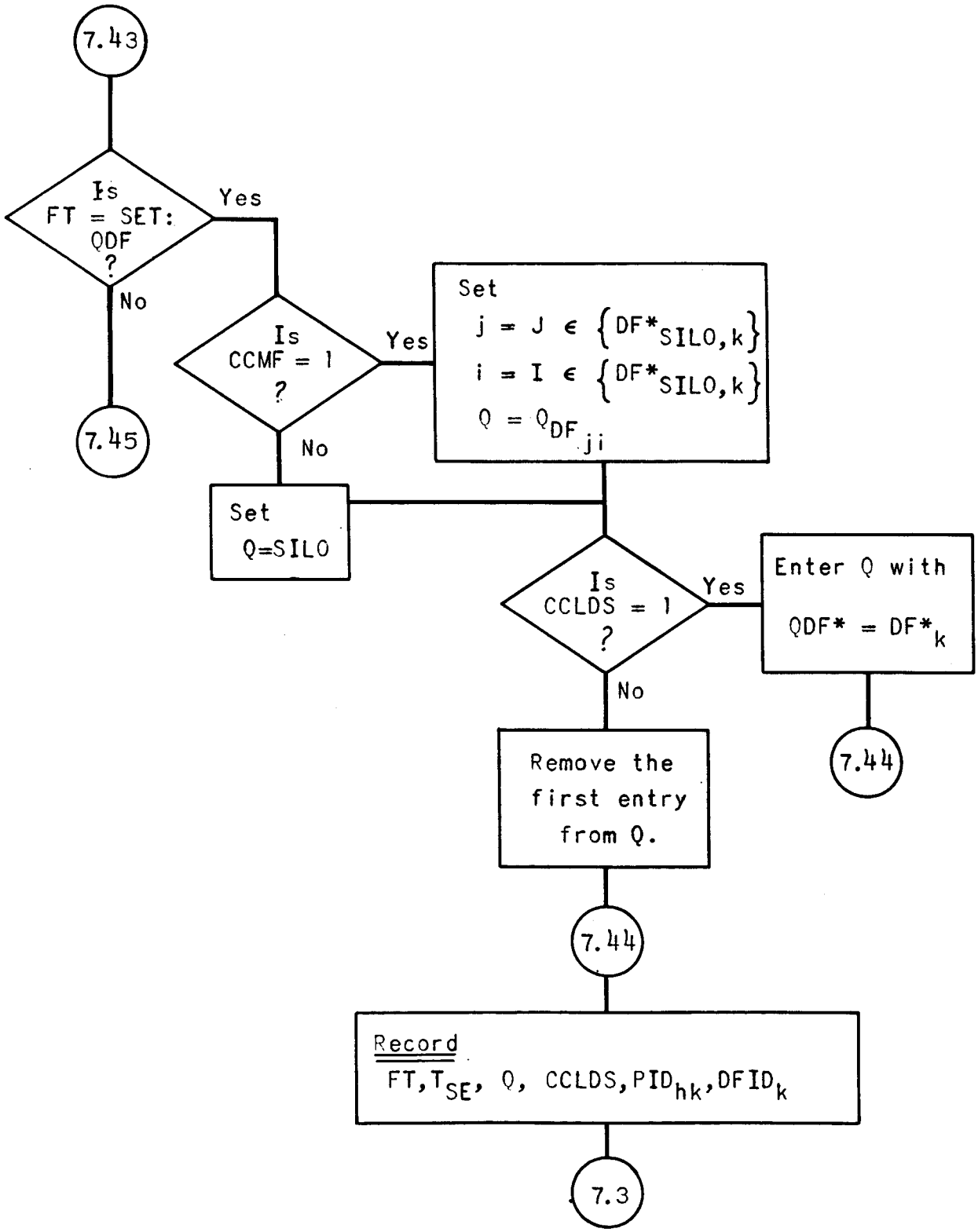
Processing Housekeeping: Change the value of a buffer counter.
UNIVAC 494 Algorithm
FIGURE A7-20



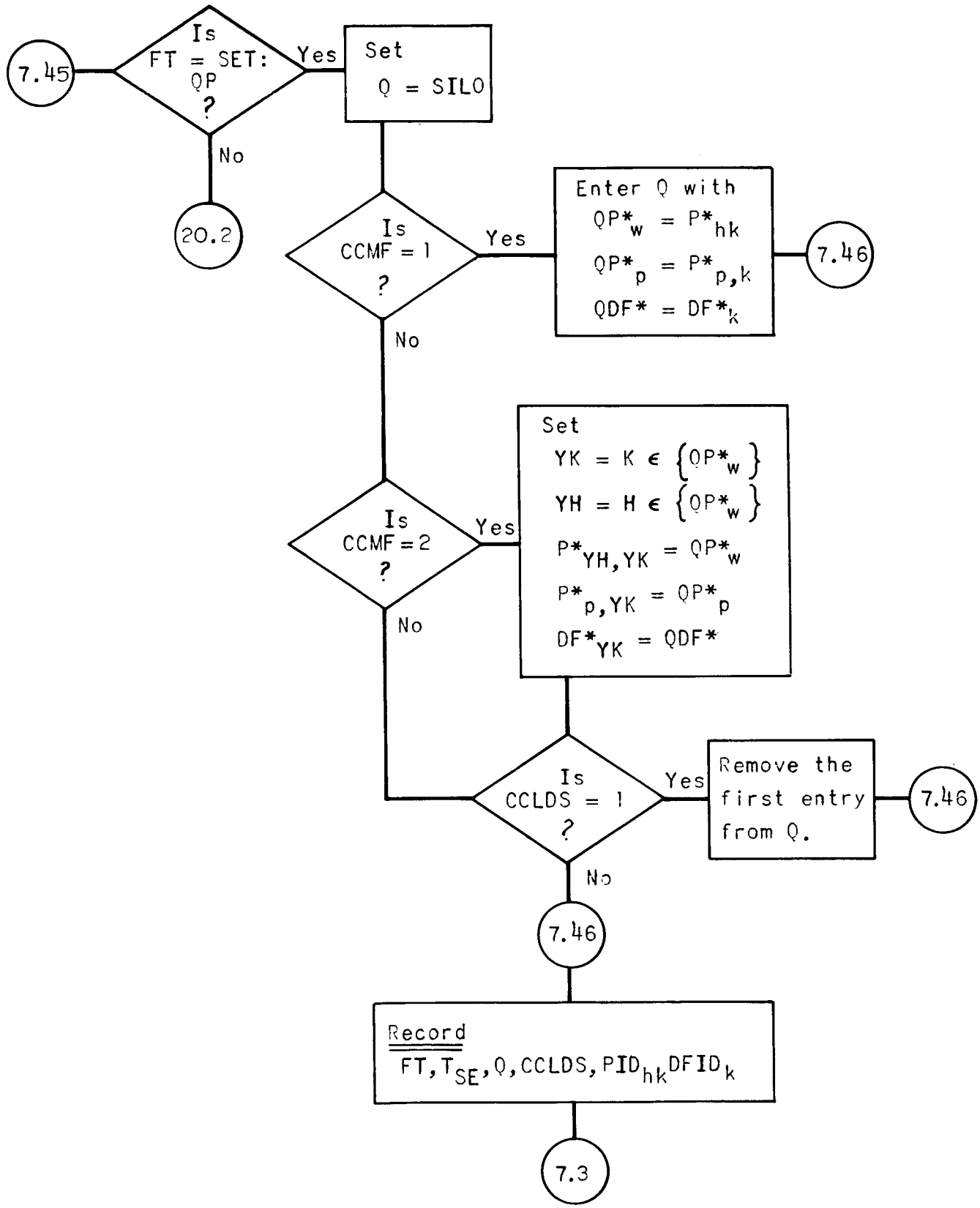
Processing Housekeeping: Change the Value of a buffer histogram.
UNIVAC 494 Algorithm
FIGURE A7-21



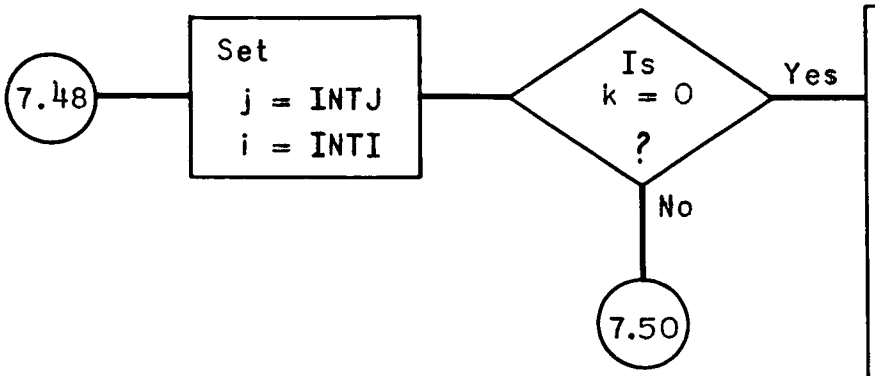
Processing Housekeeping: Change the value of a busy indicator.
UNIVAC 494 Algorithm
FIGURE A7-22



Processing Housekeeping:
Add or delete a member of a data frame queue.
UNIVAC 494 Algorithm
FIGURE A7-23

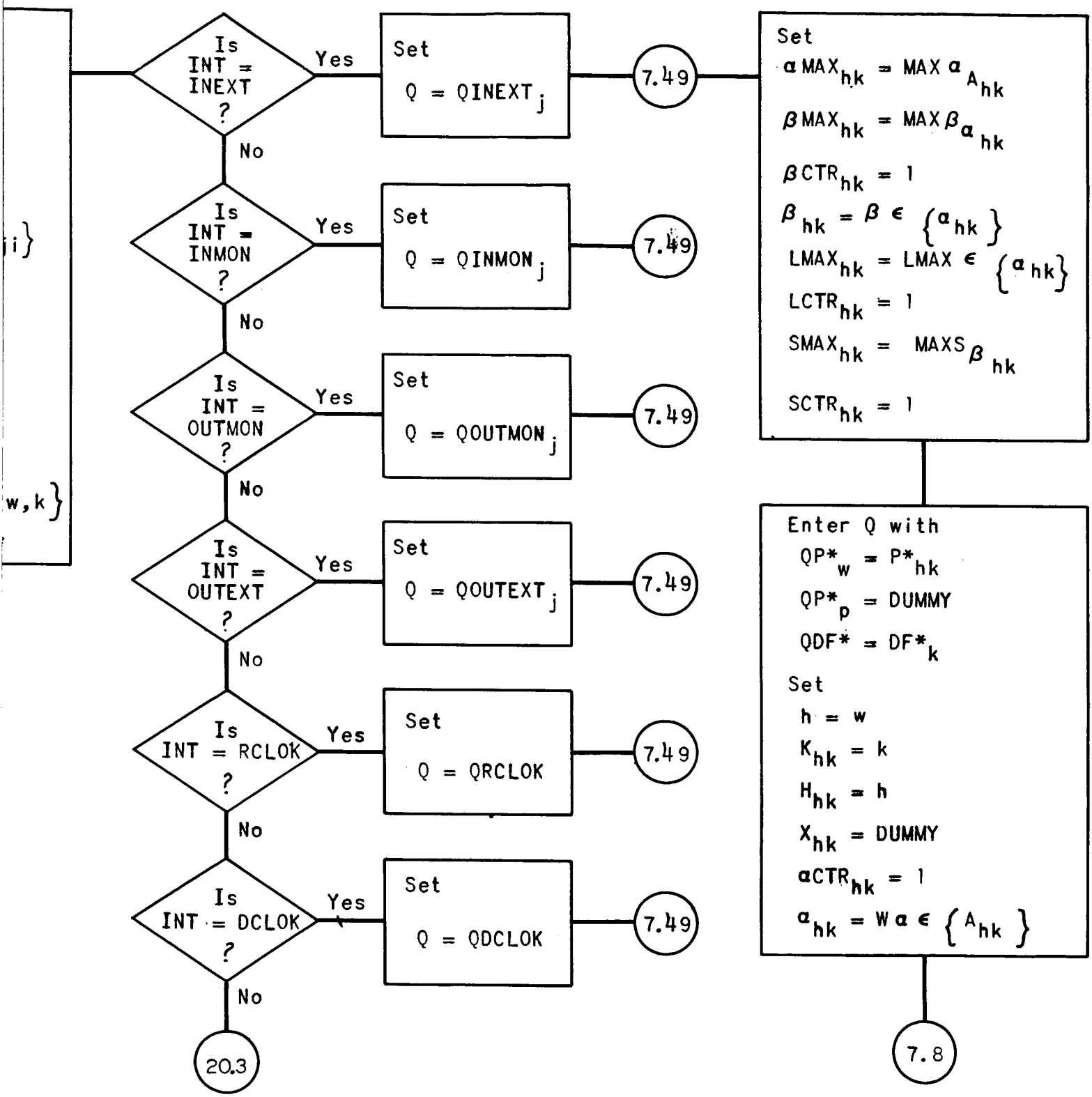


Processing Housekeeping:
Add or delete a member of a processing queue.
UNIVAC 494 Algorithm
FIGURE A7-24

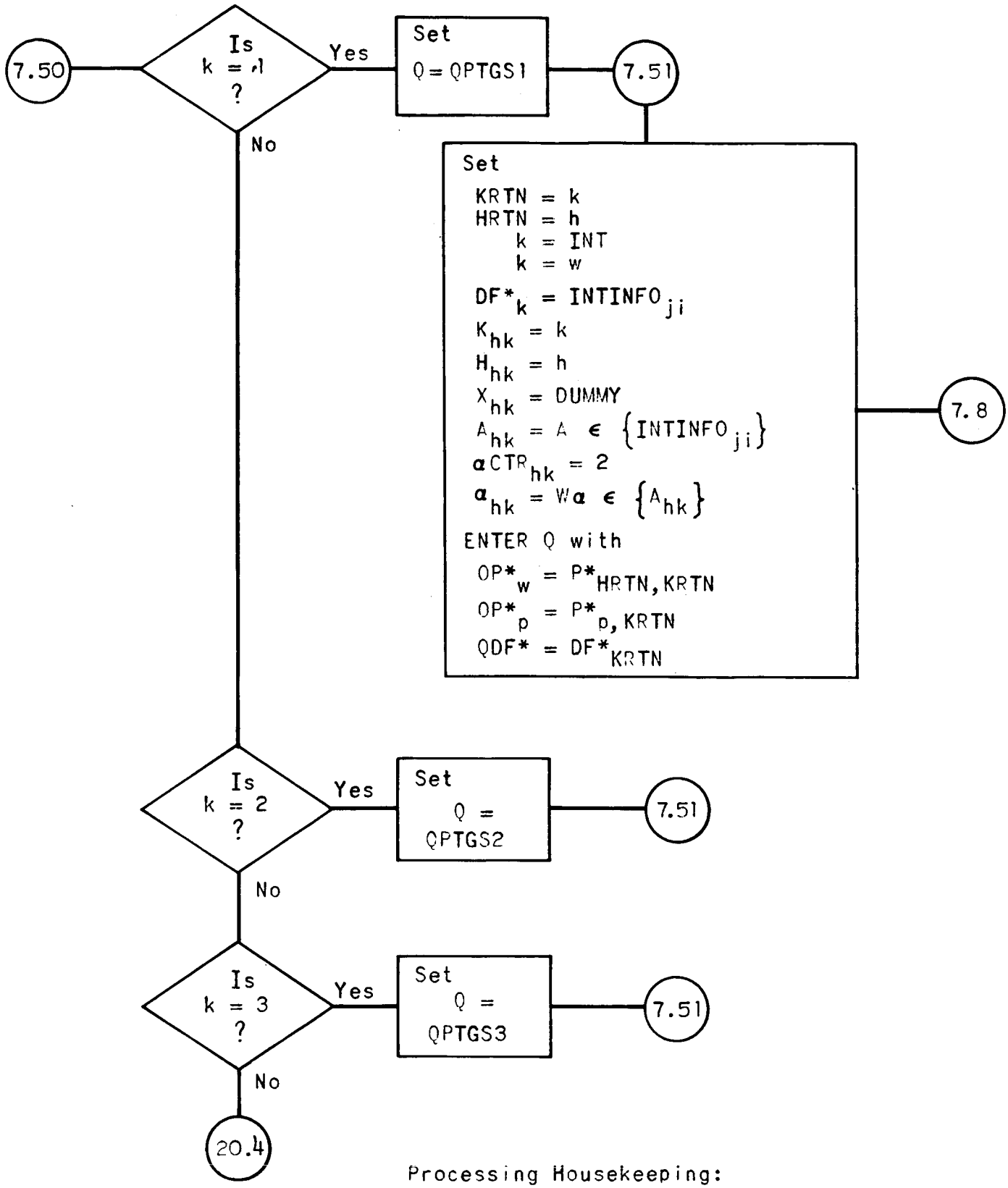


Set
$KRTN = k$
$HRTN = h$
$k = INT$
$h = p$
$DF^*_k = INTINFO_{ji}$
$A_{w,k} = A \in \{INTINFO$
$\alpha CTR_{w,k} = 3$
$K_{hk} = WK \in \{A_{w,k}\}$
$H_{hk} = WH \in \{A_{w,k}\}$
$X_{hk} = WX \in \{A_{w,k}\}$
$A_{hk} = WA \in \{A_{w,k}\}$
$\alpha CTR_{hk} = W\alpha CTR \in \{A$
$\alpha_{hk} = W\alpha \in \{A_{w,k}$

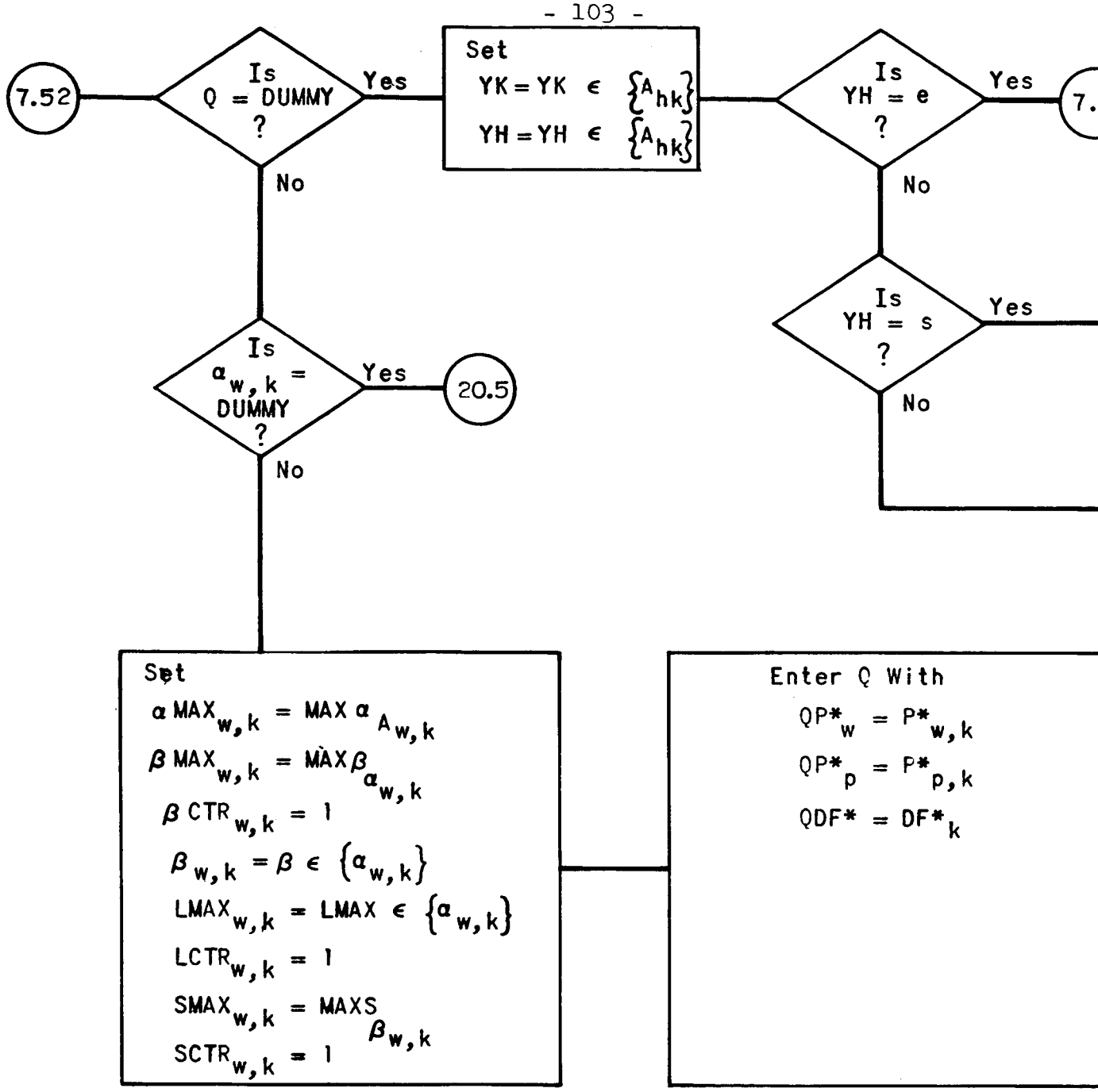
Processing Housekeeping: Set up the pro
UNIVAC 1
FI



essing for an I/O interrupt during a nonsuspendable α.
94 Algorithm
URE A7-25



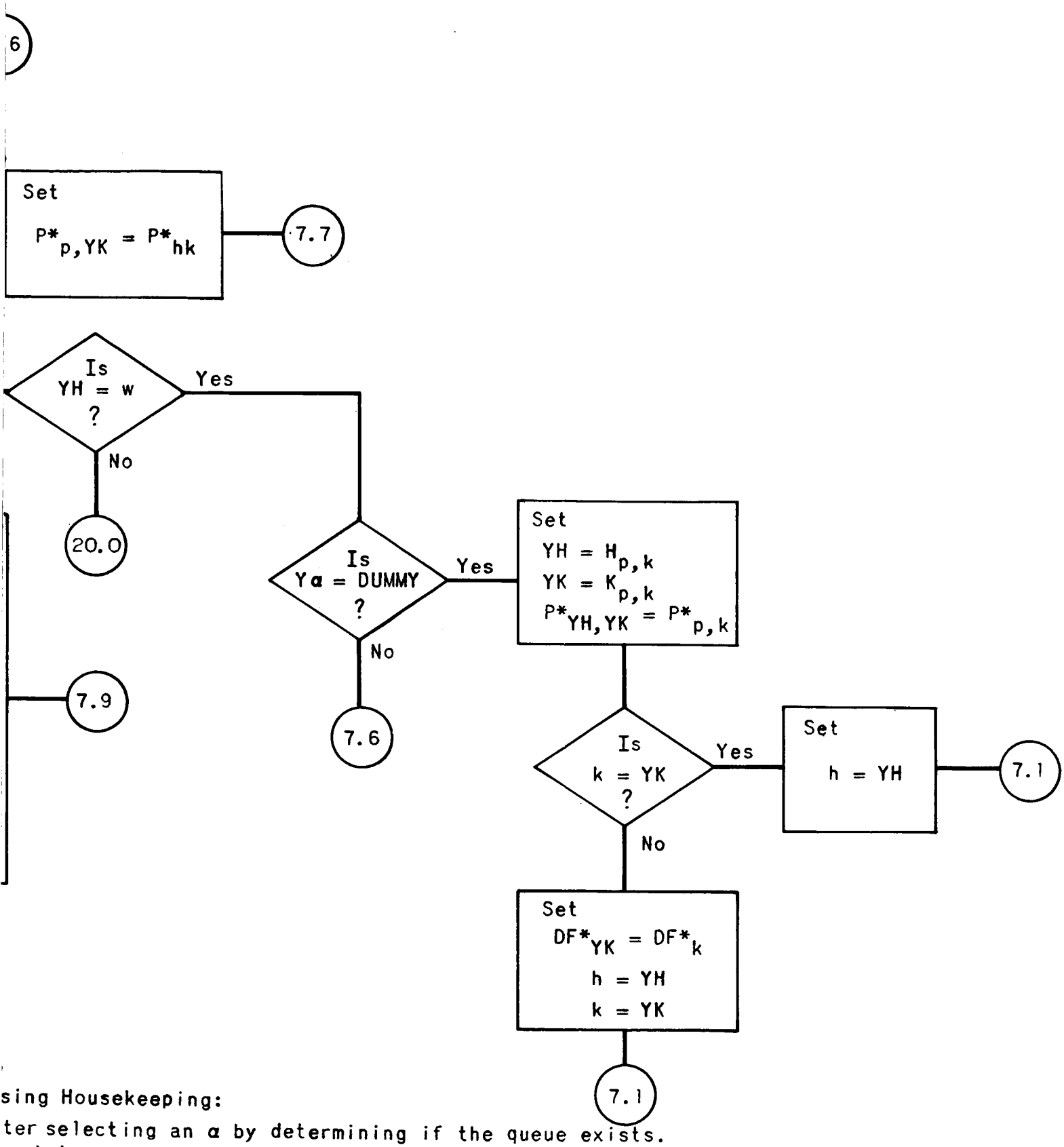
Processing Housekeeping:
Set up the processing for an I/O interrupt during a suspendable α.
UNIVAC 494 Algorithm
FIGURE A7-26



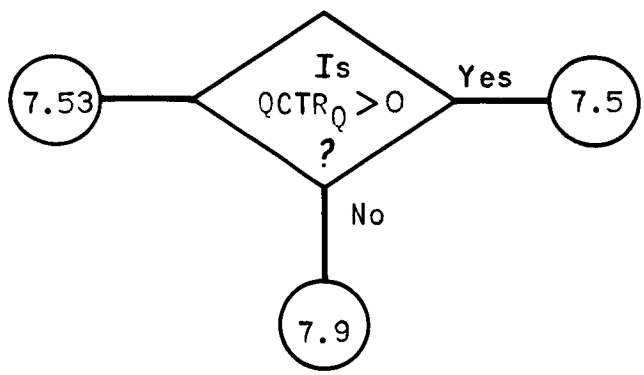
Proces

Prepare to set up housekeeping variables af

UNIV



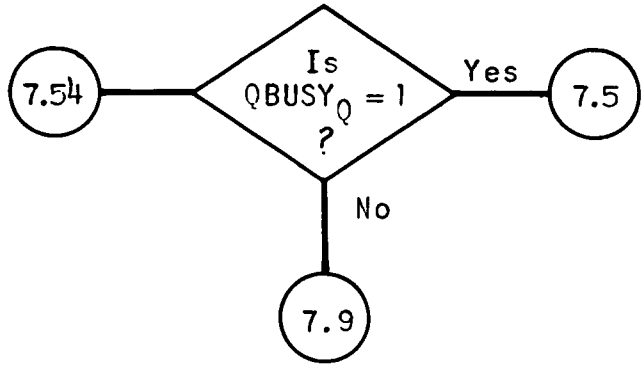
sing Housekeeping:
ter selecting an α by determining if the queue exists.
AC 494 Algorithm
FIGURE A7-27



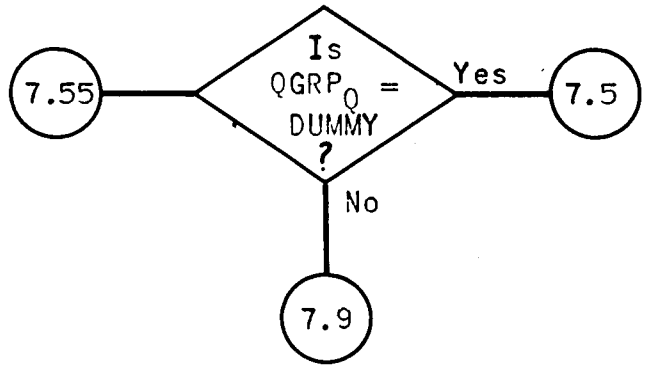
Processing Housekeeping:
Select an α by determining if there are any entries in the queue.

UNIVAC 494 Algorithm

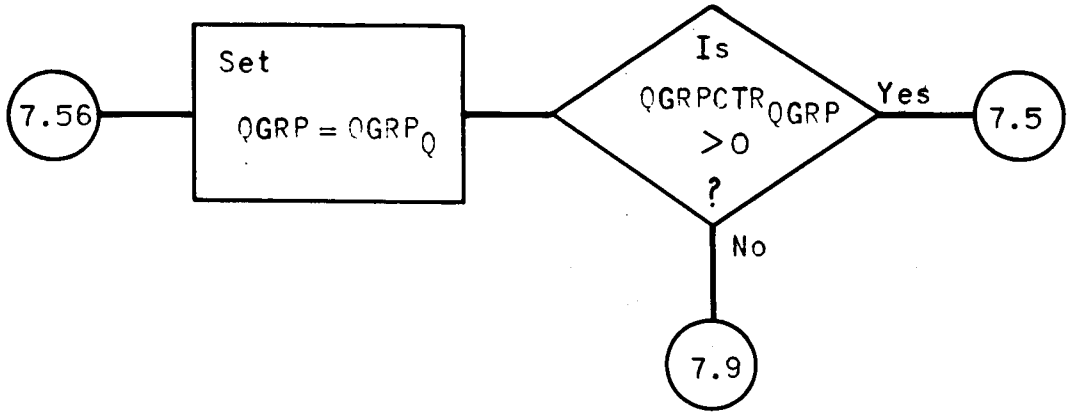
FIGURE A7-28



Processing Housekeeping:
Select an α by determining if the queue is busy.
UNIVAC 494 Algorithm
FIGURE A7-29



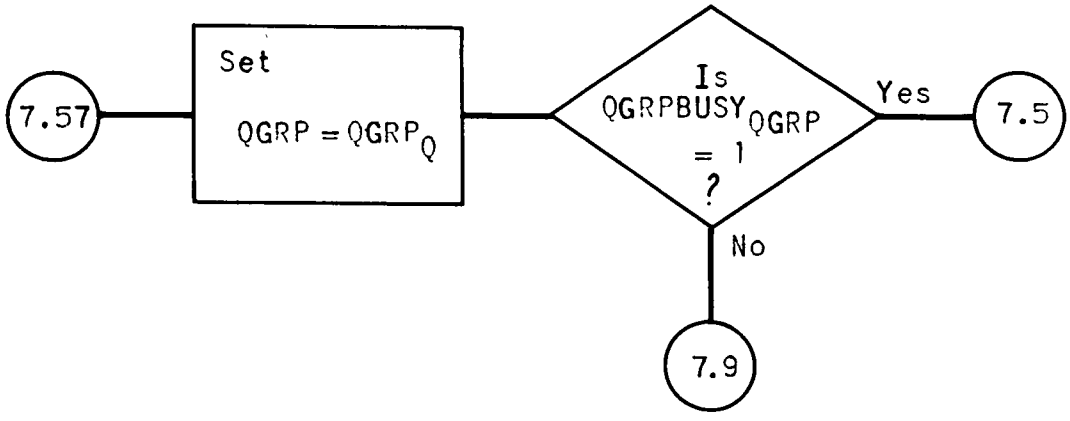
Process Housekeeping:
Select an α by determining if the queue group exists.
UNIVAC 494 Algorithm
FIGURE A7-30



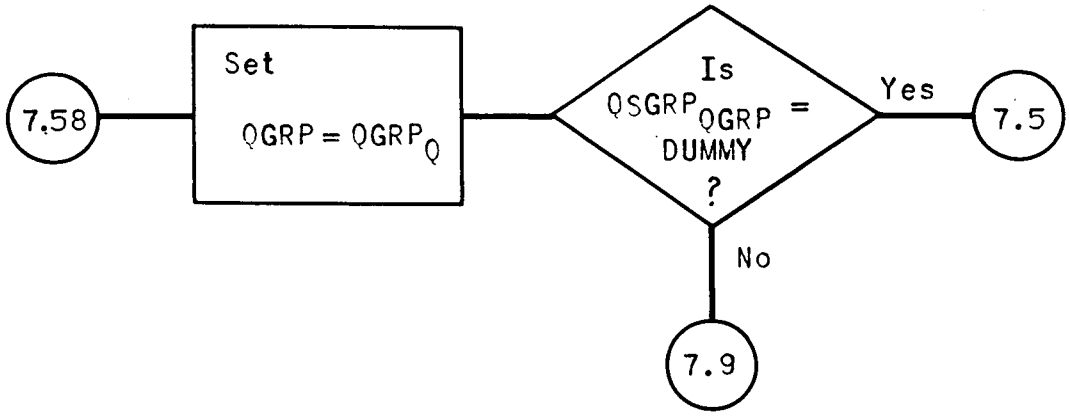
Processing Housekeeping:
Select an α by determining if there are any entries in the queue group.

UNIVAC 494 Algorithm

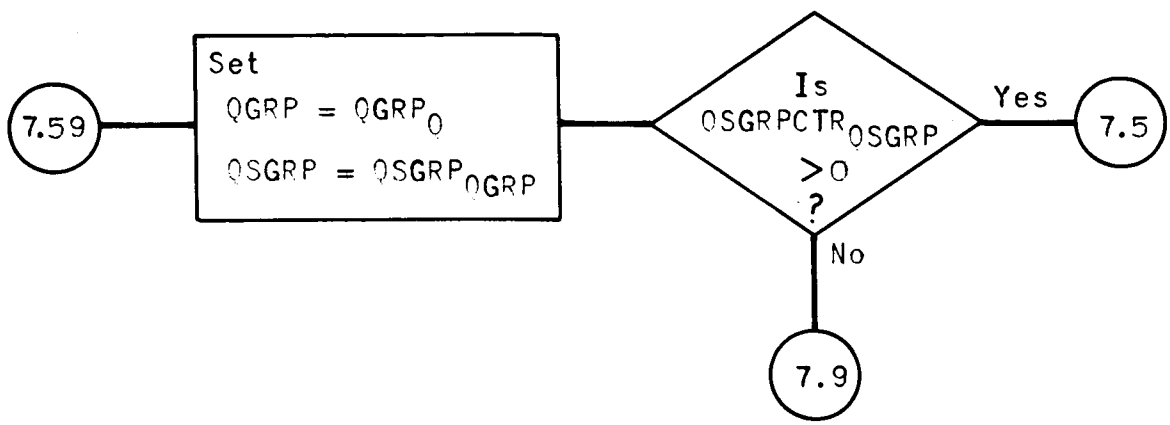
FIGURE A7-31



Processing Housekeeping:
Select an α by determining if the queue group is busy.
UNIVAC 494 Algorithm
FIGURE A7-32



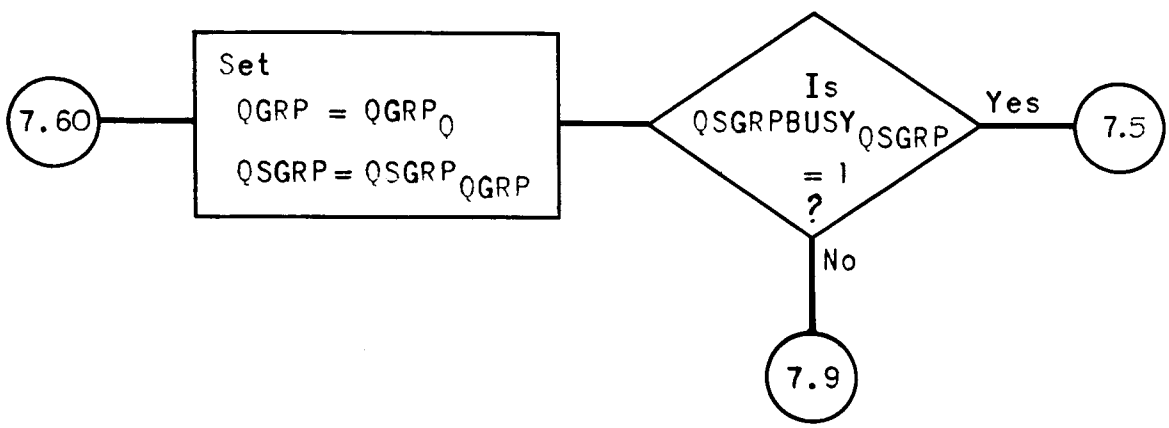
Processing Housekeeping:
Select an α by determining if the queue super group exists.
UNIVAC 494 Algorithm
FIGURE A7-33



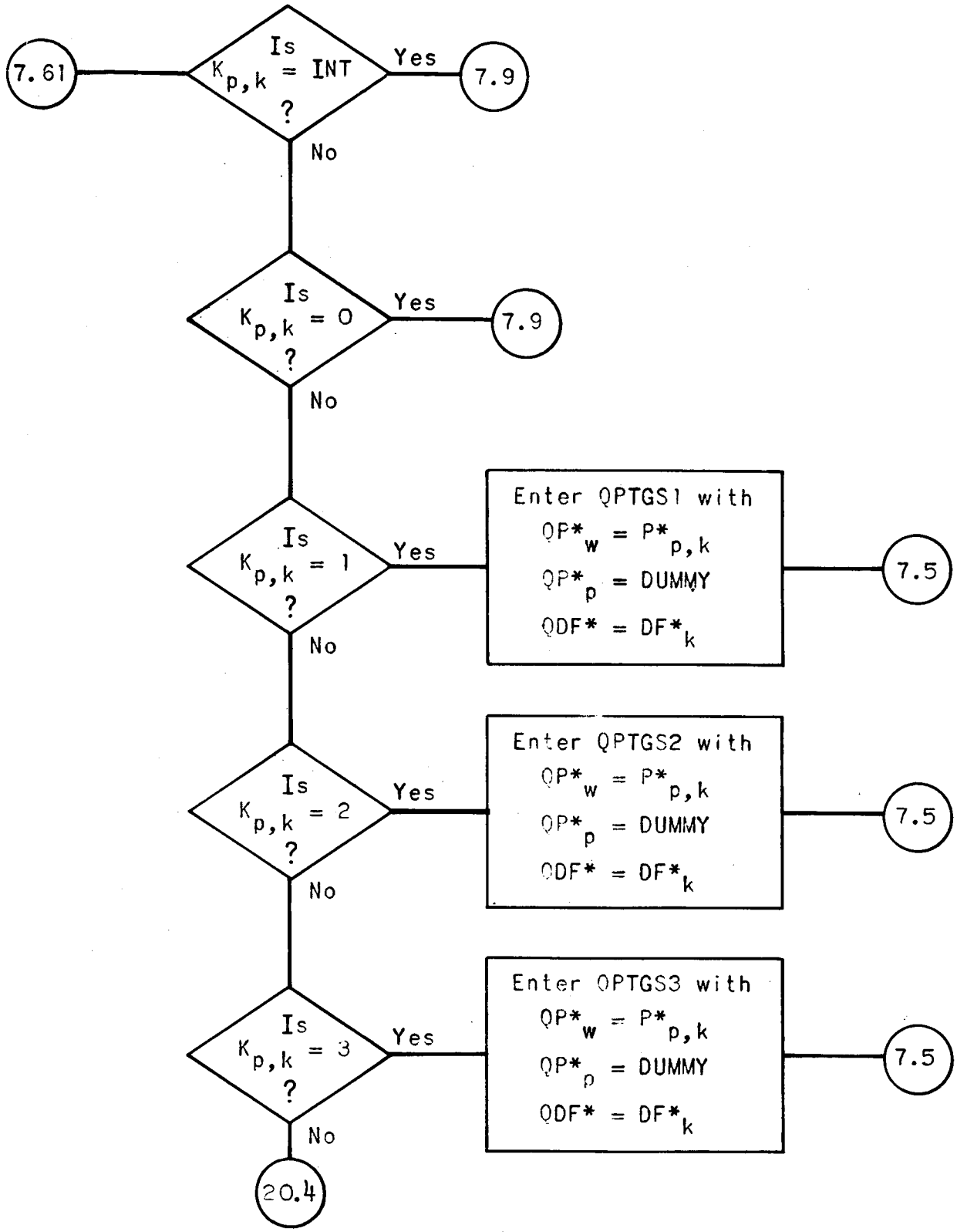
Processing Housekeeping:
Select an α by determining if there are any entries in the queue super group.

UNIVAC 494 Algorithm

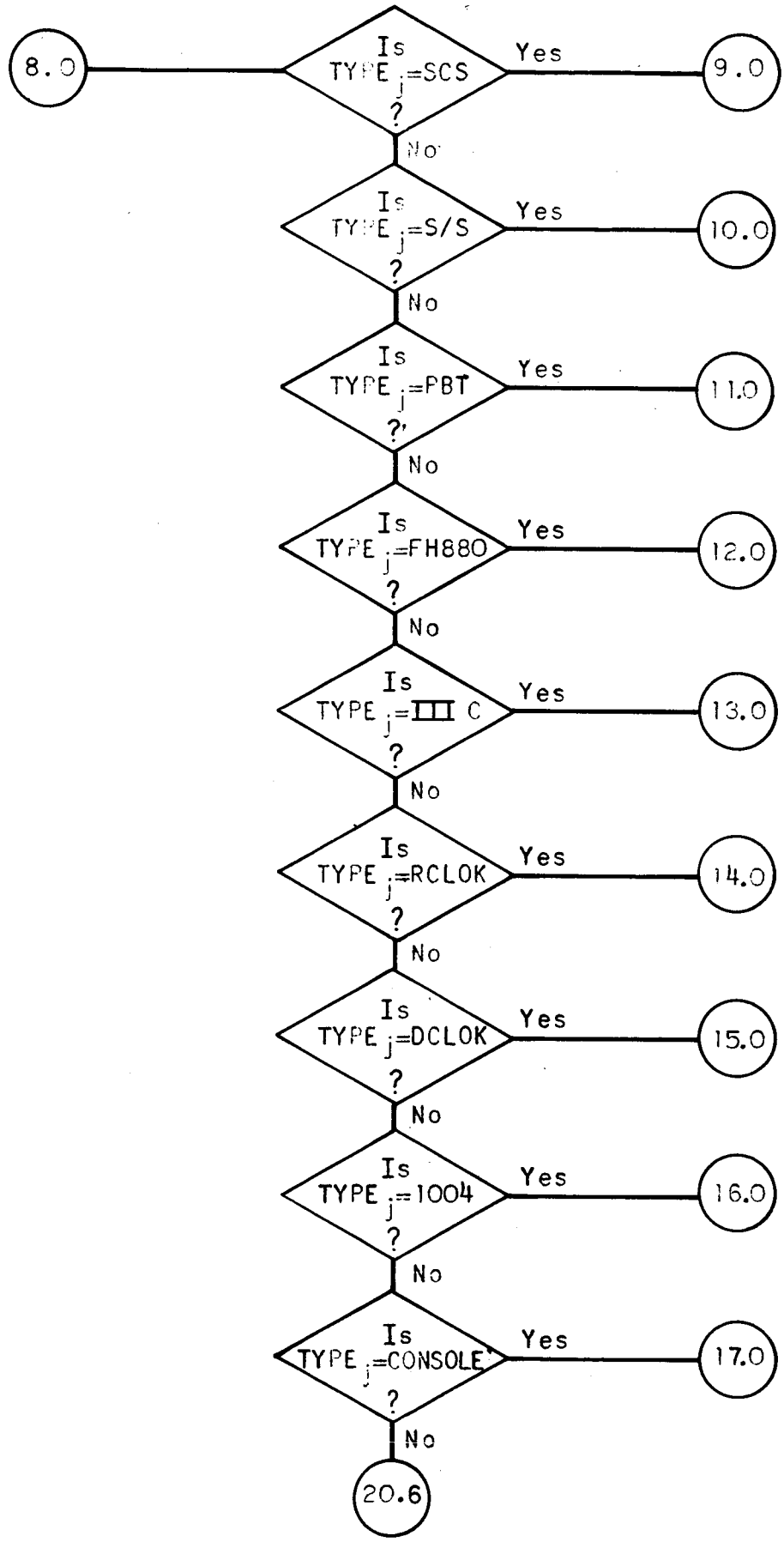
FIGURE A7-34



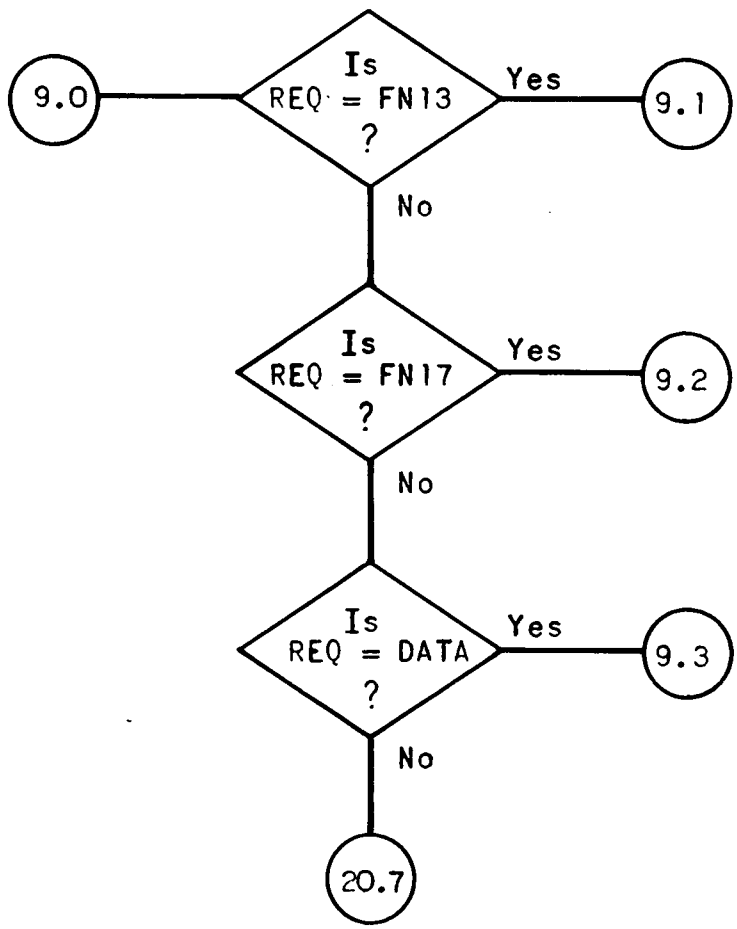
Processing Housekeeping:
Select an α by determining if the queue super group is busy.
UNIVAC 494 Algorithm
FIGURE A7-35



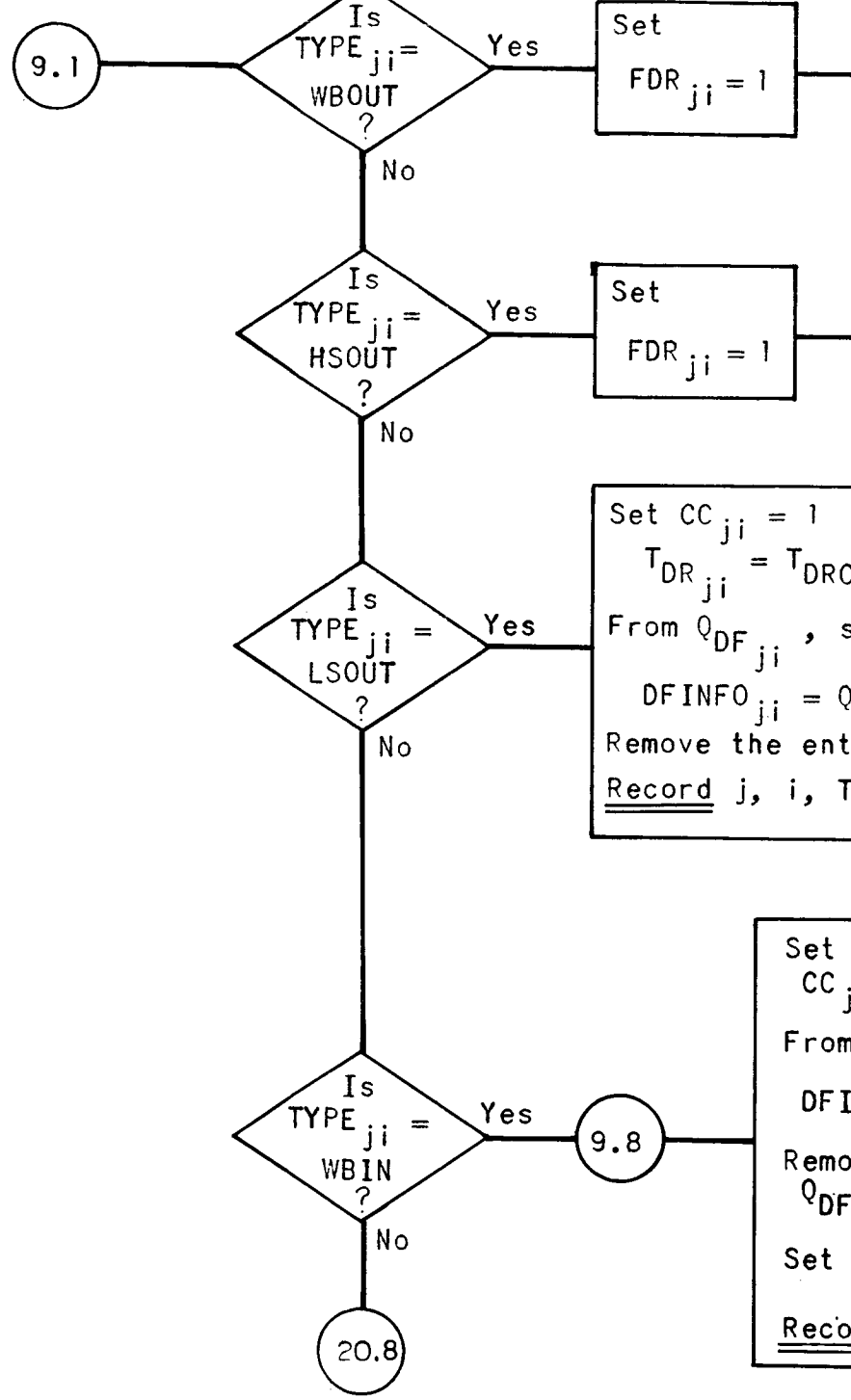
Processing Housekeeping:
Select an α by determining if the predecessor α is suspendable.
UNIVAC 494 Algorithm
FIGURE A7-36



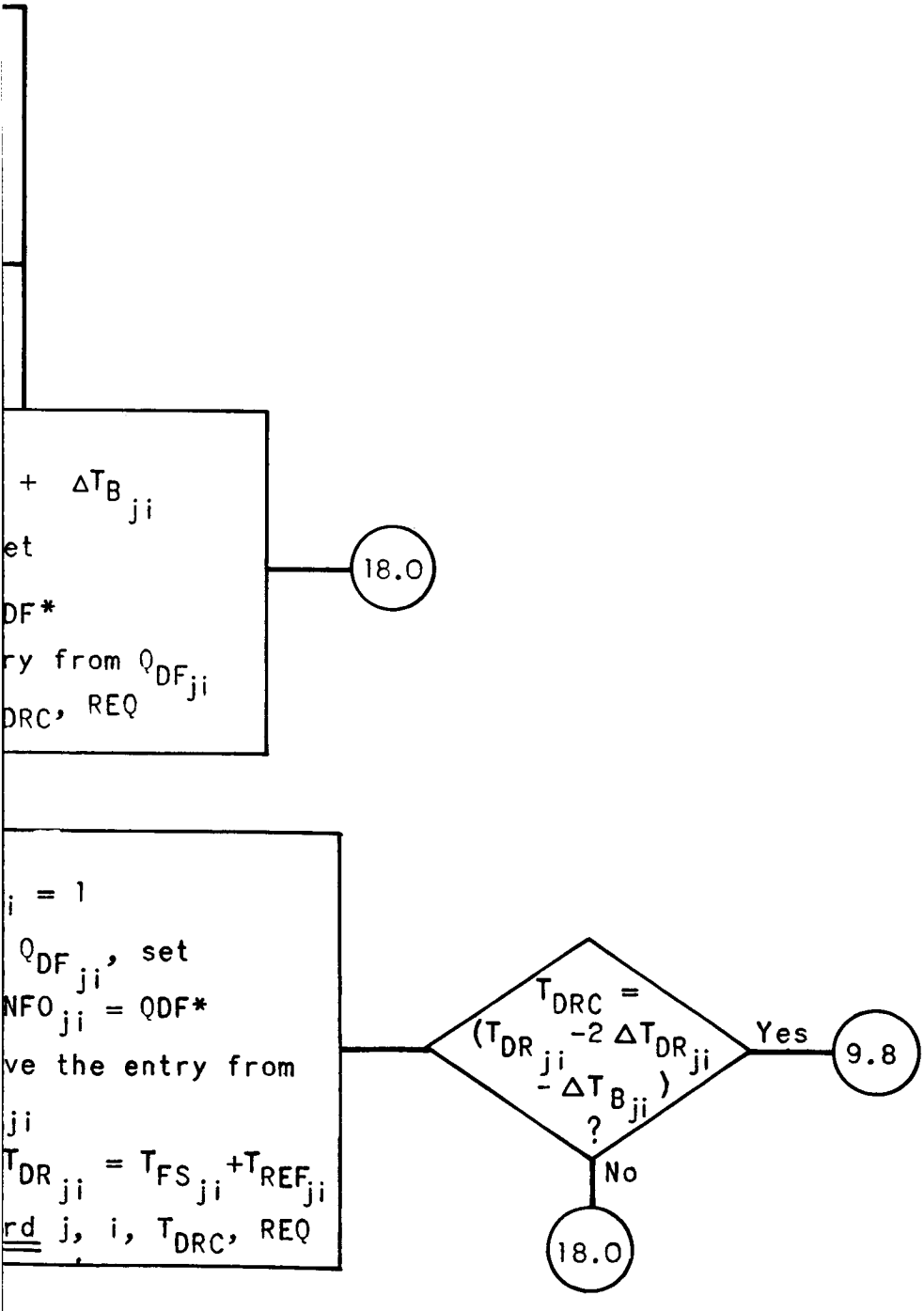
I/O Model Selection: Select the correct I/O model.
UNIVAC 494 Algorithm
FIGURE A8-1



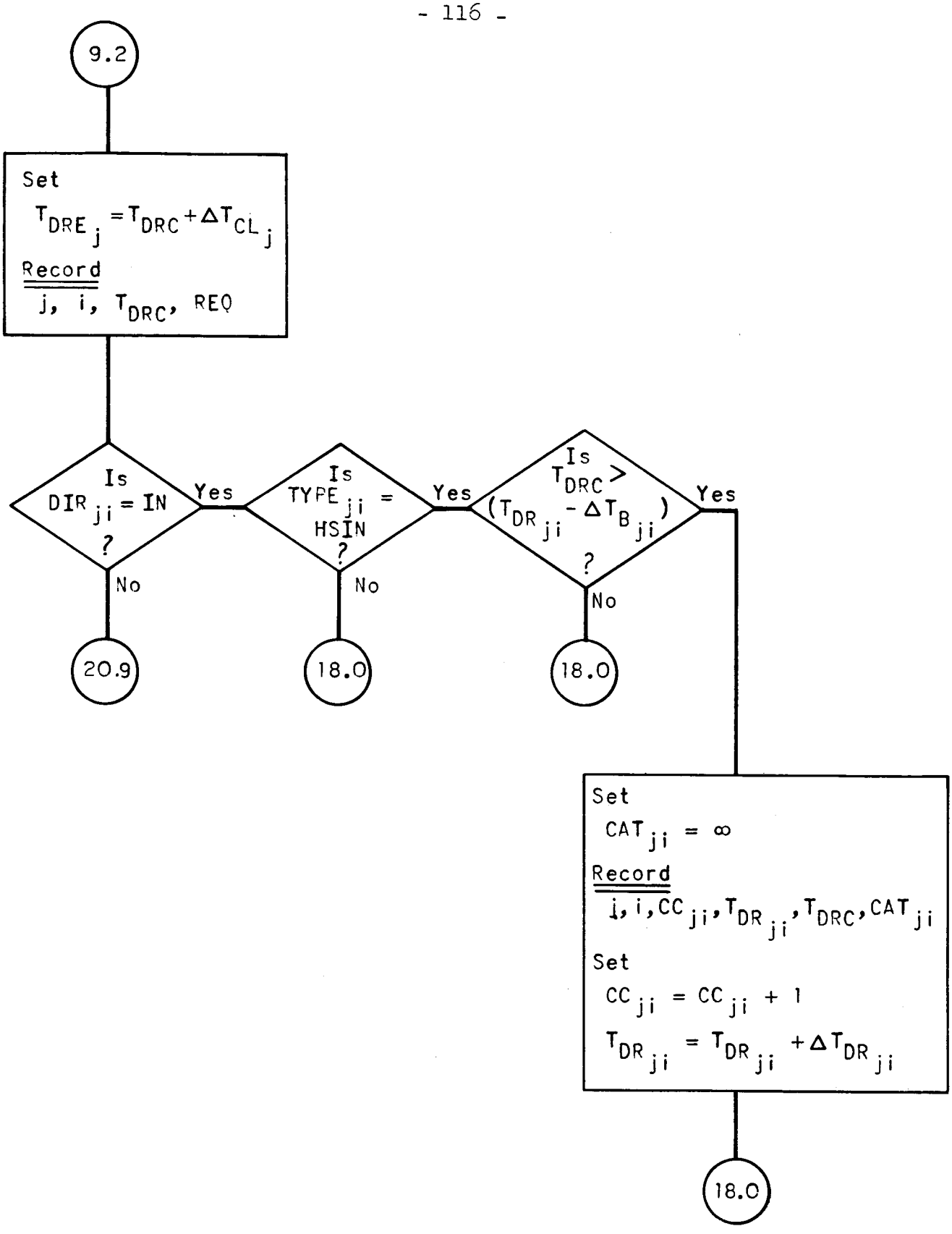
SCS Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A9-1



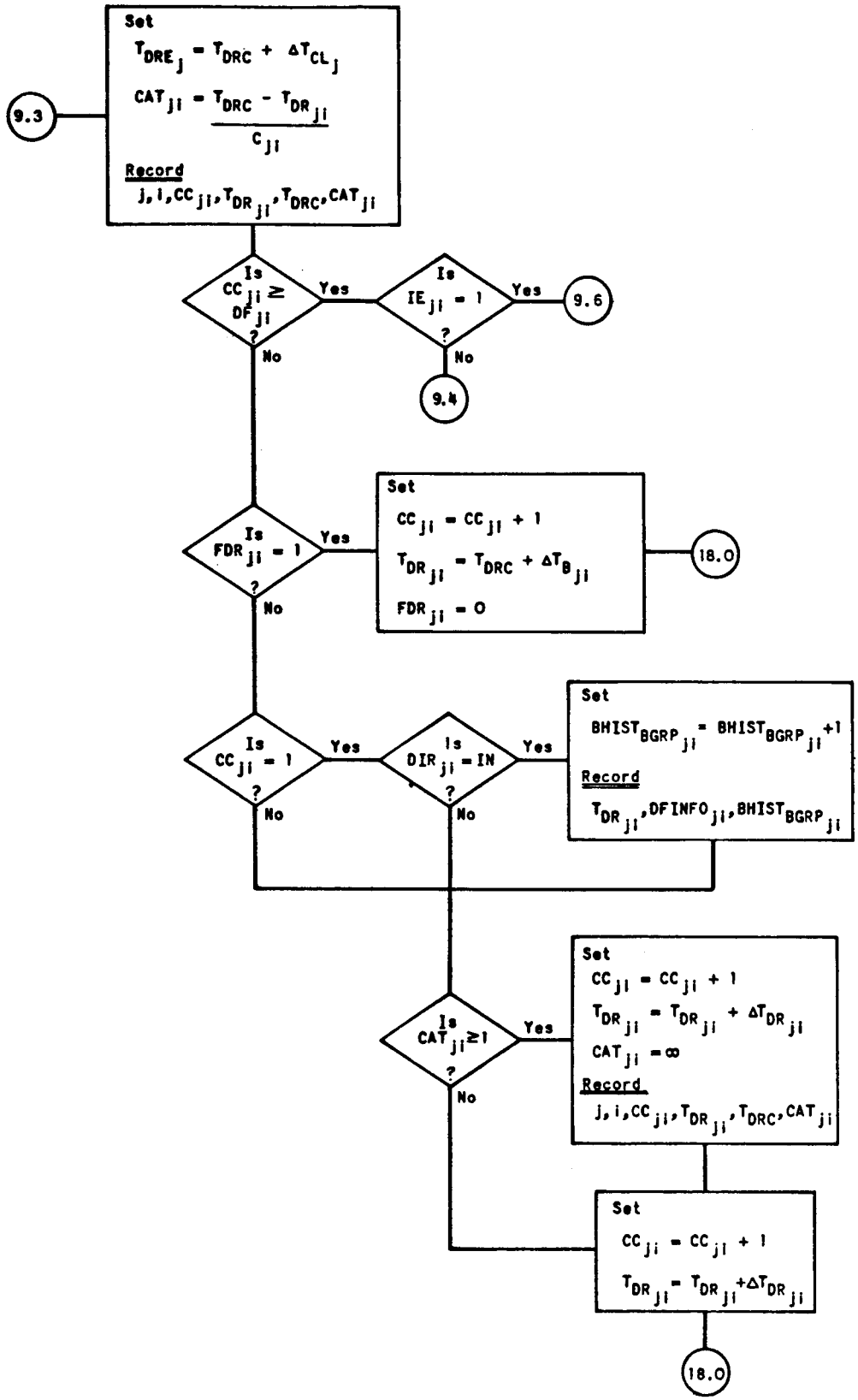
SCS Channel: Perform
UNIVAC 494 A
FIGURE A



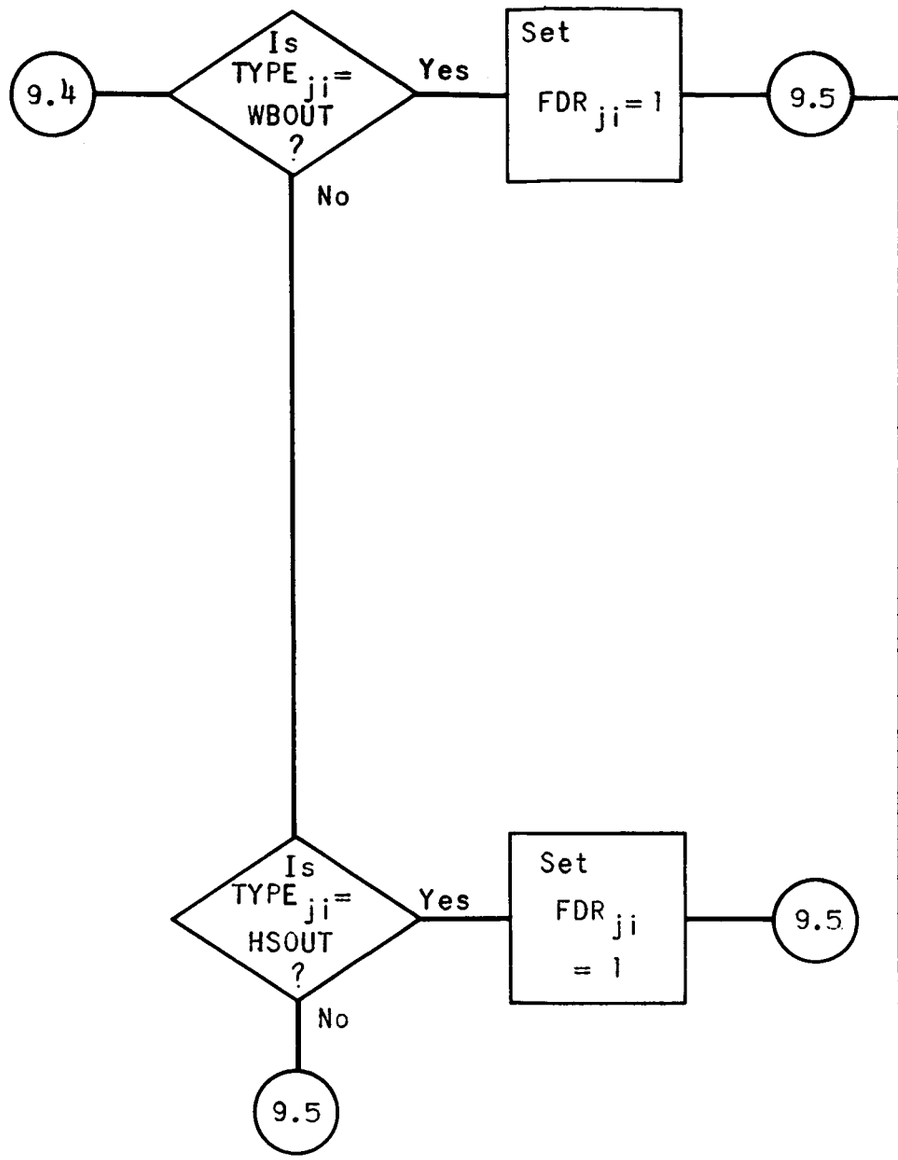
a FN13 action.
lgorithm
9-2



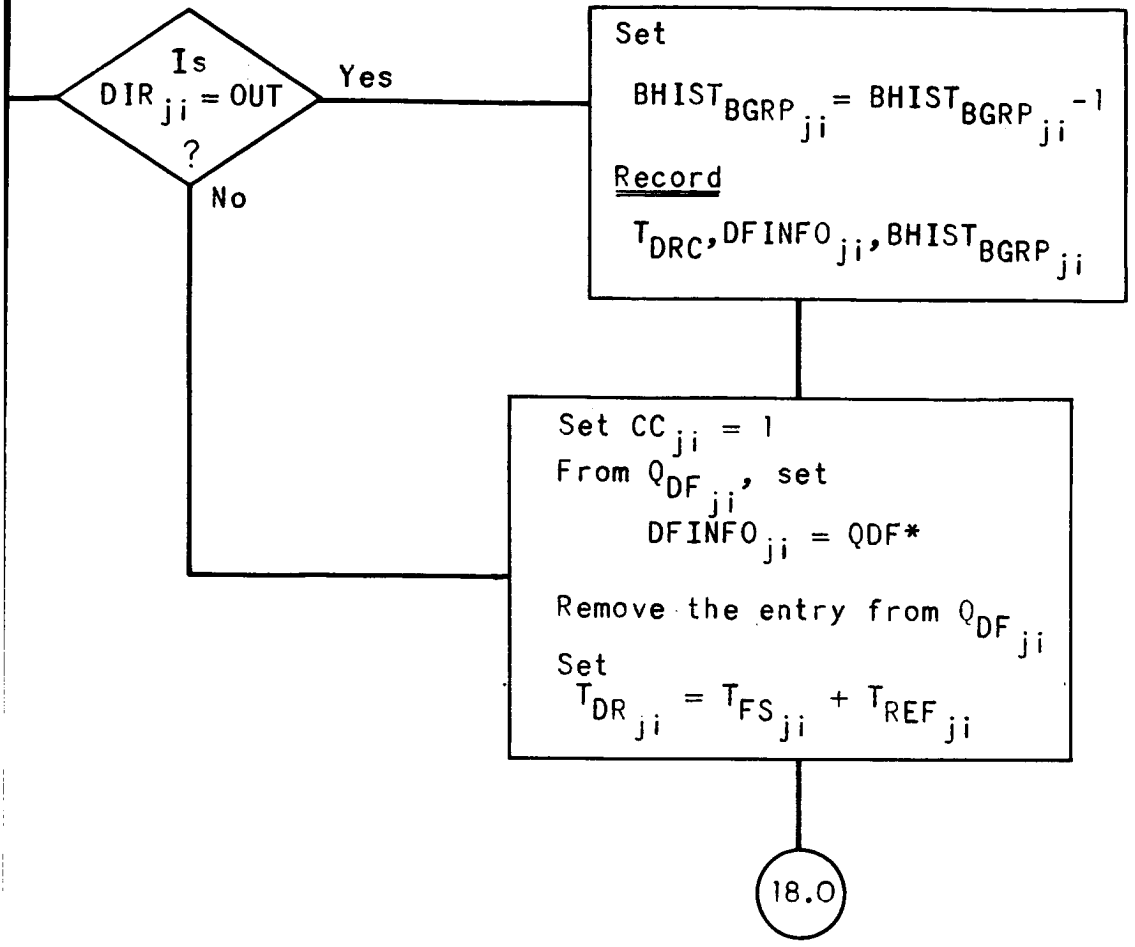
SCS Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A9-3



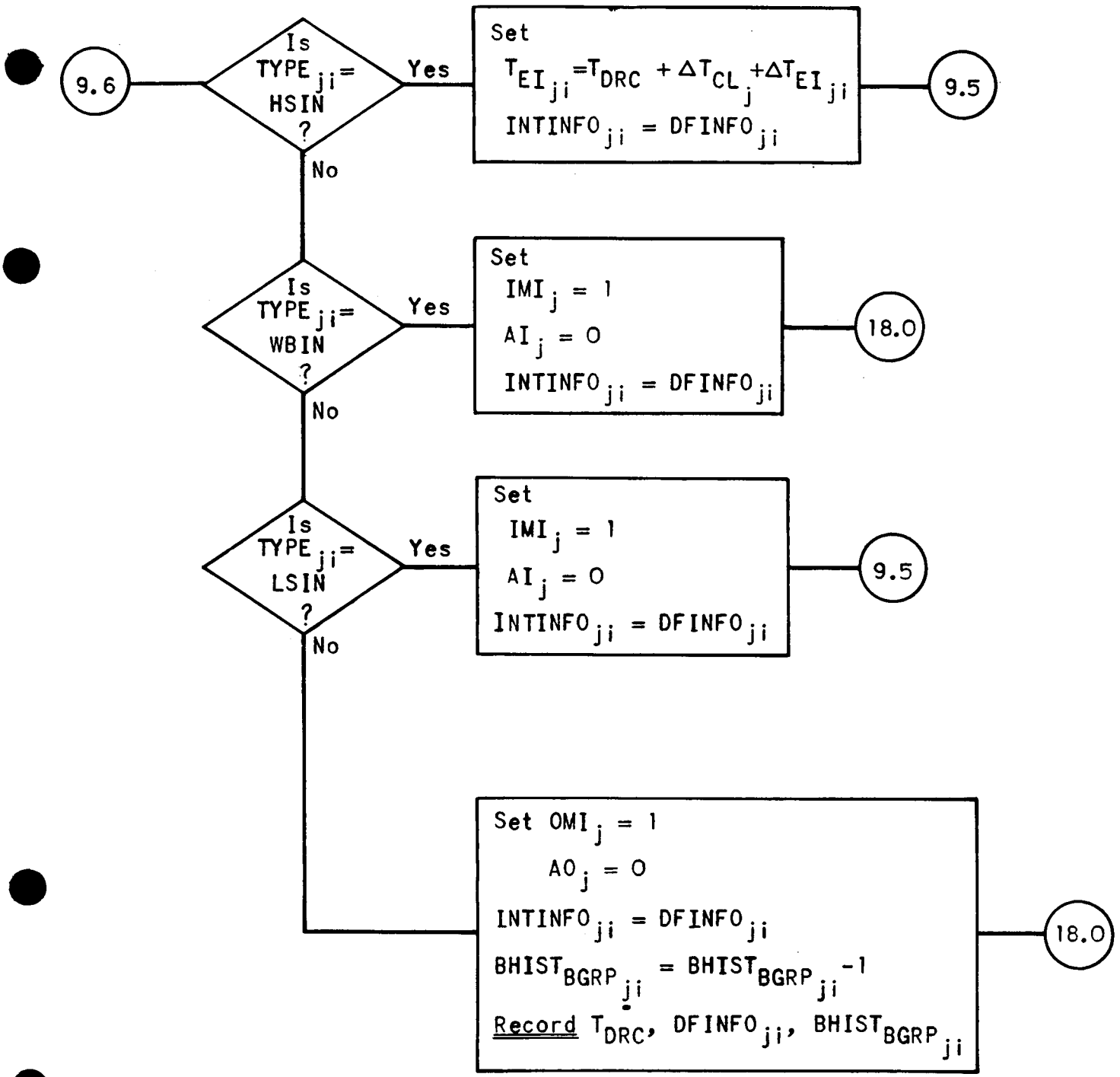
SCS Channel: Compute CAT and set the next data request.
UNIVAC 494 Algorithm
FIGURE A9-4



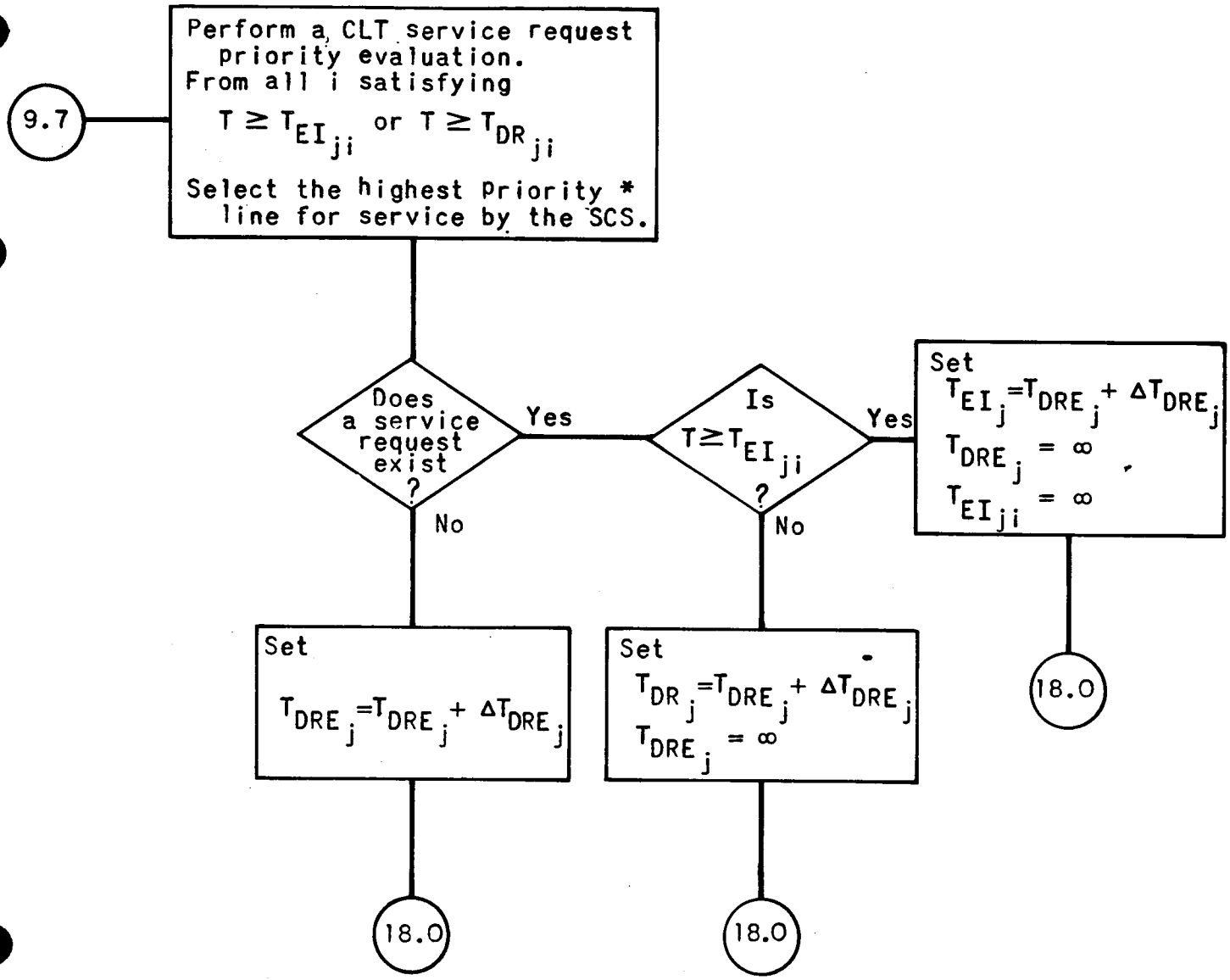
SCS Channel:
UNIVA



Set the next data frame.
C 494 Algorithm
FIGURE A9-5



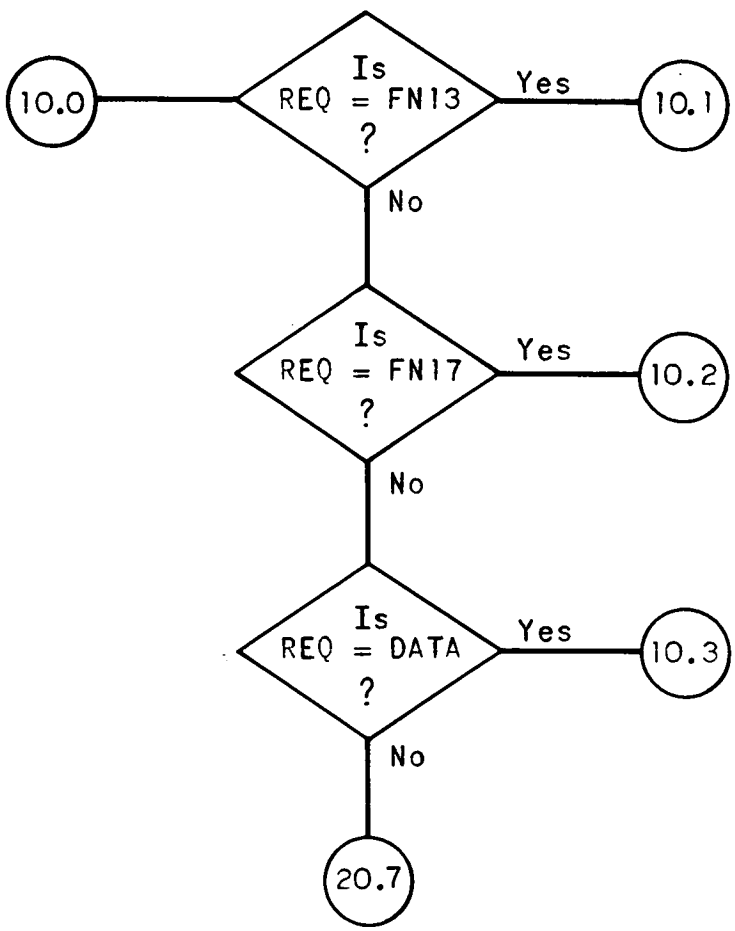
SCS Channel: Set an interrupt
UNIVAC 494 Algorithm
FIGURE A9-6



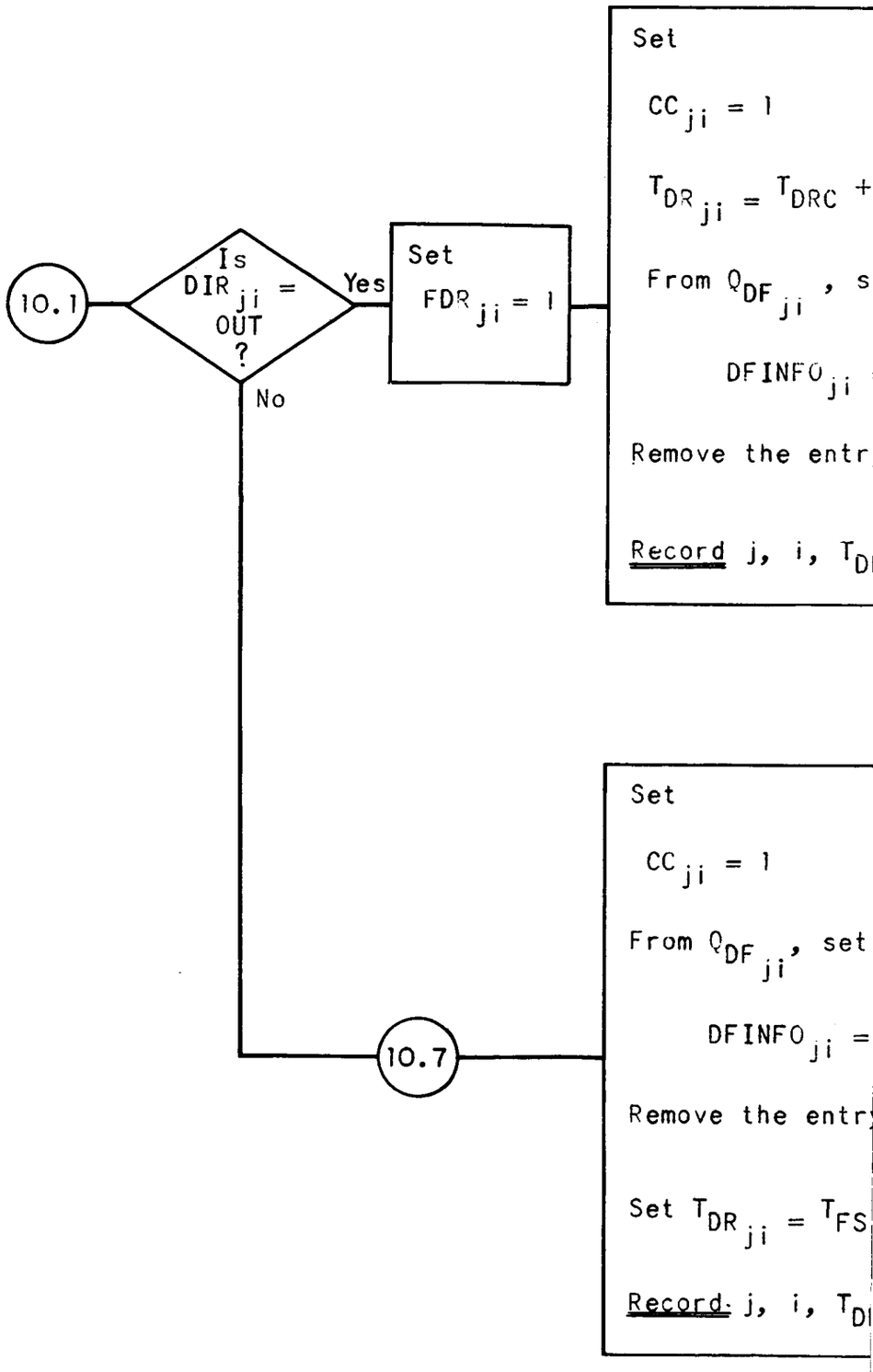
SCS Channel: Select a service request
UNIVAC 494 Algorithm

* See FIGURE A1-47

FIGURE A9-7



S/S Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A10-1



S/S Channel: P
UNIVAC
FI

122-1

$\Delta T_{FDR_{ji}}$

et

18.0

= QDF*

y from $Q_{DF_{ji}}$

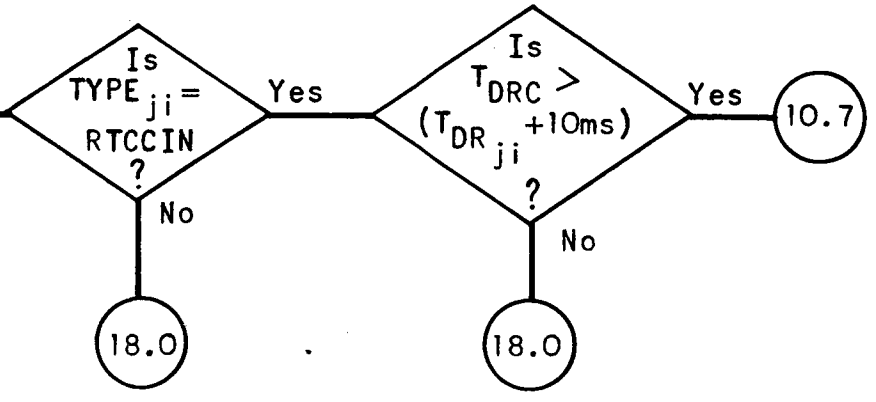
RC, REQ

QDF*

/ from $Q_{DF_{ji}}$

$_{ji}$ + $T_{REF_{ji}}$

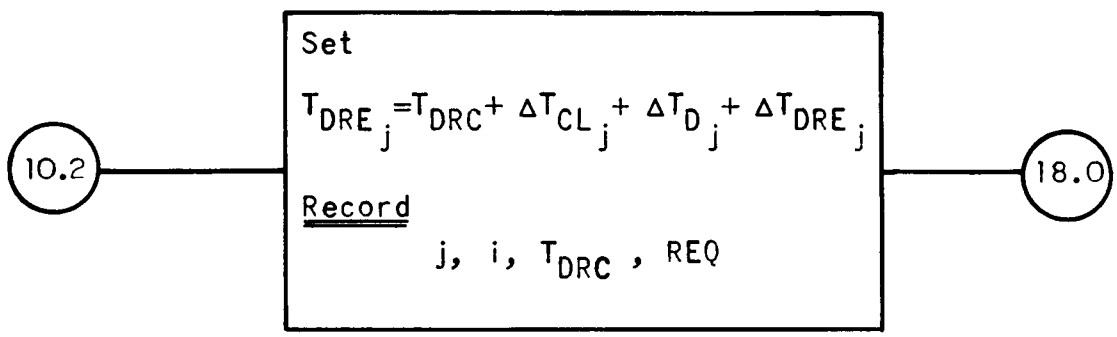
RC, REQ



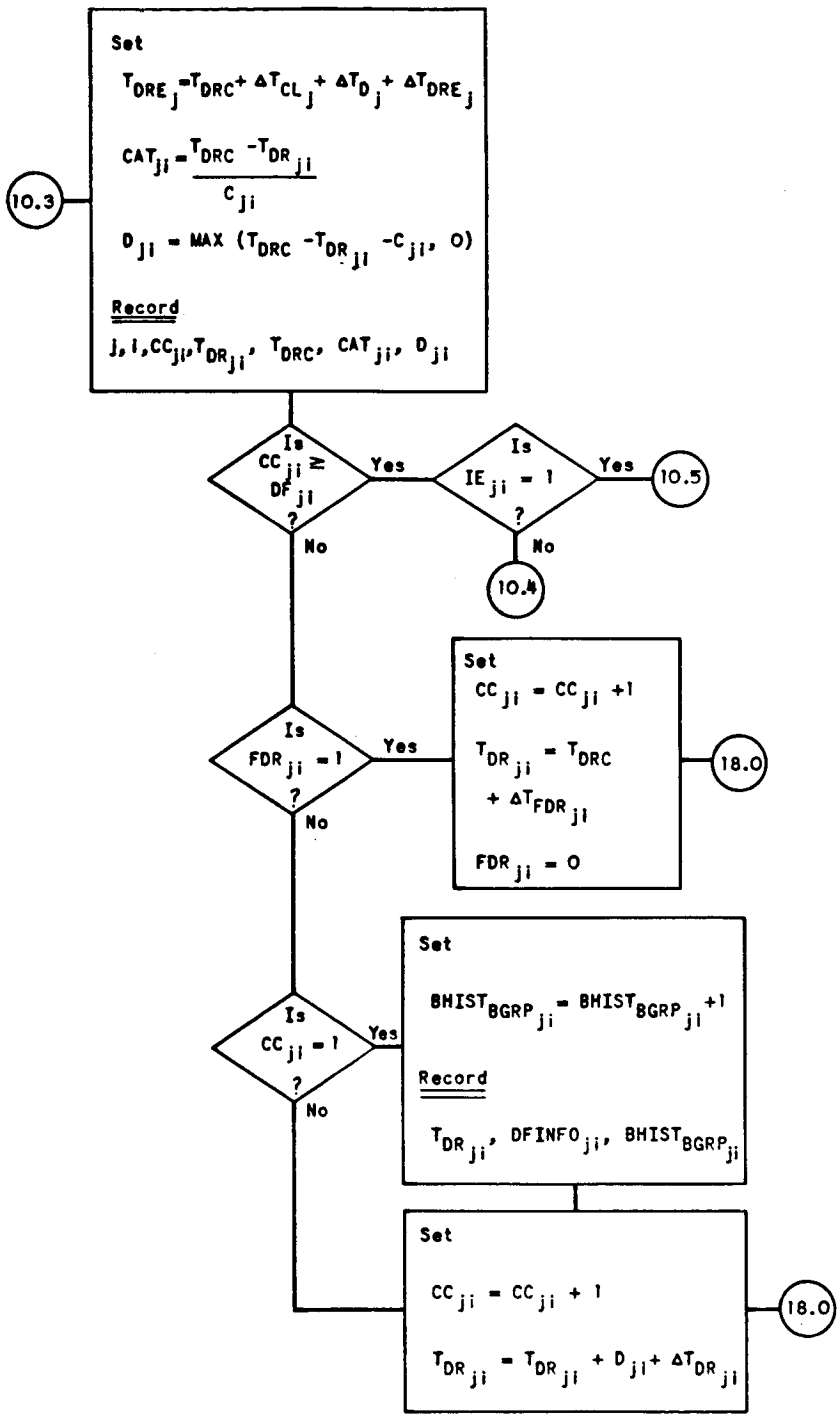
erform a FN13 action.

494 Algorithm

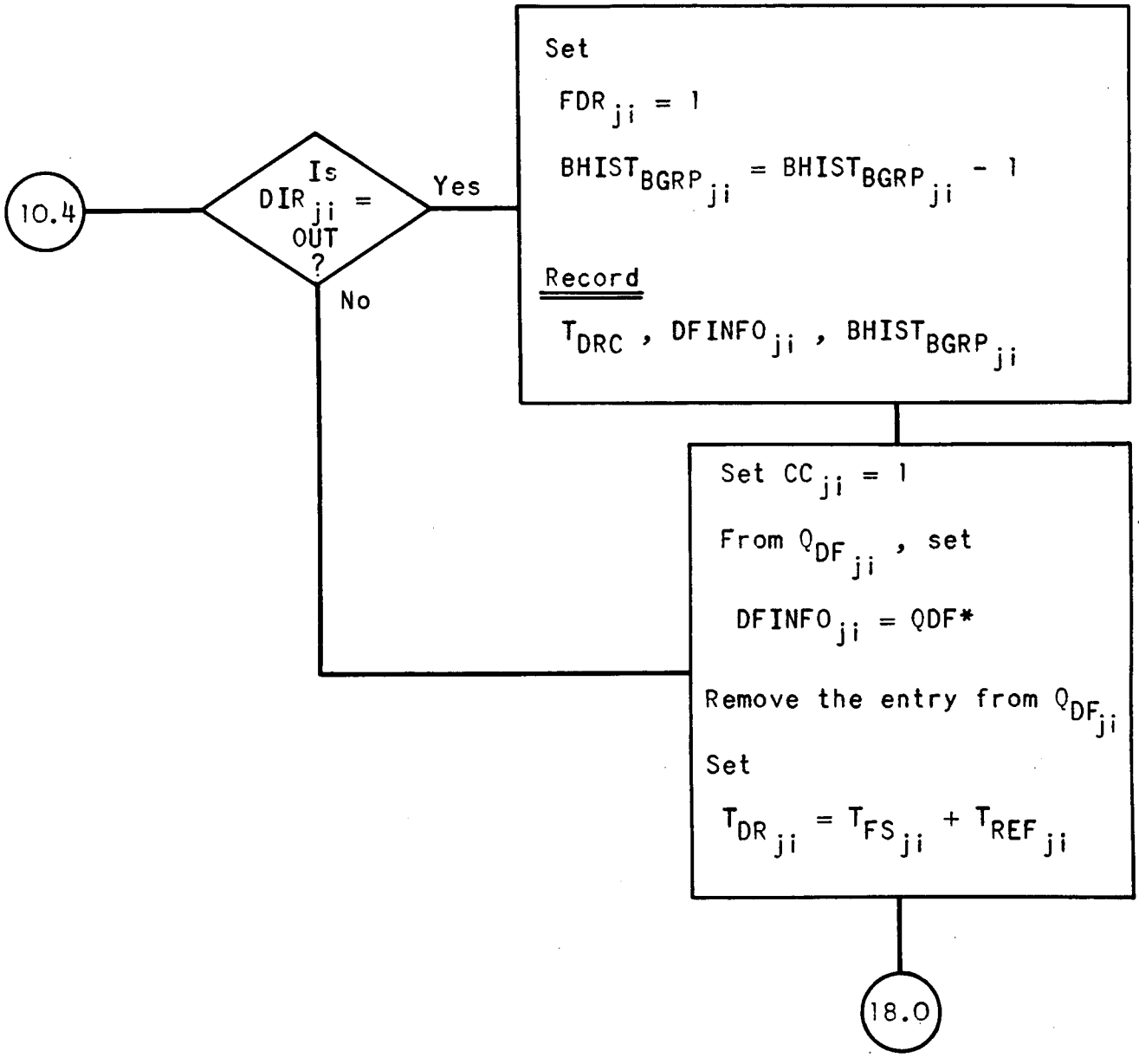
URE A10-2



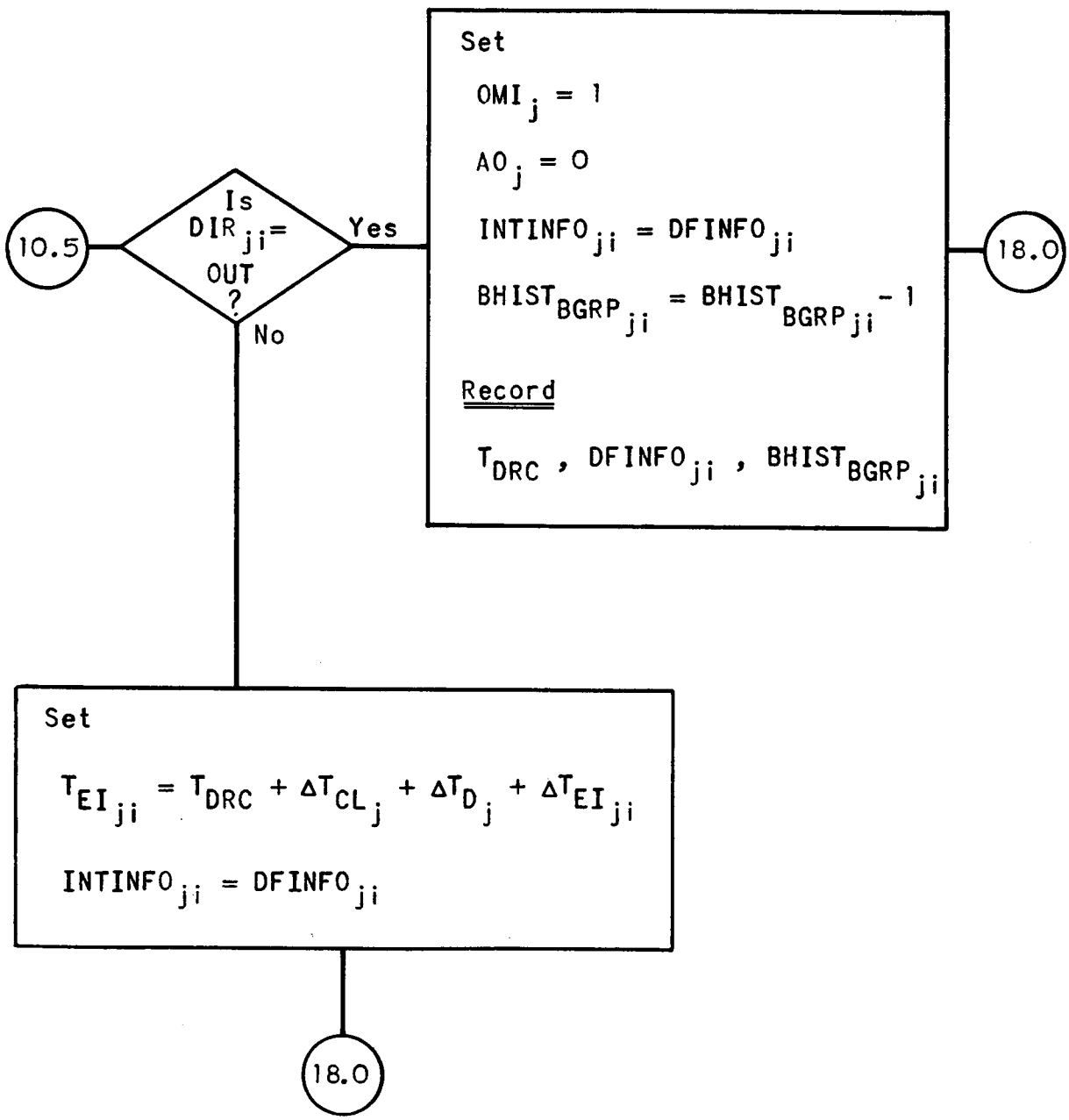
S/S Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A10-3



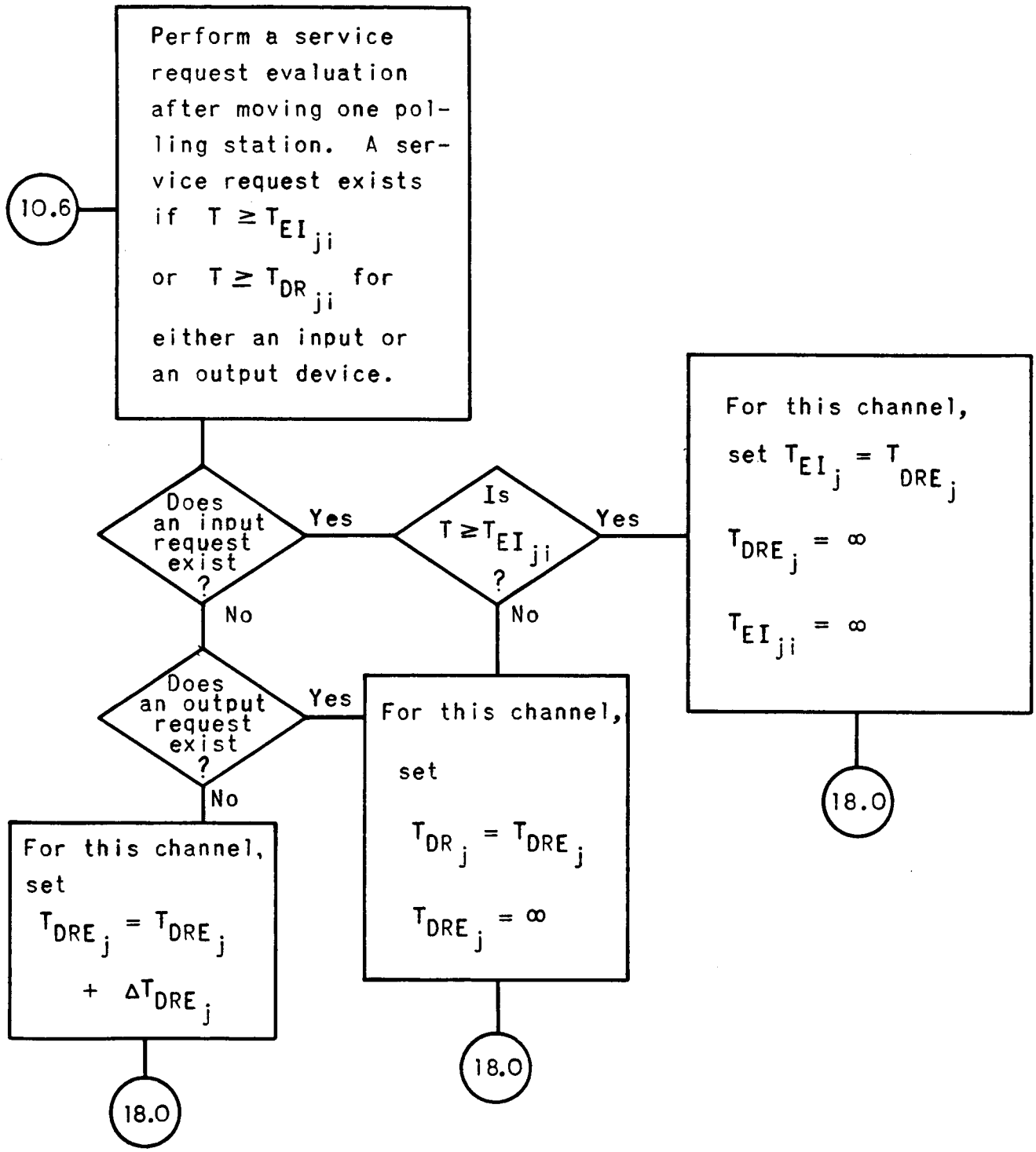
S/S Channel: Compute CAT and set the next data request.
UNIVAC 494 Algorithm
FIGURE A10-4



S/S Channel: Set the next data frame.
UNIVAC 494 Algorithm
FIGURE A10-5



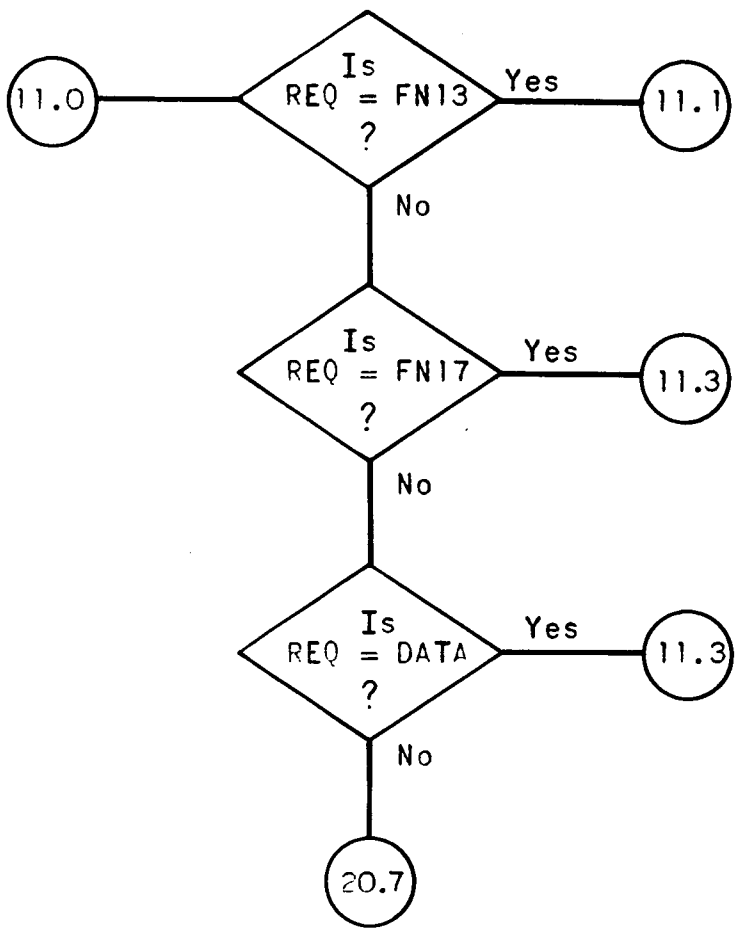
S/S Channel: Set an interrupt.
UNIVAC 494 Algorithm
FIGURE A10-6



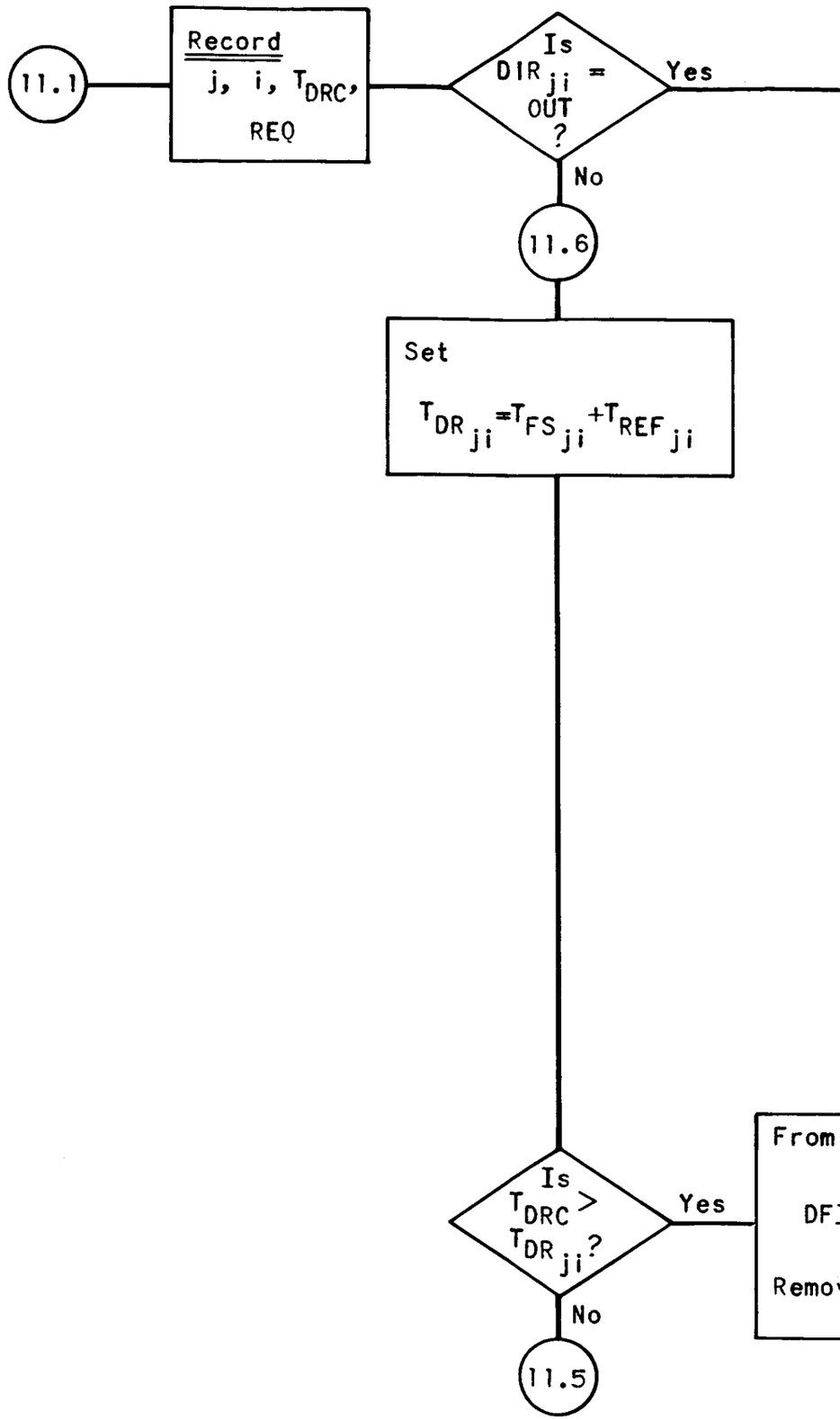
S/S Channel: Select a service request.

UNIVAC 494 Algorithm

FIGURE A10-7



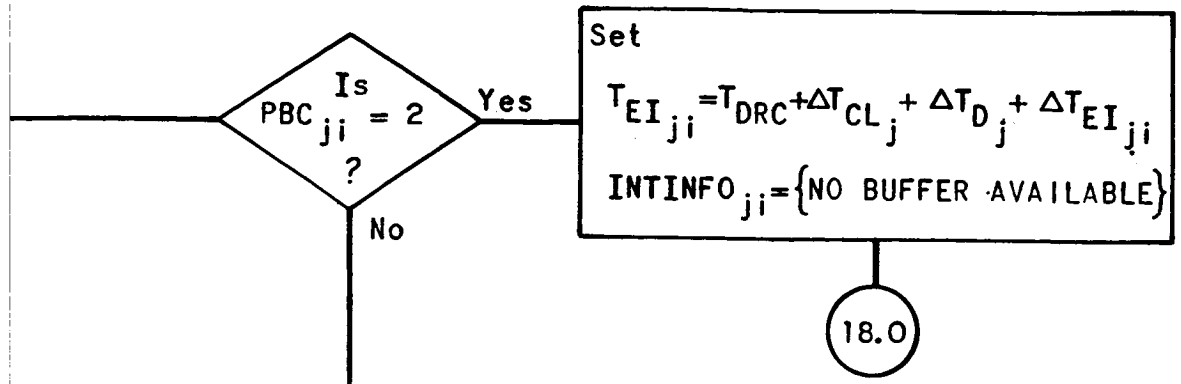
PBT Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A11-1

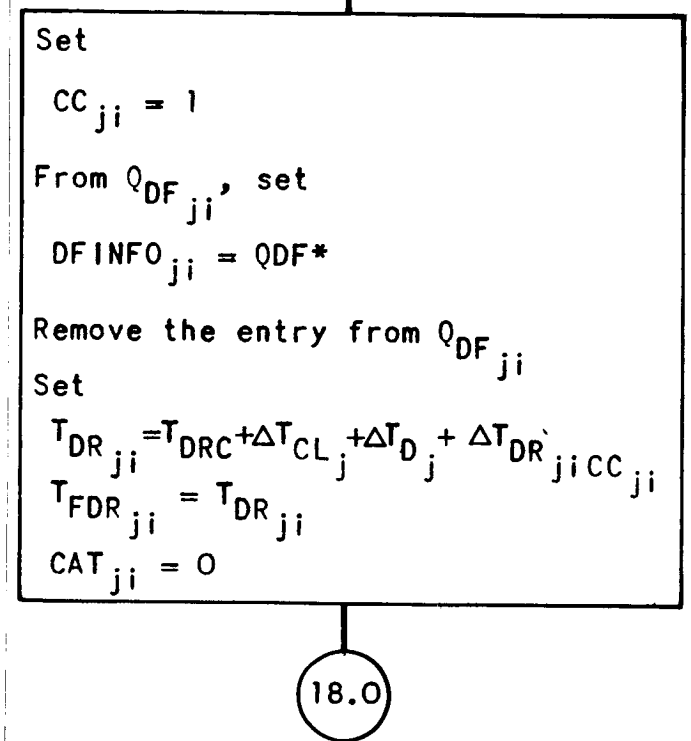


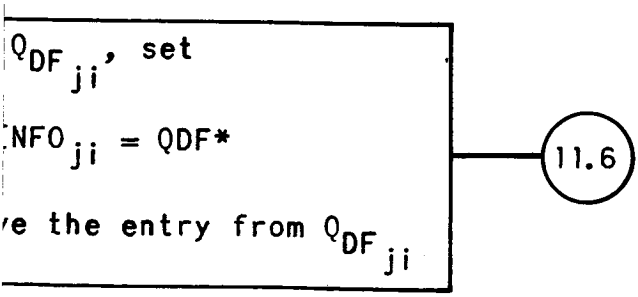
PBT Channel:

UNIVAC

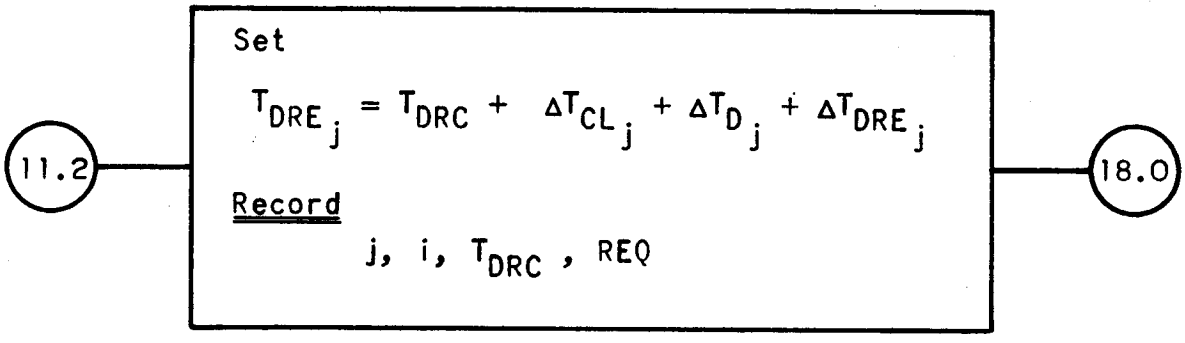
FIG



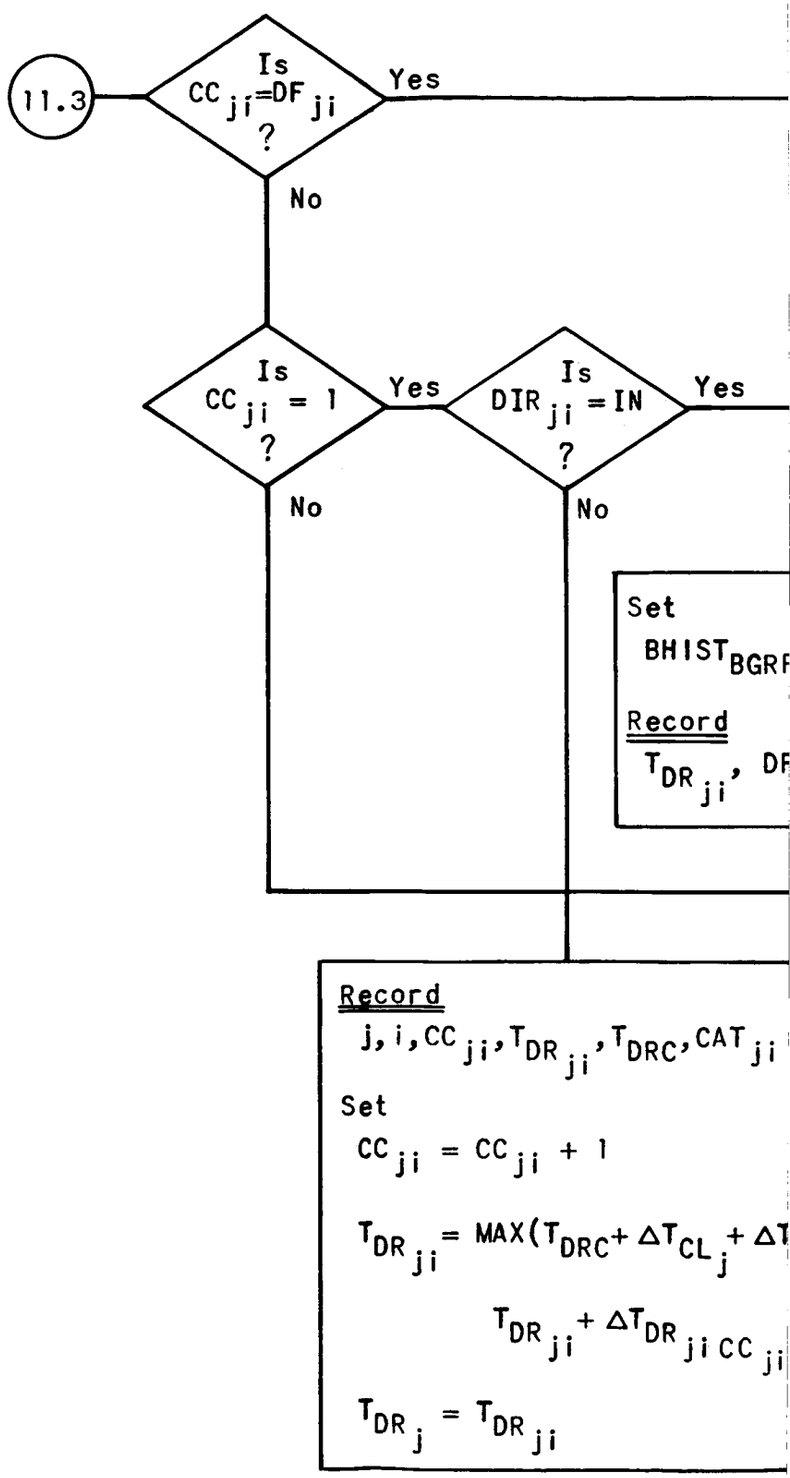




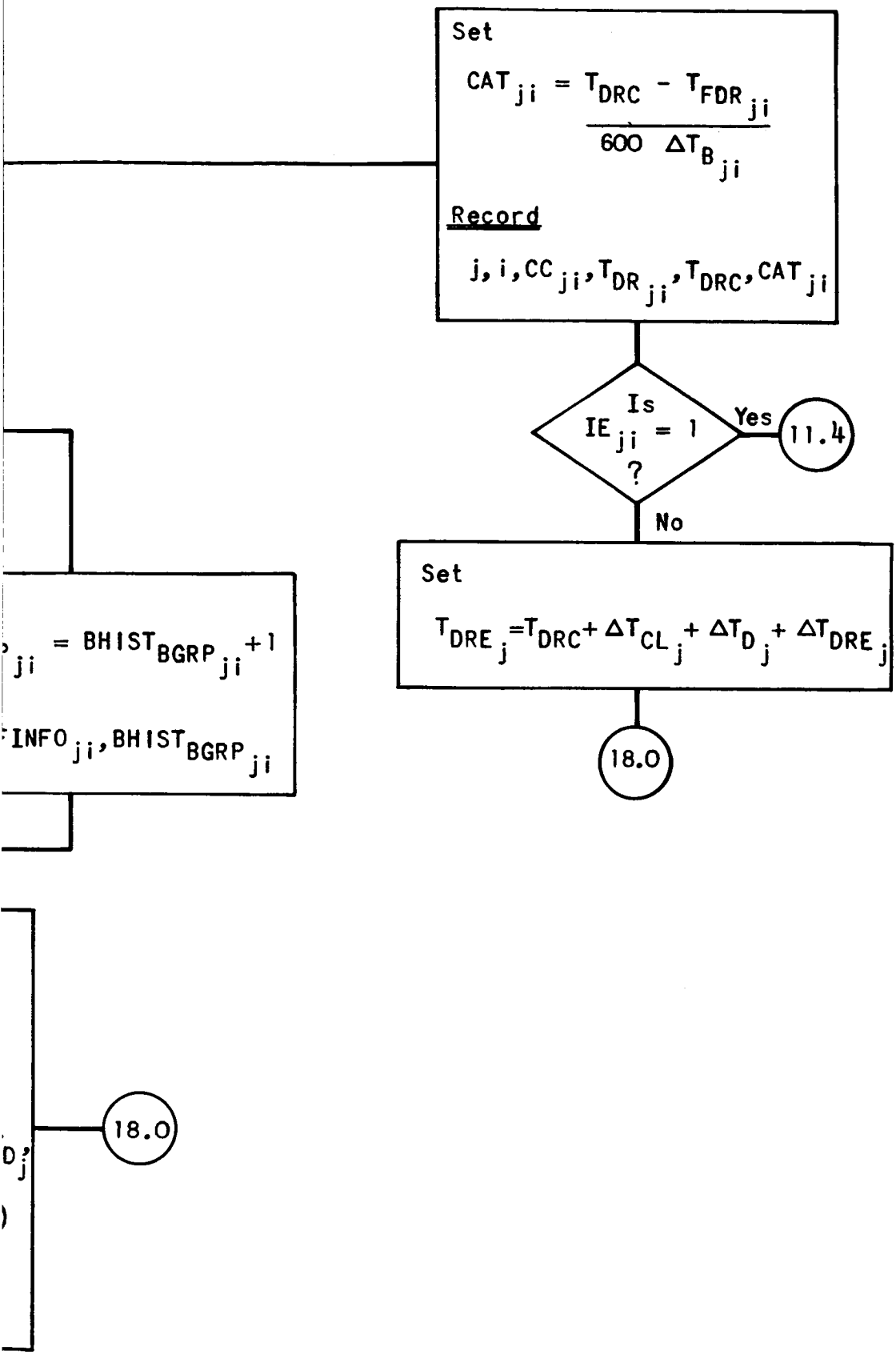
Perform a FN13 action.
494 Algorithm
JRE A11-2



PBT Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A11-3



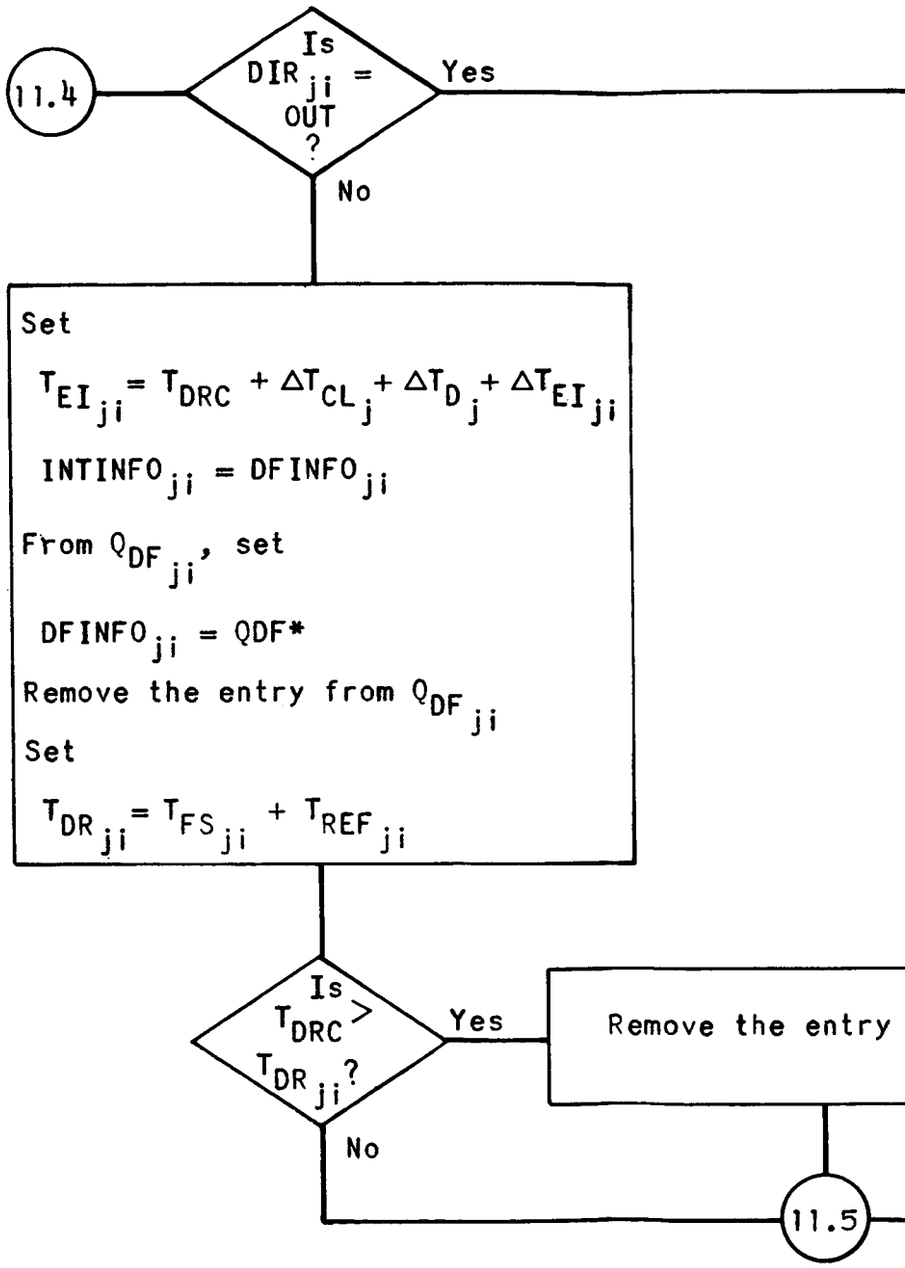
PBT Channel: Compute
UNI



CAT or set the next data request.

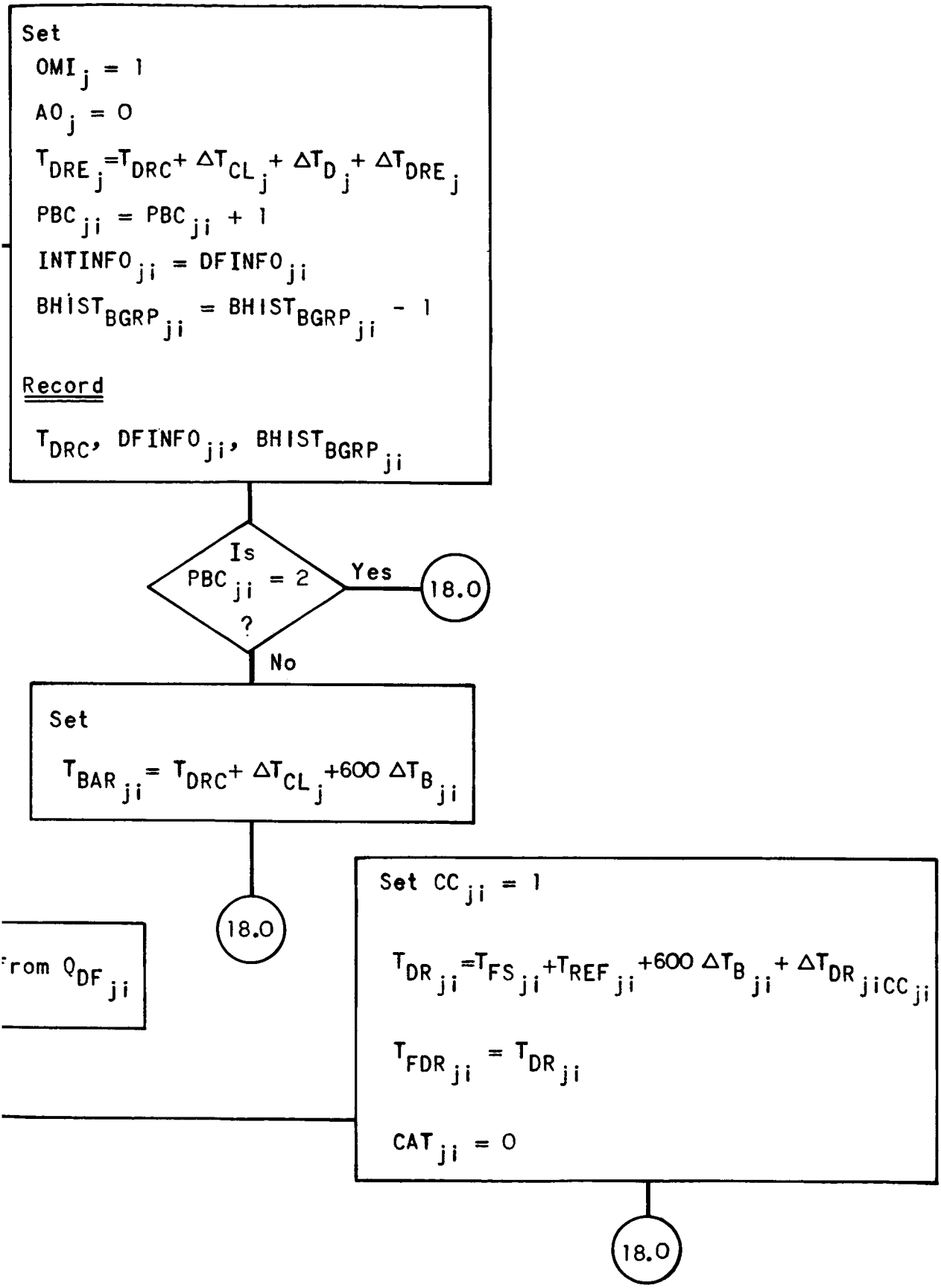
'AC 494 Algorithm

FIGURE A11-4

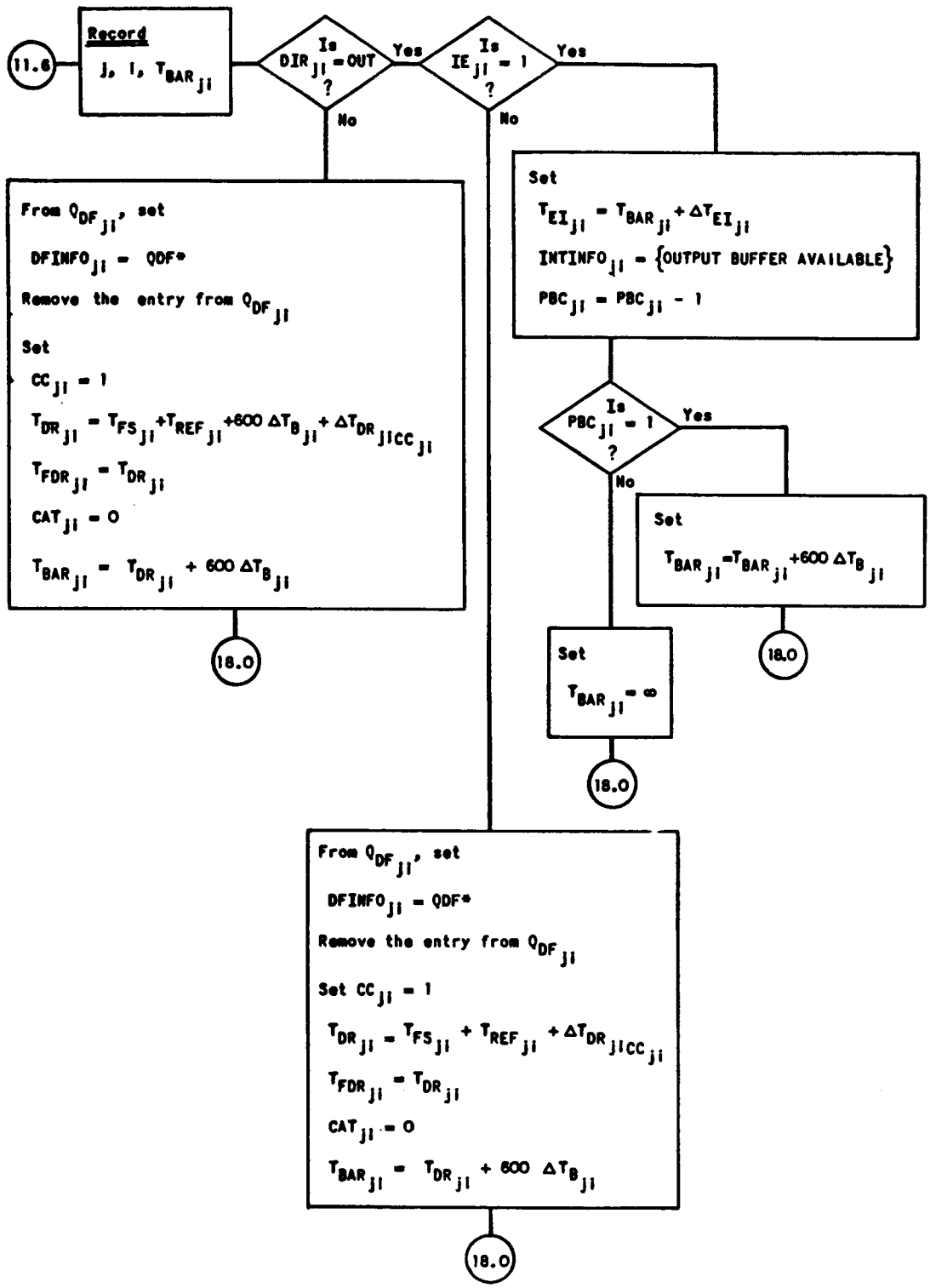


PBT Channel:
UNIVAC
FIGU

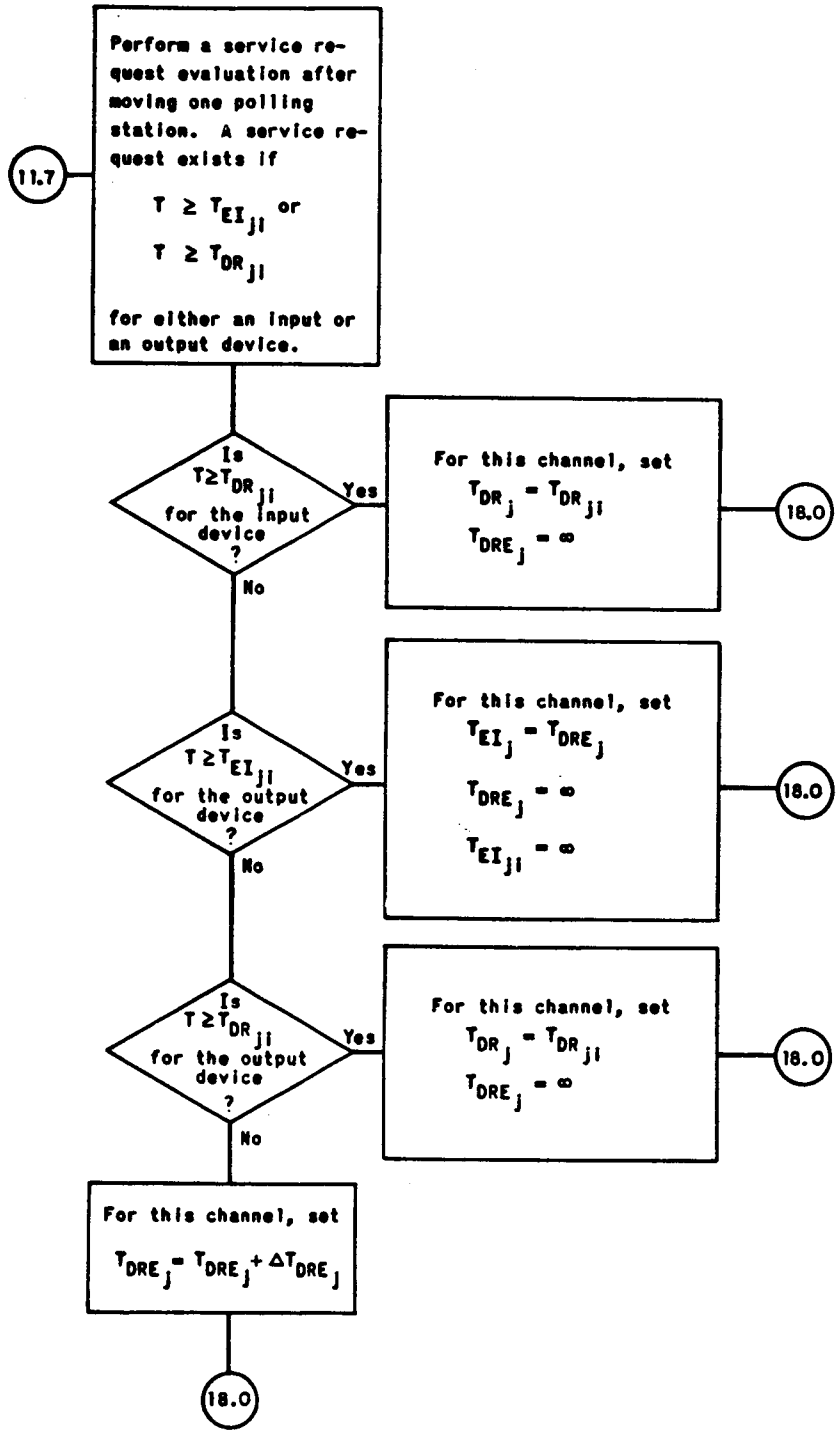
132-1



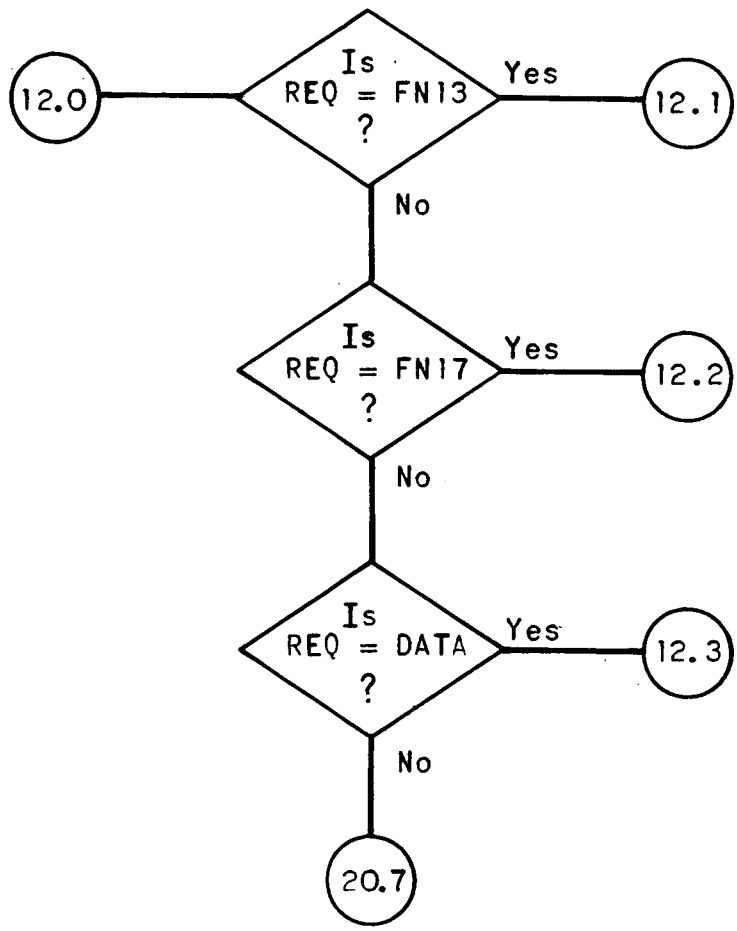
Set an interrupt.
94 Algorithm
RE A11-5



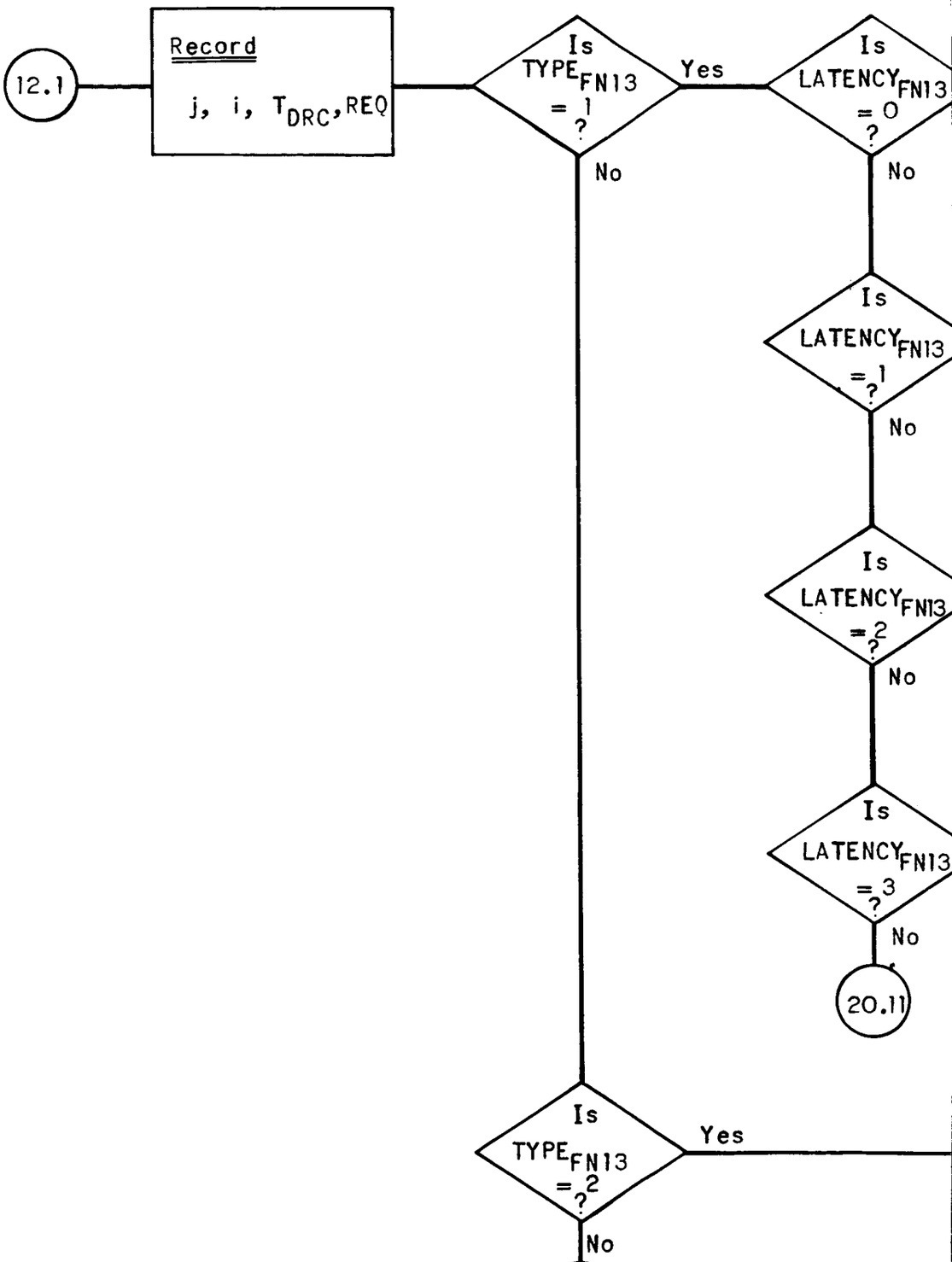
PBT Channel: Set the next buffer action request.
UNIVAC 494 Algorithm
FIGURE A11-6



PBT Channel: Select a service request.
UNIVAC 494 Algorithm
FIGURE A11-7



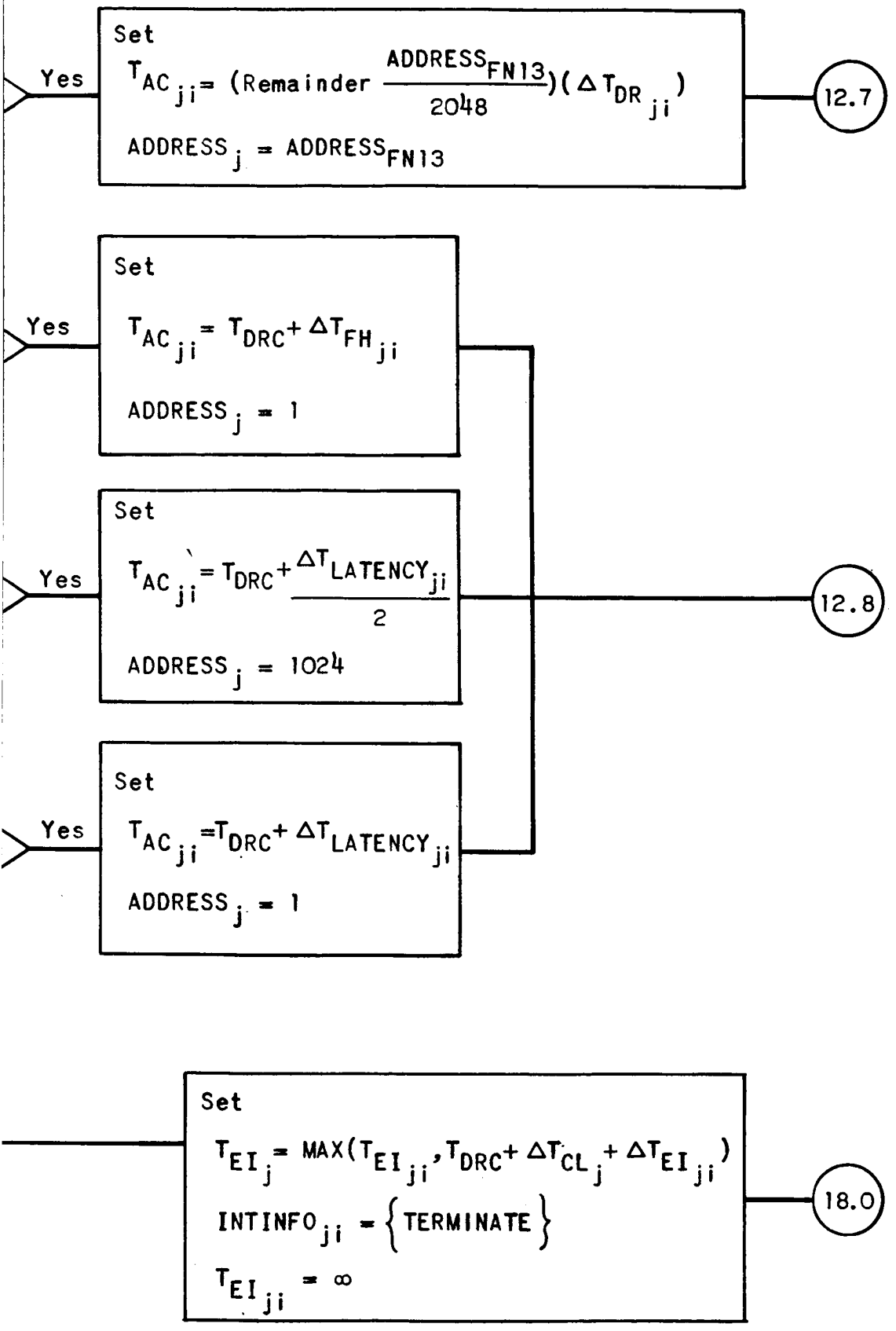
FH880 Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A12-1



FH880 Channel: Perform a
UNIVAC 494 Algori

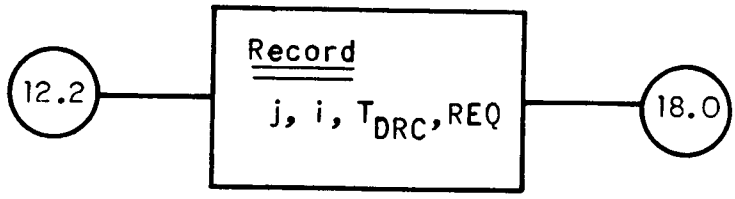
136-1

FIGURE A12-2

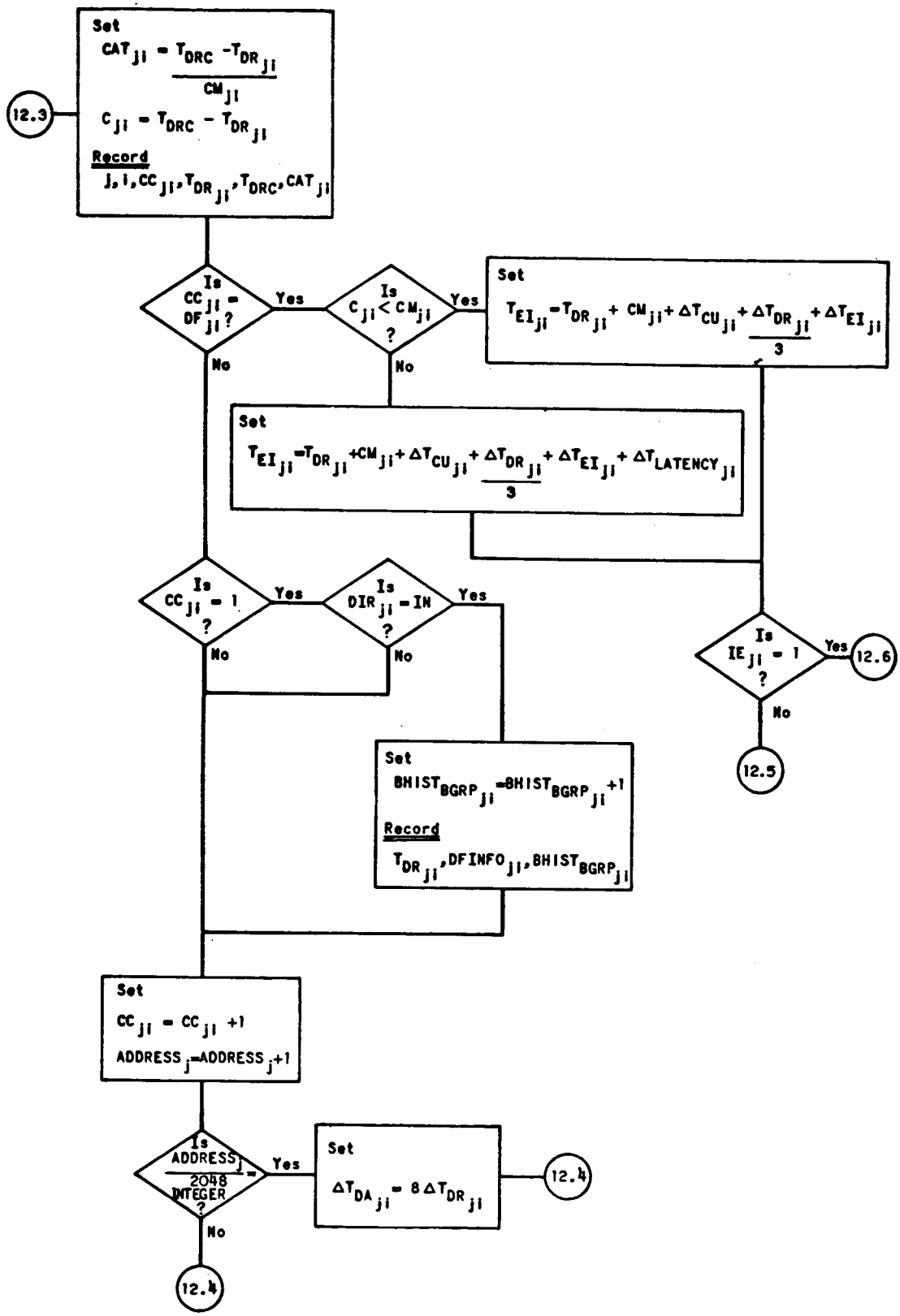


FN13 action.

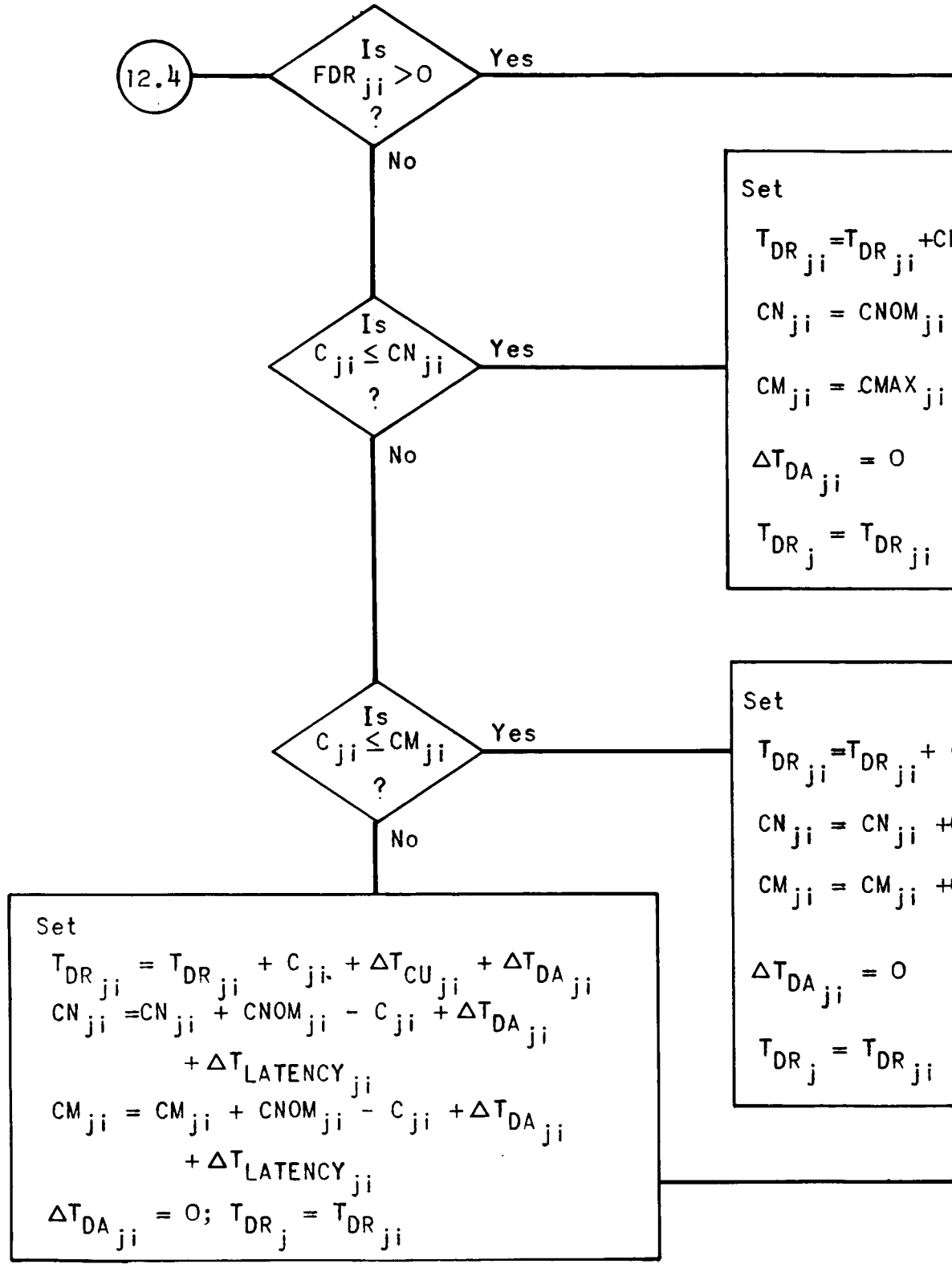
thm



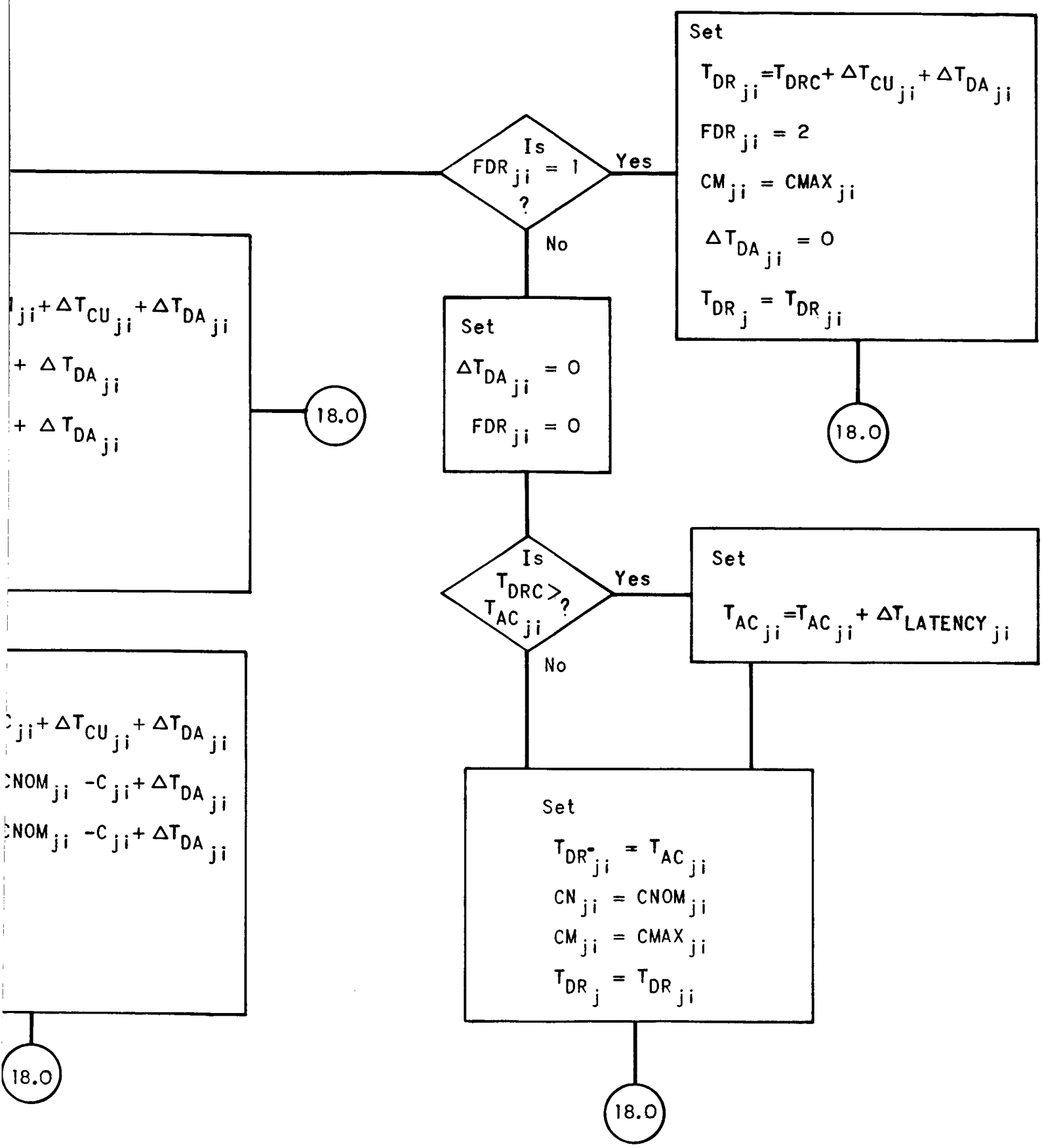
FH880 Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A12-3



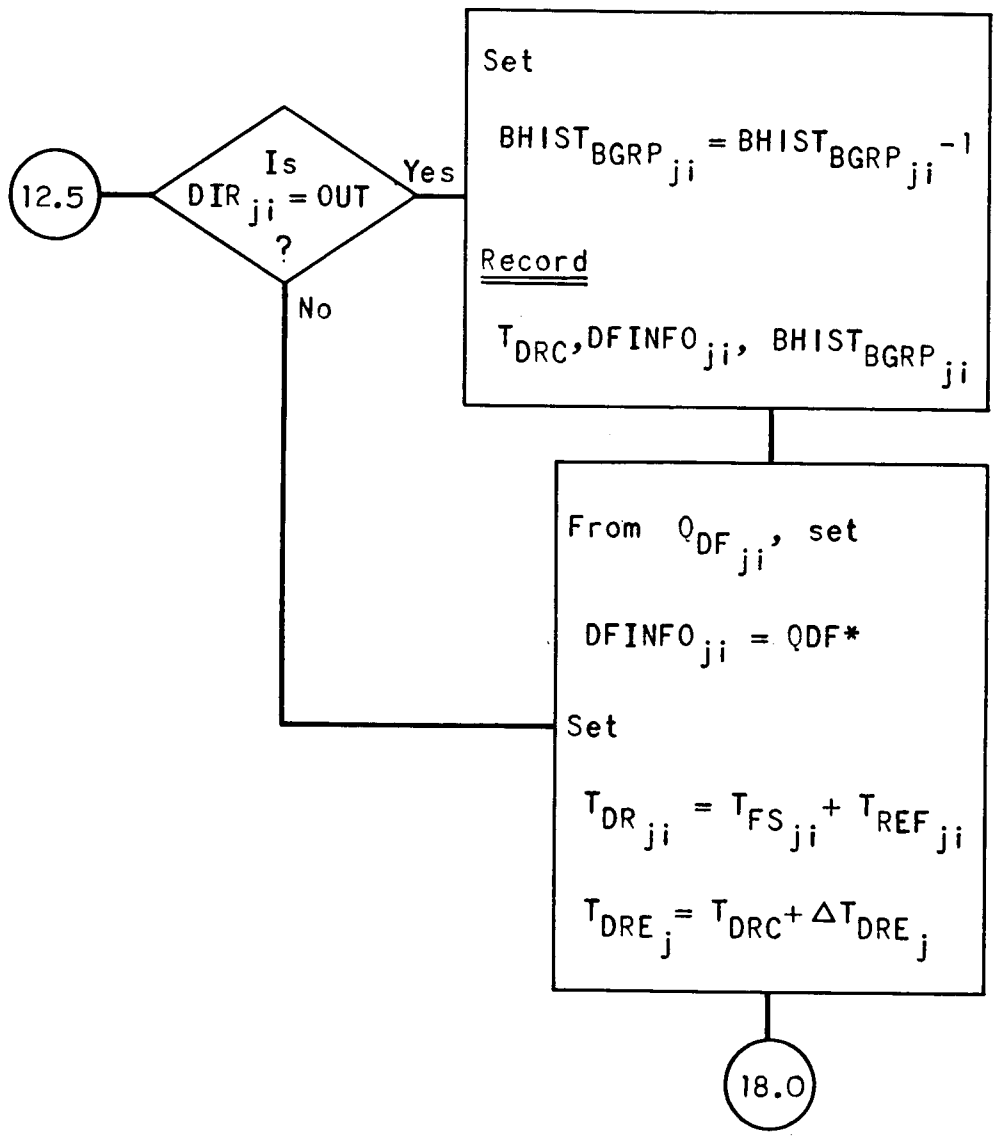
FH880 Channel: Compute CAT.
UNIVAC 494 Algorithm
FIGURE A12-4


12.4
Is $FDR_{ji} > 0$?
Yes
No
Set
$T_{DR_{ji}} = T_{DR_{ji}} + CI$
$CN_{ji} = CNOM_{ji}$
$CM_{ji} = CMAX_{ji}$
$\Delta T_{DA_{ji}} = 0$
$T_{DR_j} = T_{DR_{ji}}$
Is $C_{ji} \leq CN_{ji}$?
Yes
No
Set
$T_{DR_{ji}} = T_{DR_{ji}} +$
$CN_{ji} = CN_{ji} +$
$CM_{ji} = CM_{ji} +$
$\Delta T_{DA_{ji}} = 0$
$T_{DR_j} = T_{DR_{ji}}$
Is $C_{ji} \leq CM_{ji}$?
Yes
No
Set
$T_{DR_{ji}} = T_{DR_{ji}} + C_{ji} + \Delta T_{CU_{ji}} + \Delta T_{DA_{ji}}$
$CN_{ji} = CN_{ji} + CNOM_{ji} - C_{ji} + \Delta T_{DA_{ji}} + \Delta T_{LATENCY_{ji}}$
$CM_{ji} = CM_{ji} + CNOM_{ji} - C_{ji} + \Delta T_{DA_{ji}} + \Delta T_{LATENCY_{ji}}$
$\Delta T_{DA_{ji}} = 0$; $T_{DR_j} = T_{DR_{ji}}$

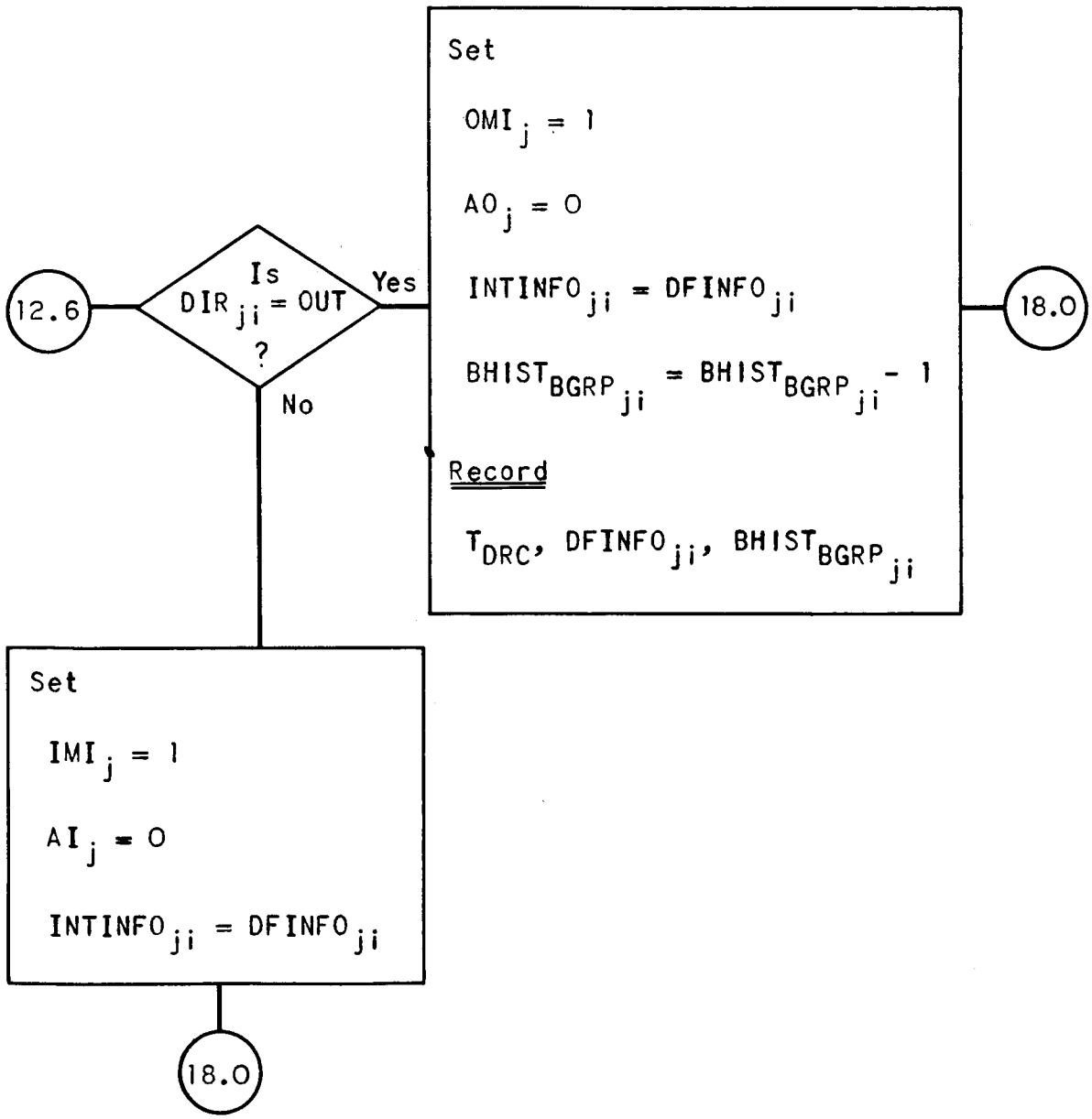
FH880 Cha
UN



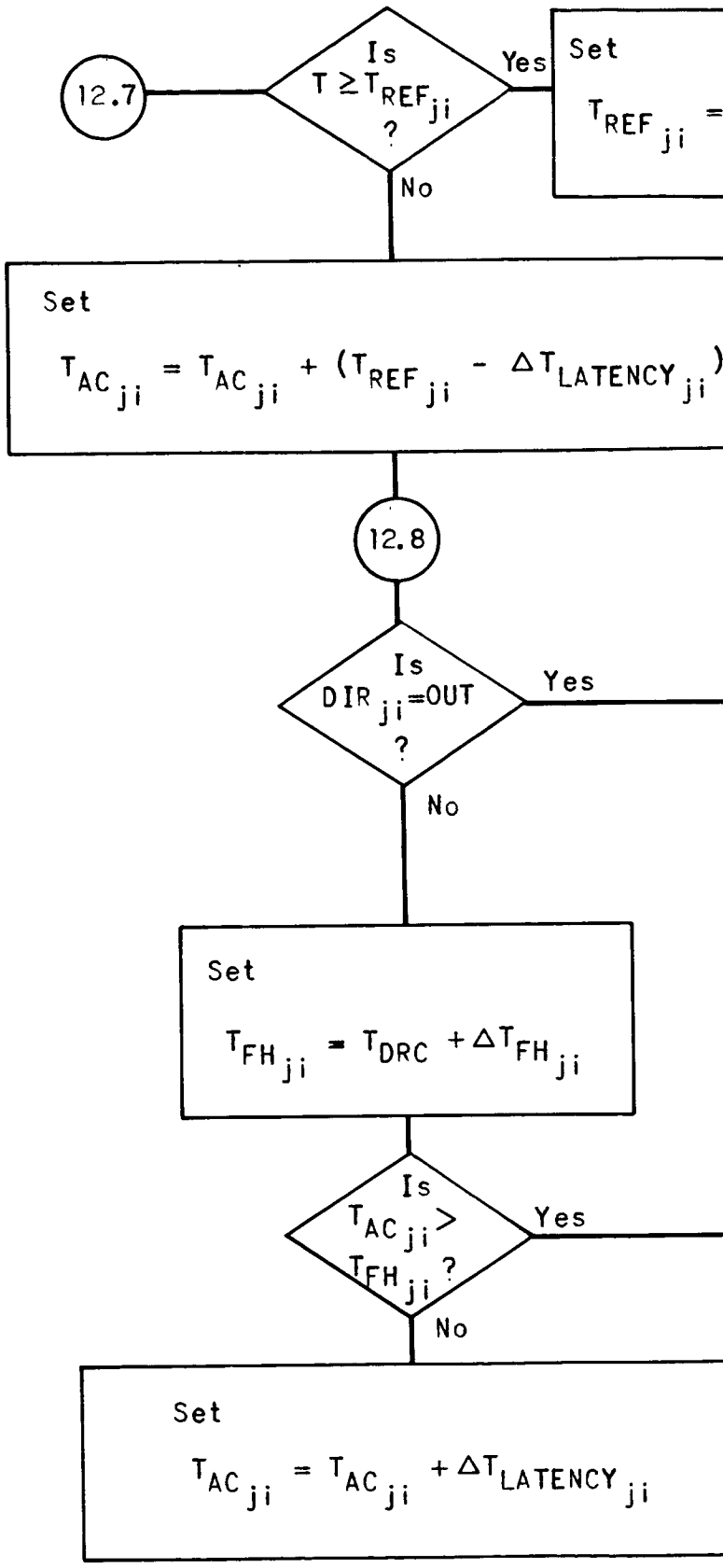
nnel: Set a data request
IVAC 494 Algorithm
FIGURE A12-5

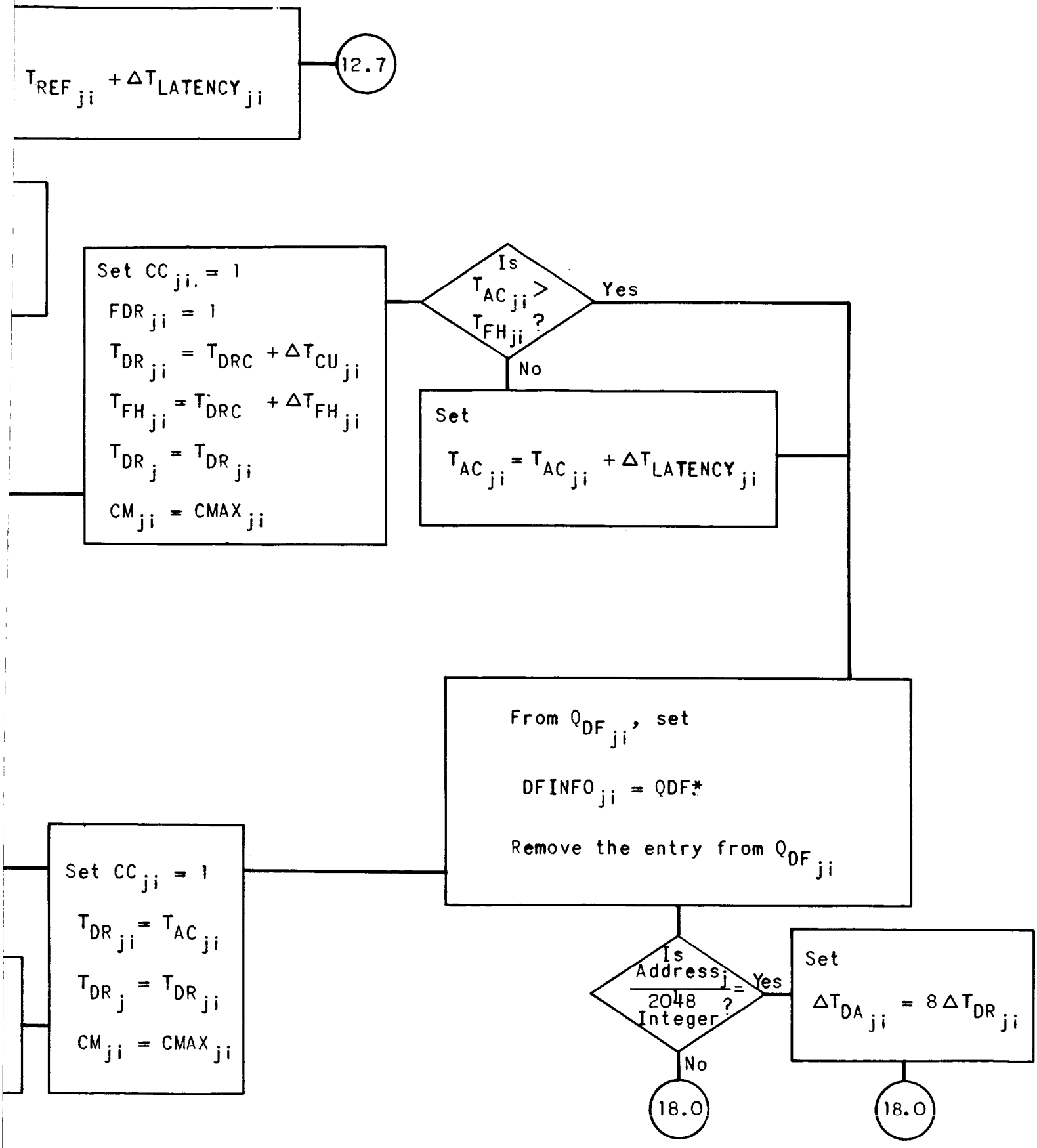


FH880 Channel: Set the next data request evaluation.
UNIVAC 494 Algorithm
FIGURE A12-6

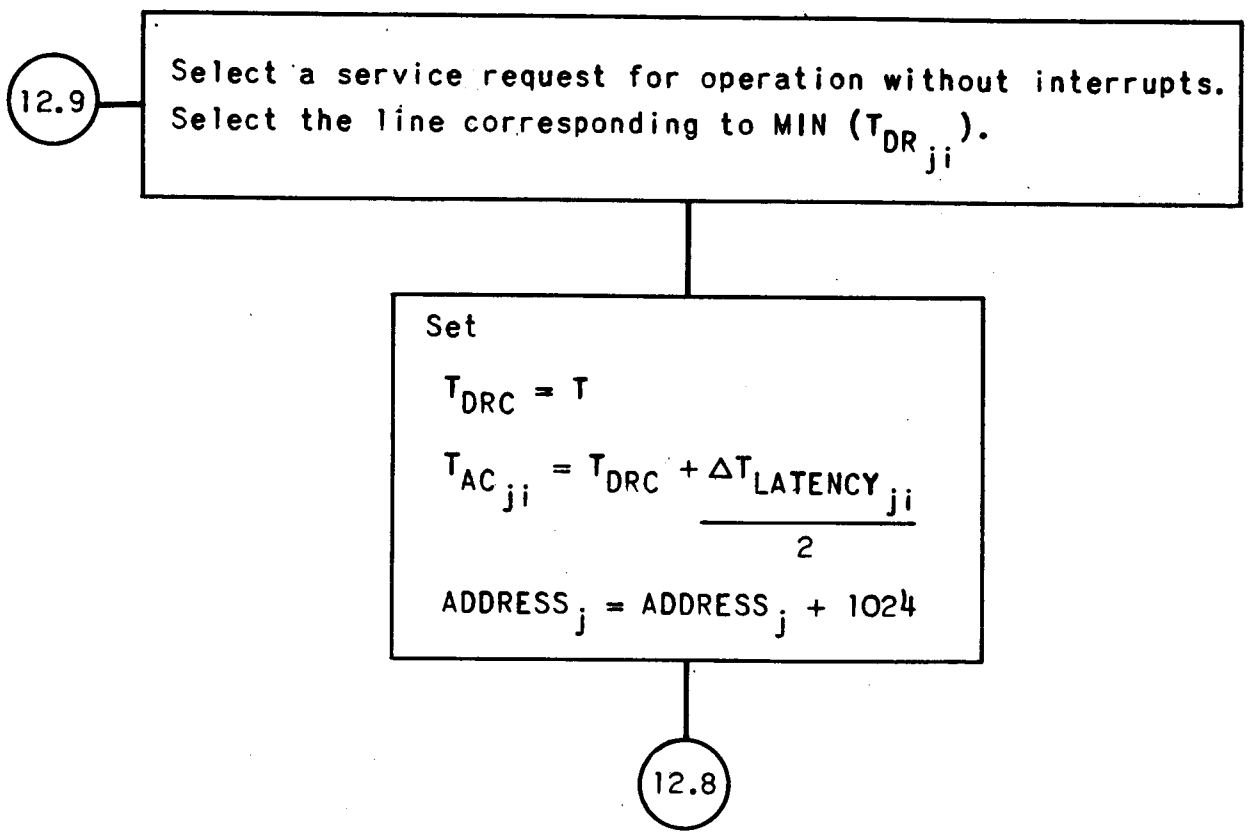


FH880 Channel: Set an interrupt.
UNIVAC 494 Algorithm
FIGURE A12-7

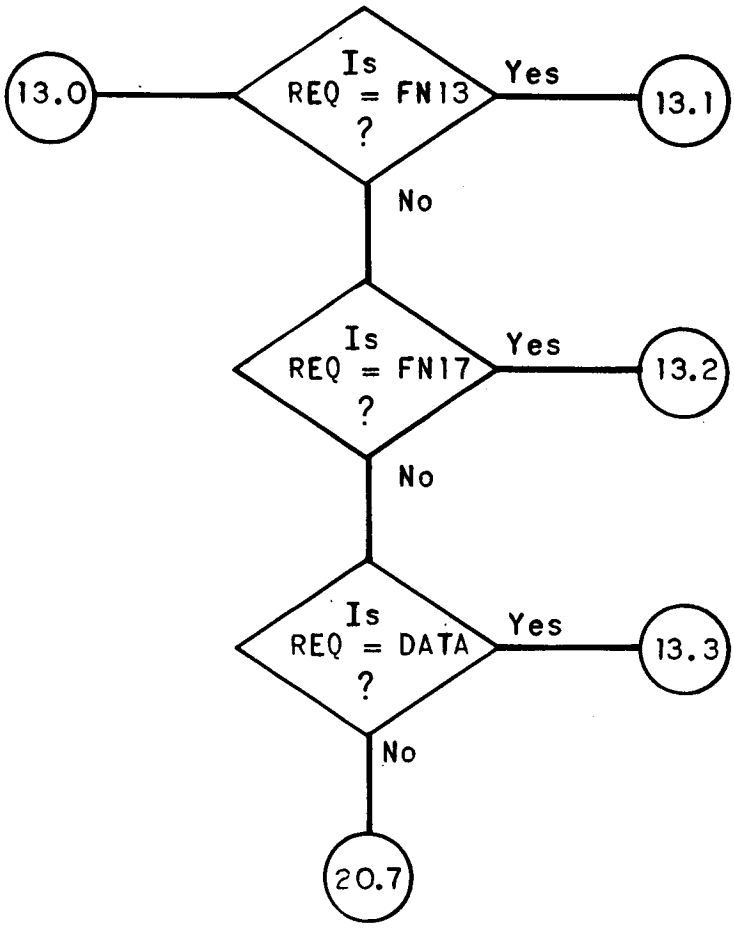

12.7
Is $T \geq T_{REF_{ji}}$?
Yes
Set $T_{REF_{ji}}$ =
No
Set
$T_{AC_{ji}} = T_{AC_{ji}} + (T_{REF_{ji}} - \Delta T_{LATENCY_{ji}})$
12.8
Is DIR_{ji}=OUT ?
Yes
No
Set
$T_{FH_{ji}} = T_{DRC} + \Delta T_{FH_{ji}}$
Is $T_{AC_{ji}} > T_{FH_{ji}}$?
Yes
No
Set
$T_{AC_{ji}} = T_{AC_{ji}} + \Delta T_{LATENCY_{ji}}$



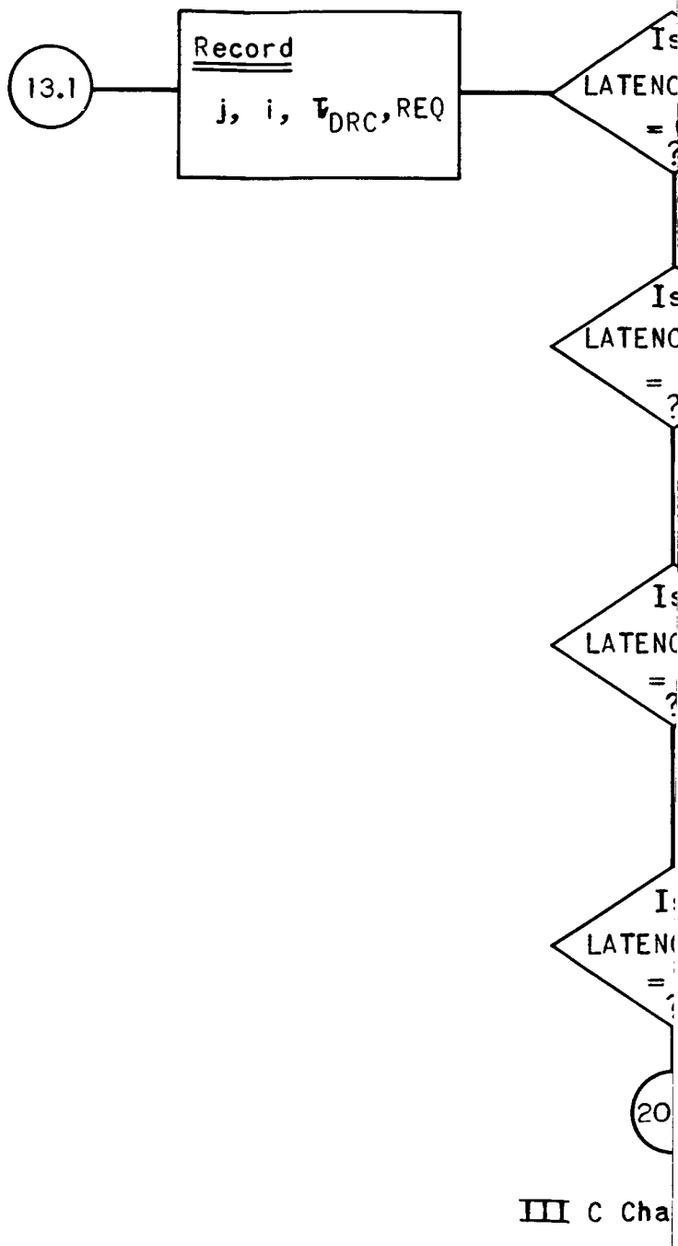
H880 Channel: Set the next data frame.
UNIVAC 494 Algorithm
FIGURE A12-8



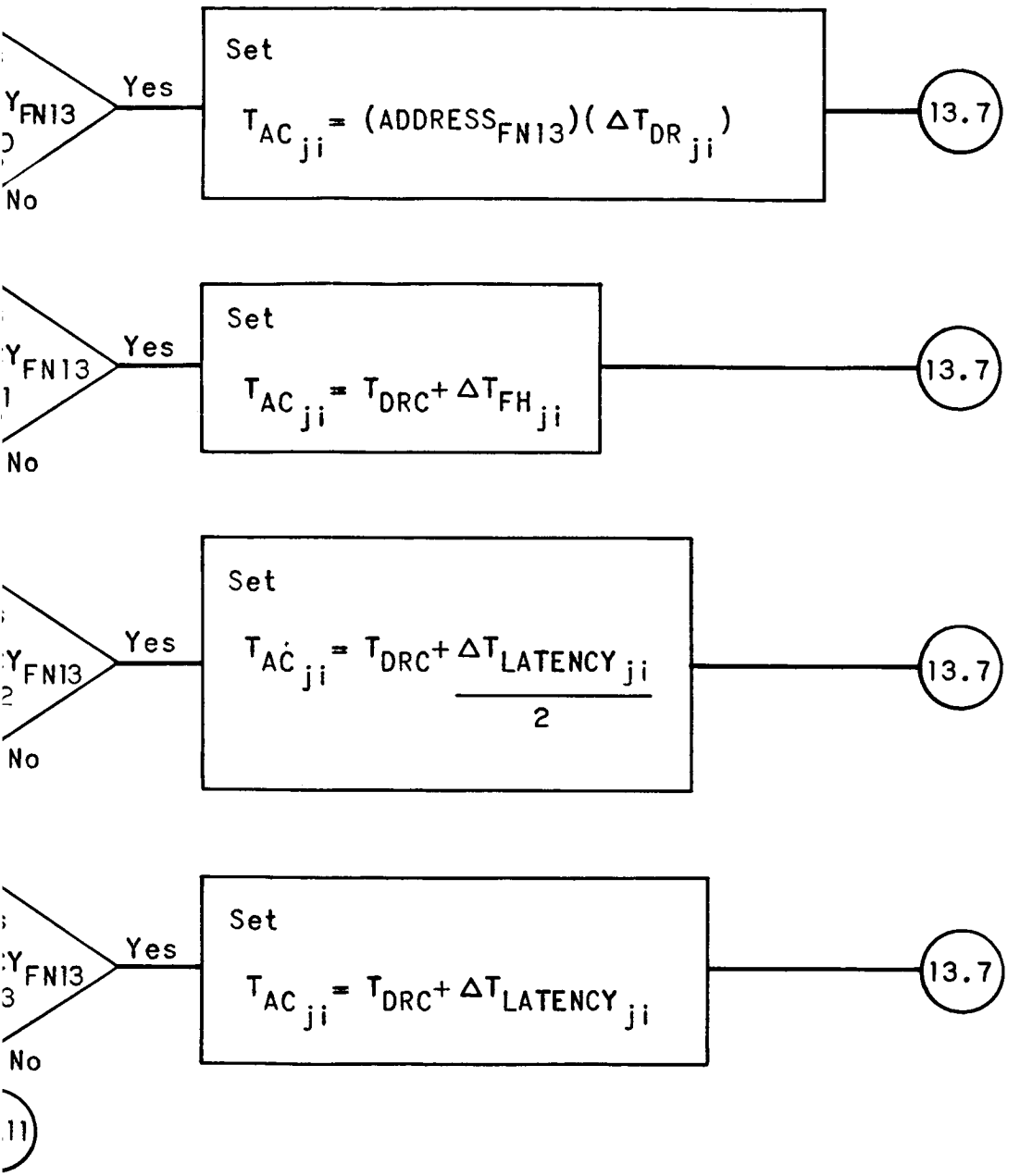
FH880 Channel: Select a service request.
UNIVAC 494 Algorithm
FIGURE A12-9



III C Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A13-1

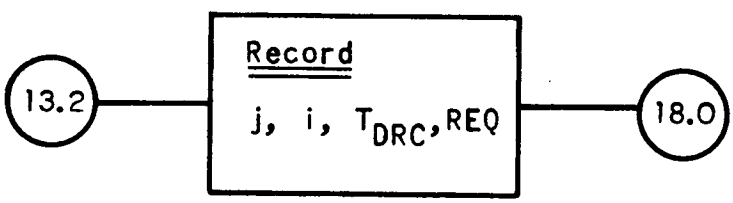

13.1
Record
j, i, τ_{DRC}, REQ
Is
LATENC
=
Is
LATENC
=
Is
LATENC
=
I
LATENC
=
20

III C Cha

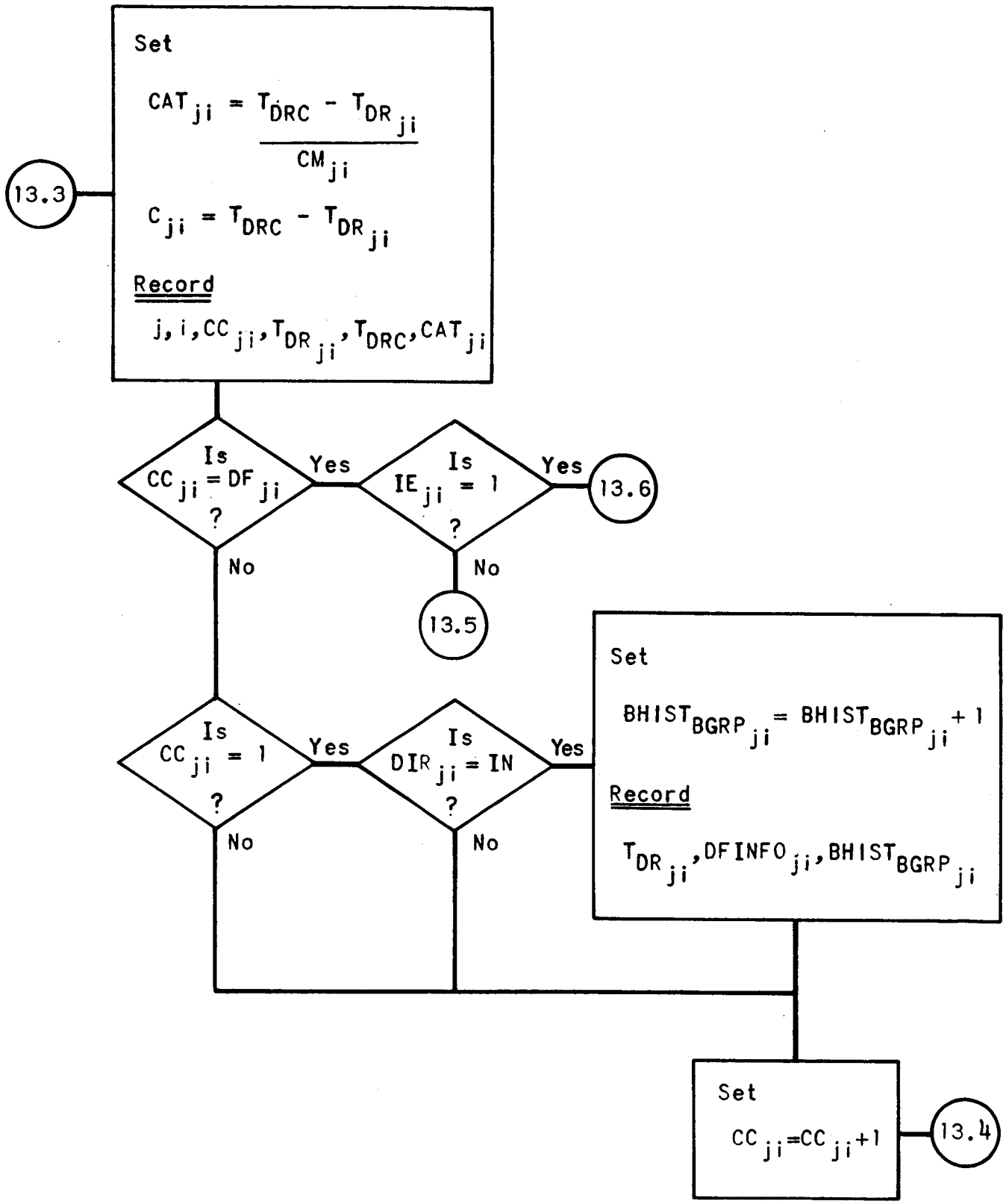


inel: Perform a FN13 action.
UNIVAC 494 Algorithm

FIGURE A13-2

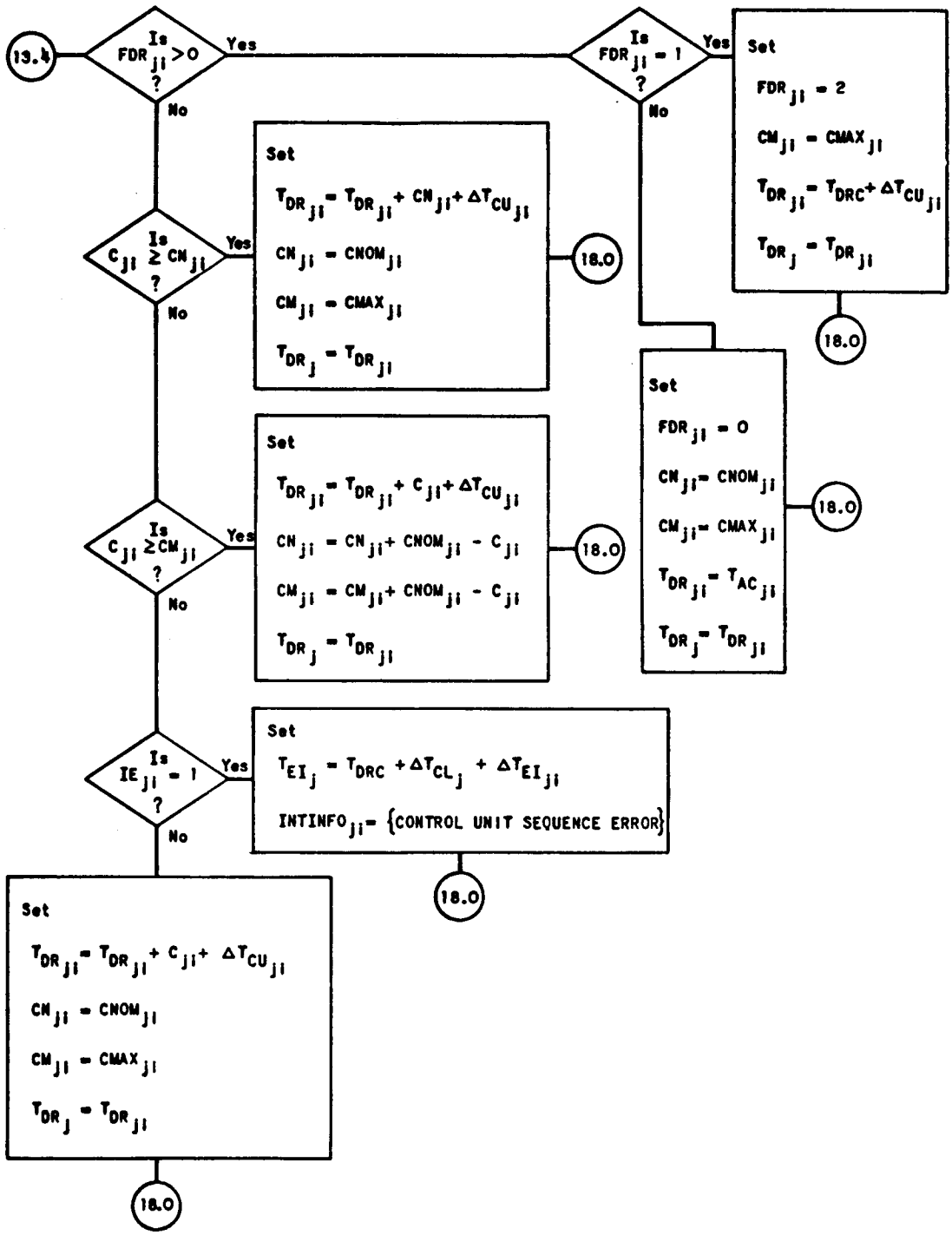


III C Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A13-3

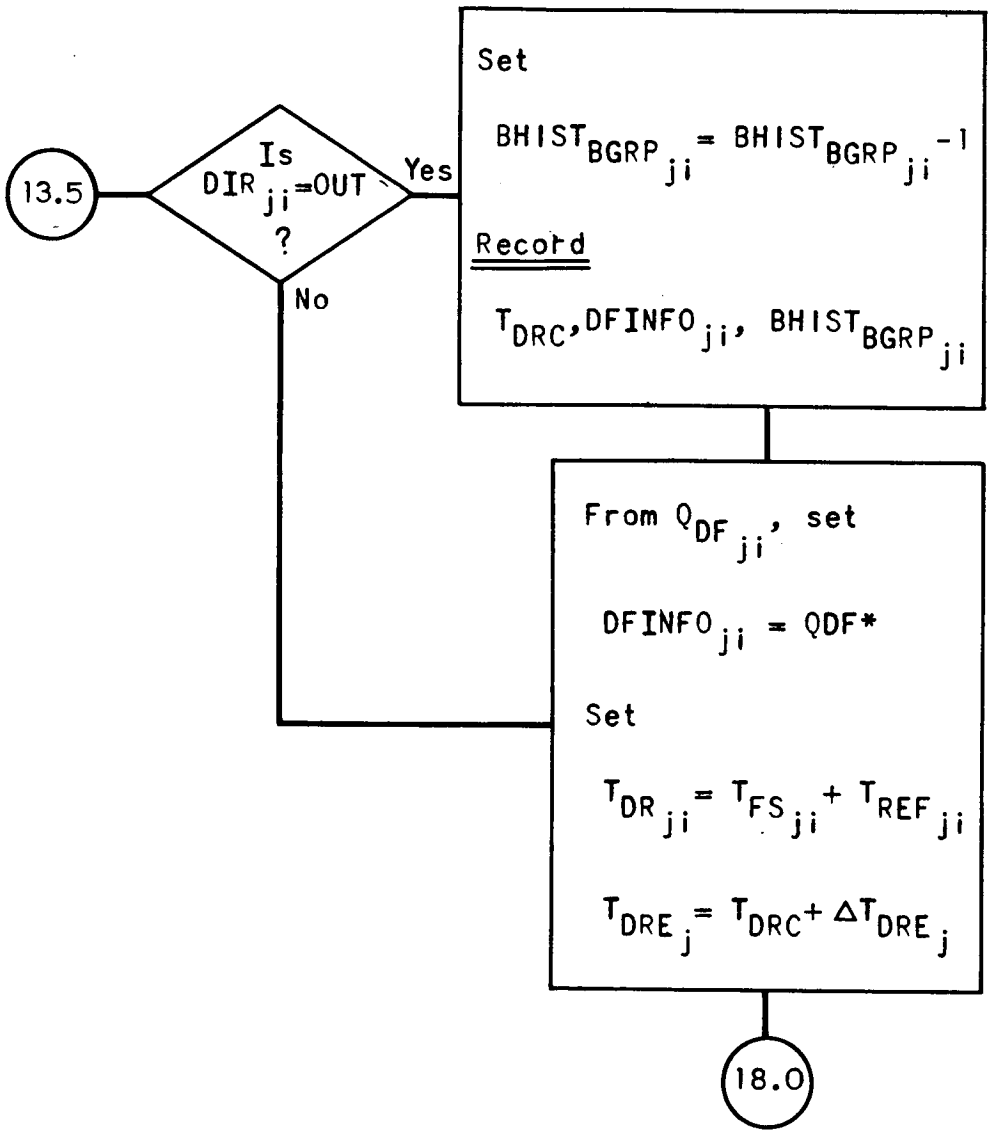


III C Channel: Compute CAT.

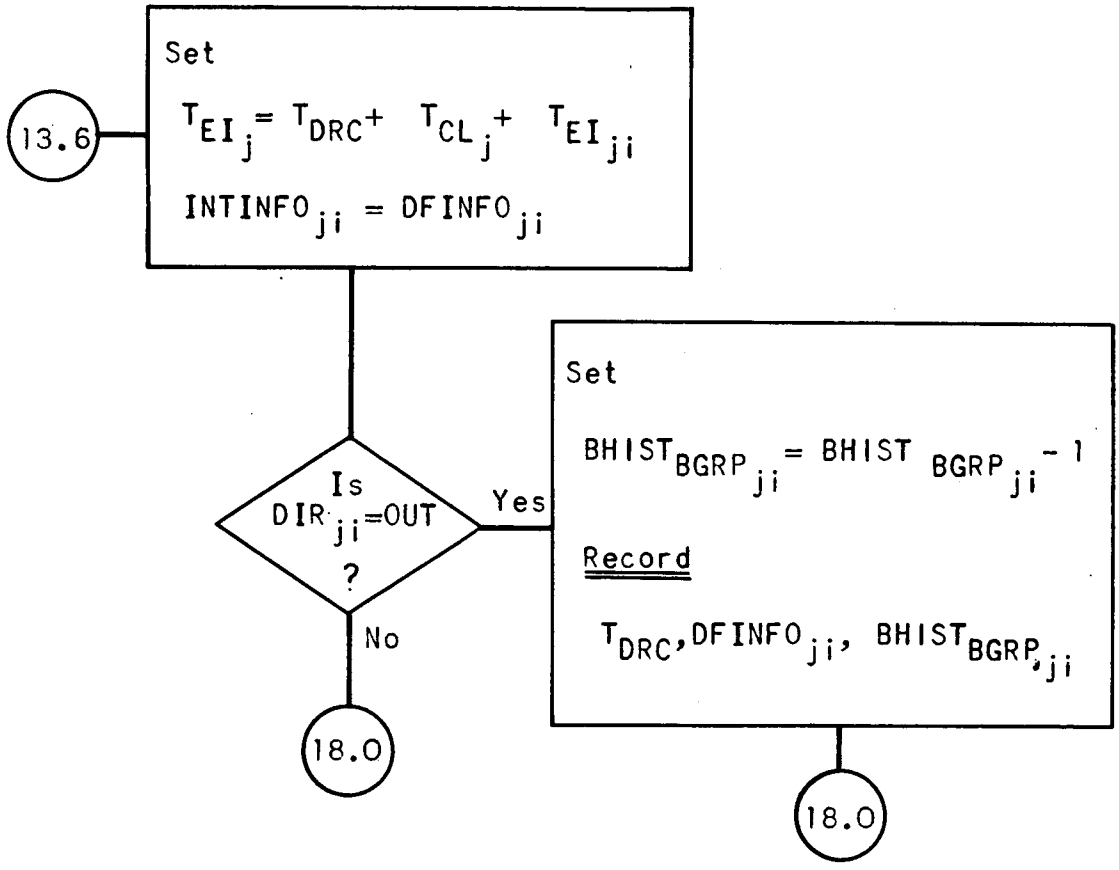
UNIVAC 494 Algorithm
FIGURE A13-4



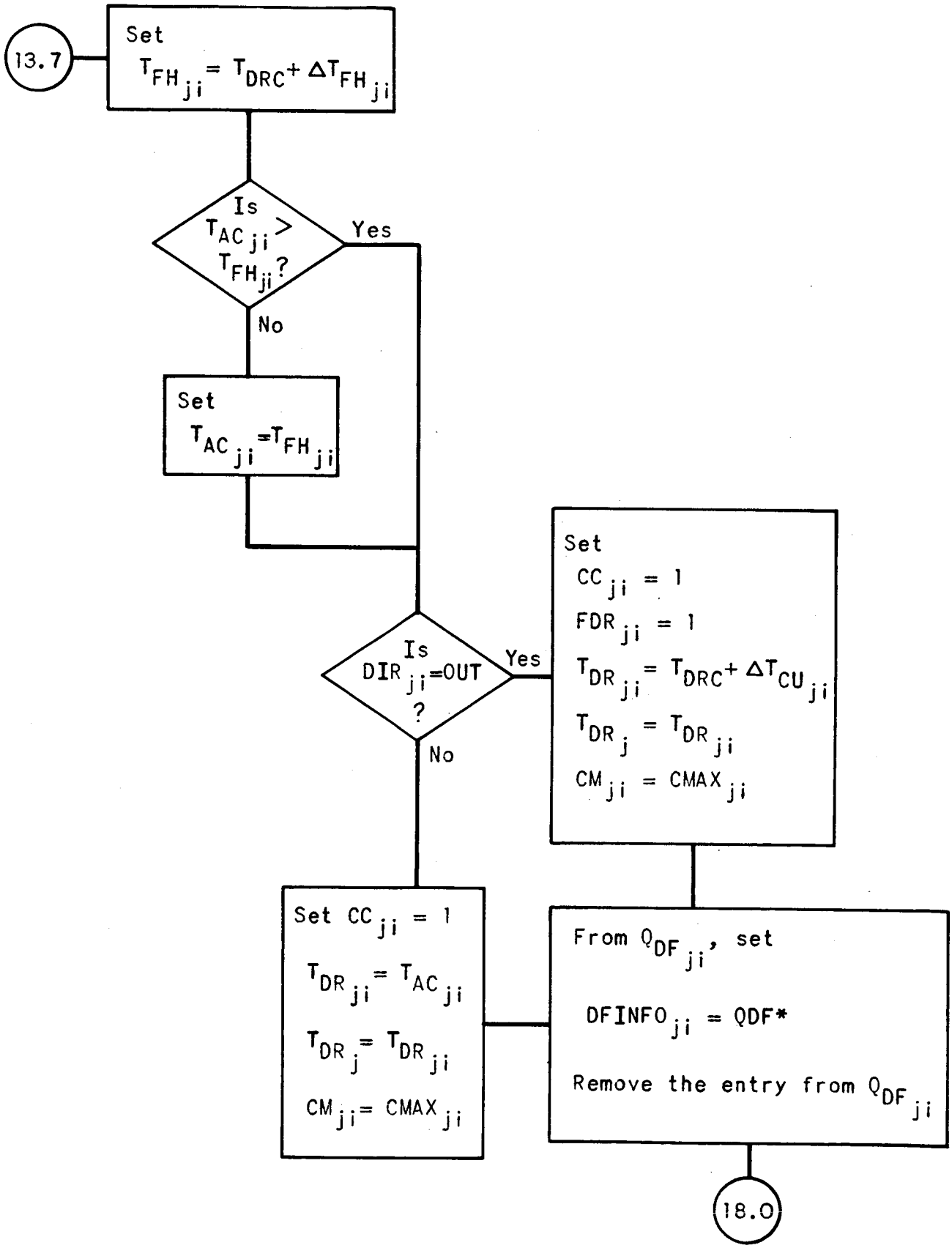
III C Channel: Set a data request.
UNIVAC 494 Algorithm
FIGURE A13-5



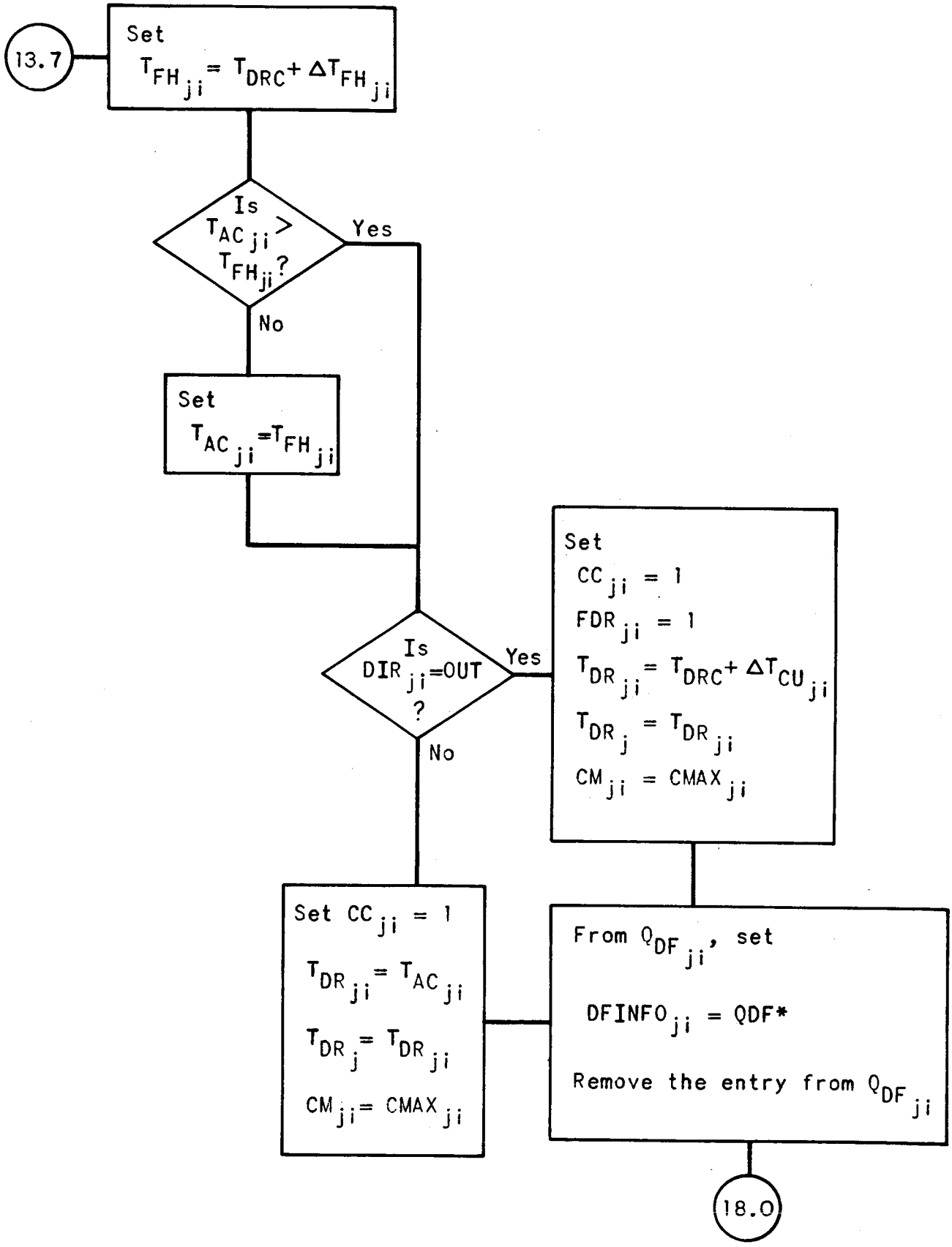
III C Channel: Set the next data request evaluation.
UNIVAC 494 Algorithm
FIGURE A13-6



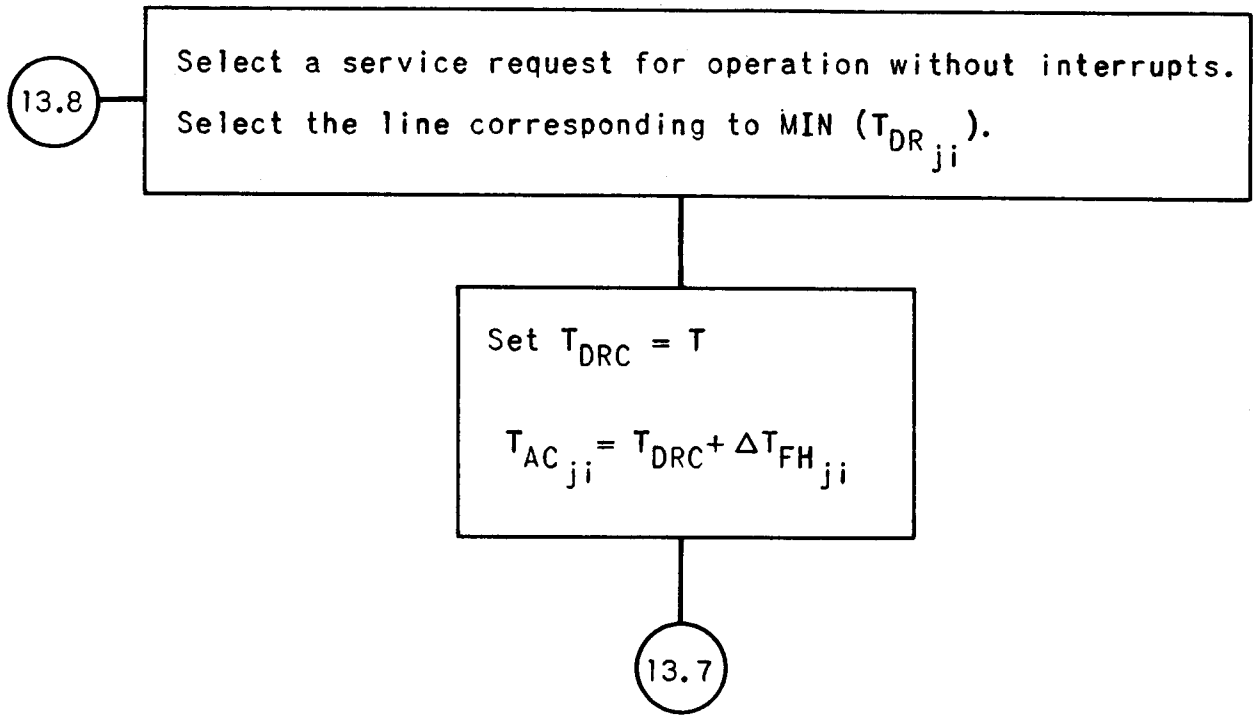
III C Channel: Set an interrupt.
UNIVAC 494 Algorithm
FIGURE A13-7



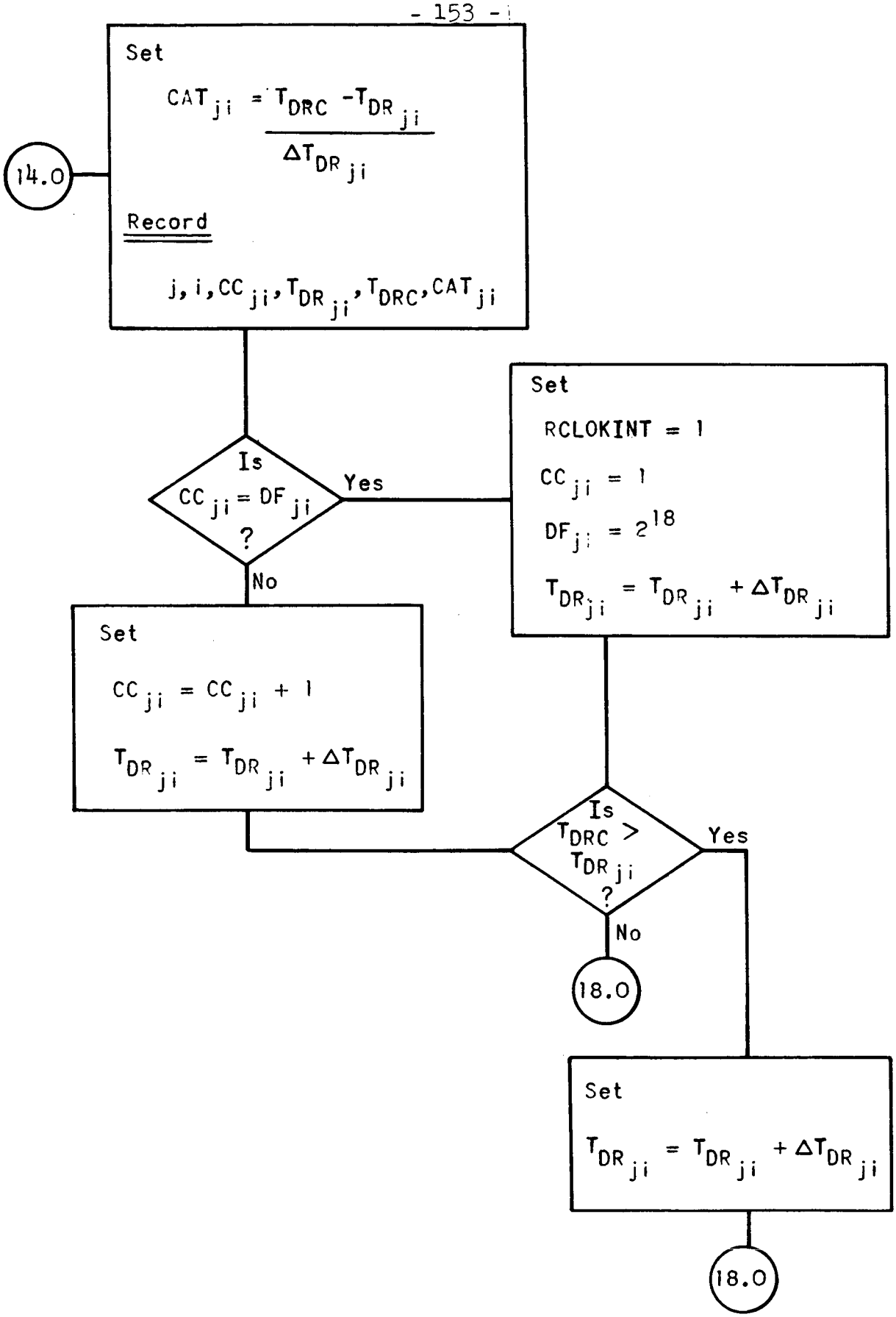
III C Channel: Set the next data frame.
UNIVAC 494 Algorithm
FIGURE A13-8



III C Channel: Set the next data frame.
UNIVAC 494 Algorithm
FIGURE A13-8



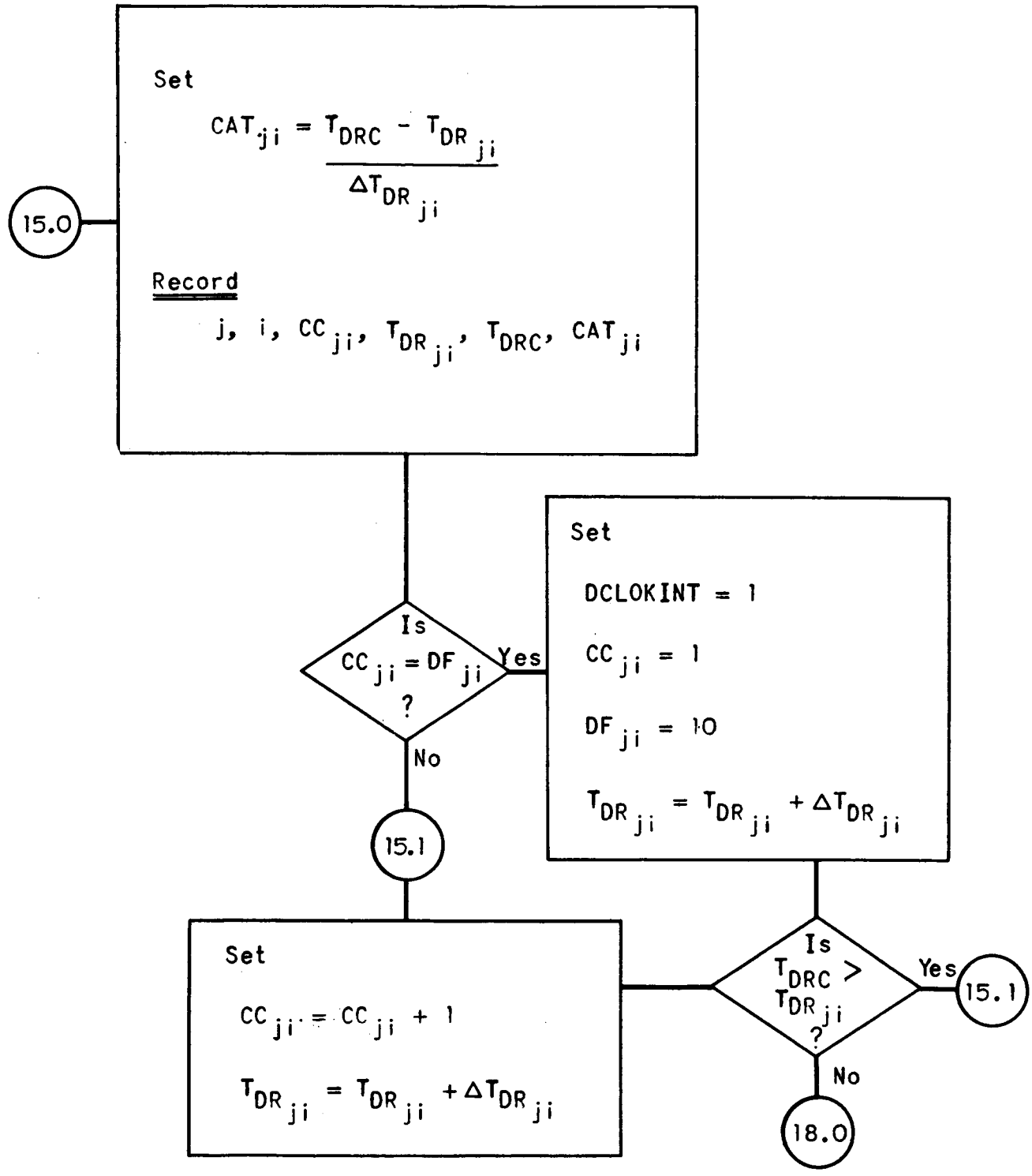
III C Channel: Select a service request.
UNIVAC 494 Algorithm
FIGURE A13-9



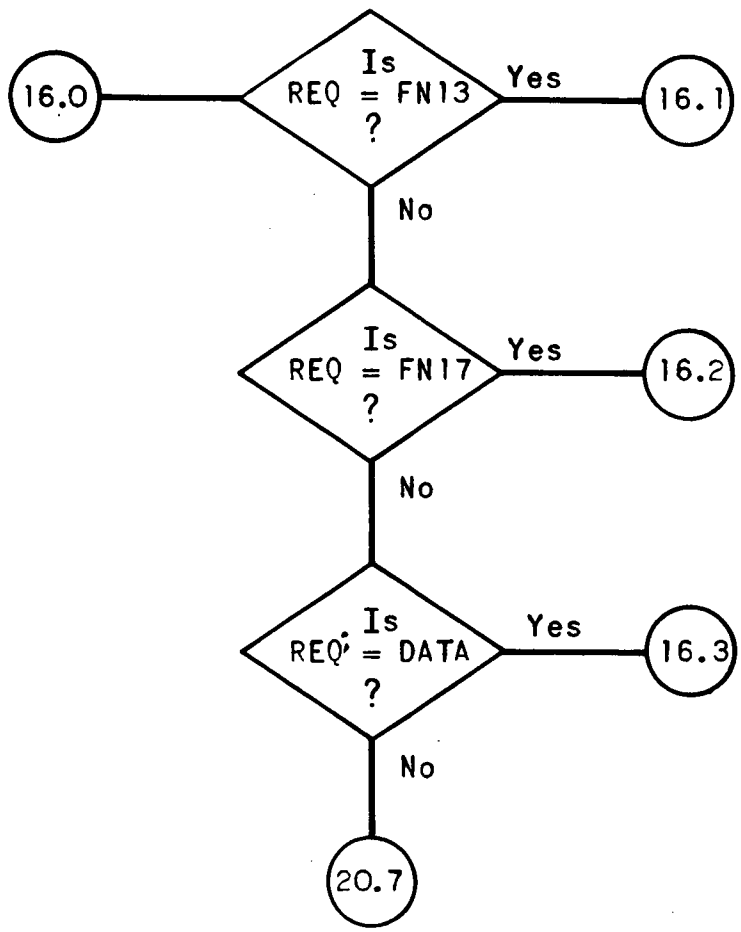
RCLOK: Compute CAT and set the next data request.

UNIVAC 494 Algorithm

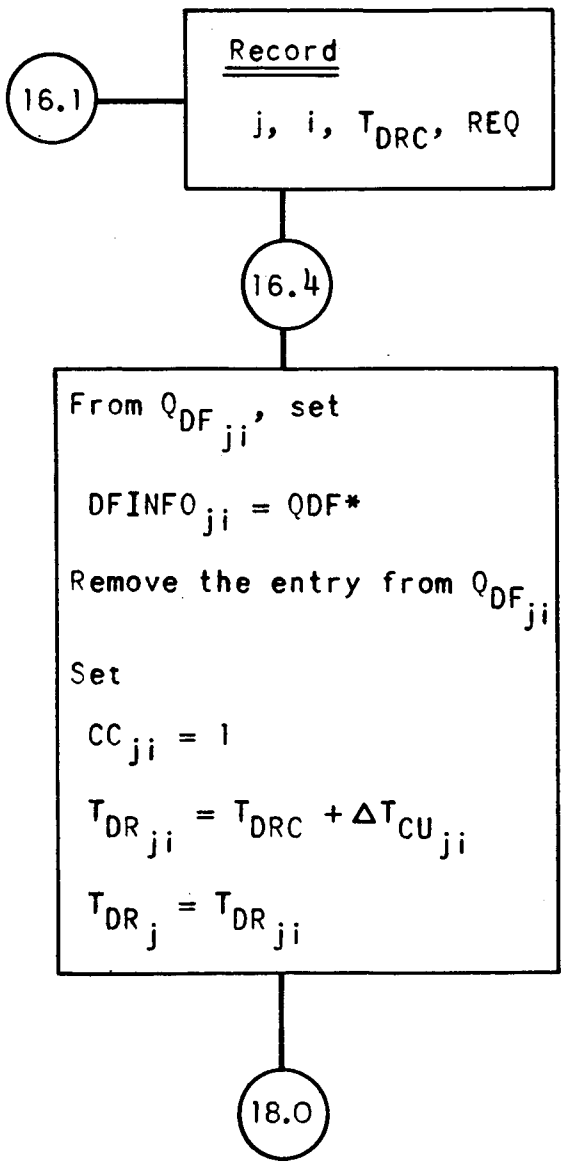
FIGURE A14-1



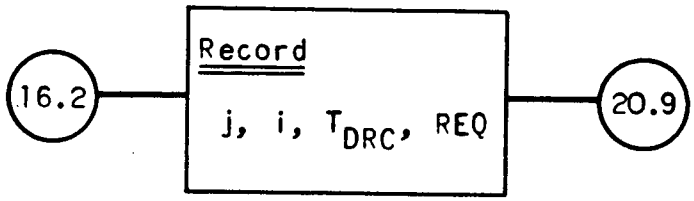
DCLOK: Compute CAT and set the next data request.
UNIVAC 494 Algorithm
FIGURE A15-1



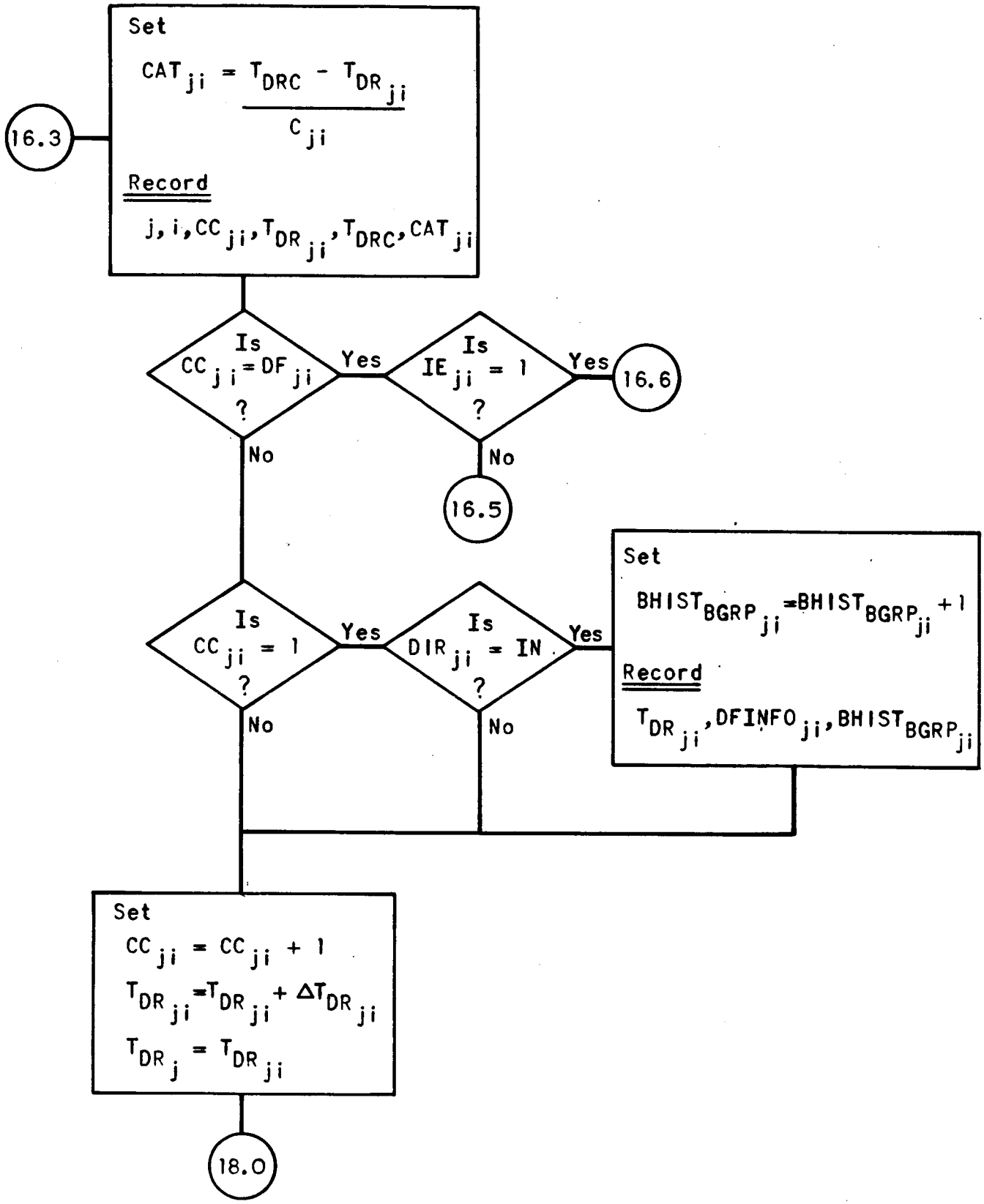
1004 Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A16-1



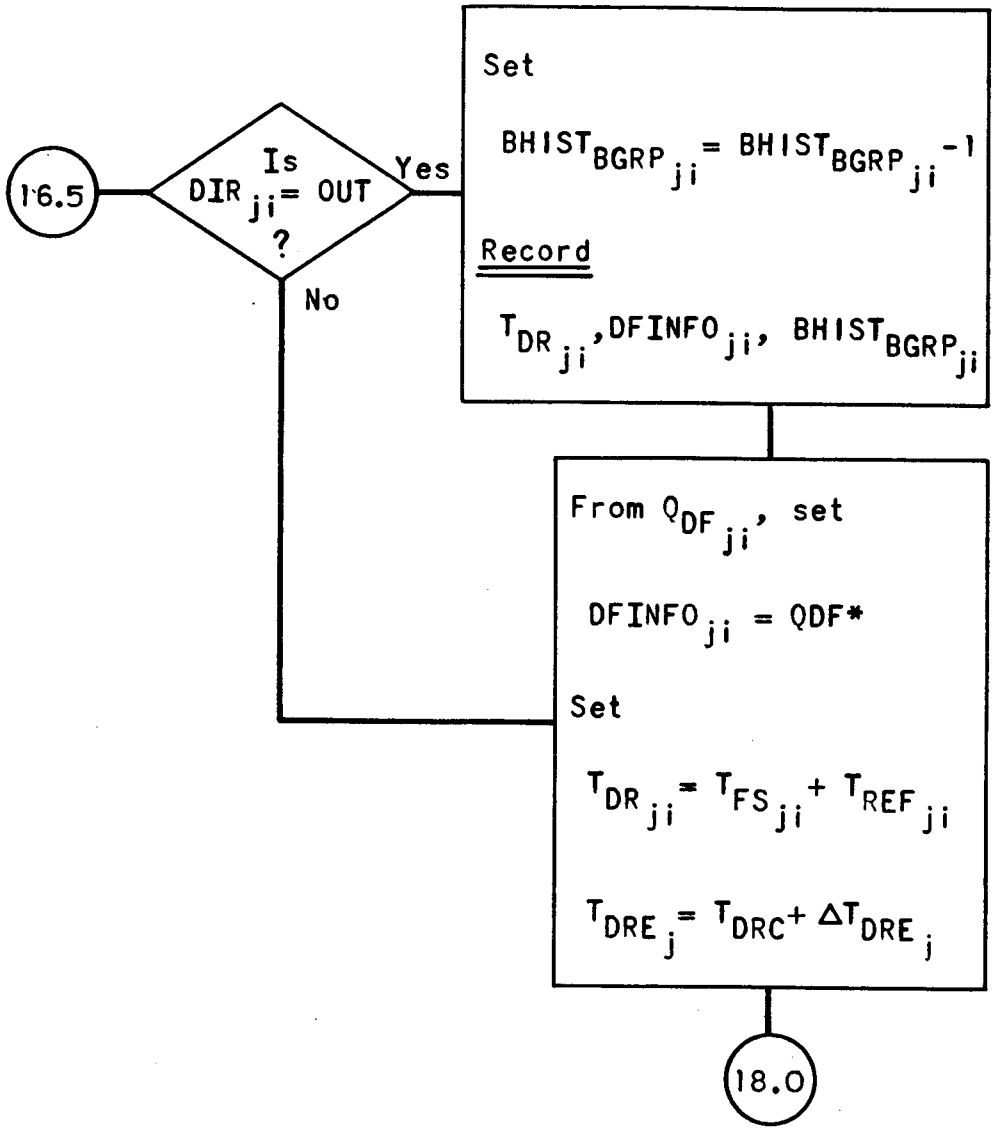
1004 Channel: Perform a FN13 action.
UNIVAC 494 Algorithm
FIGURE A16-2



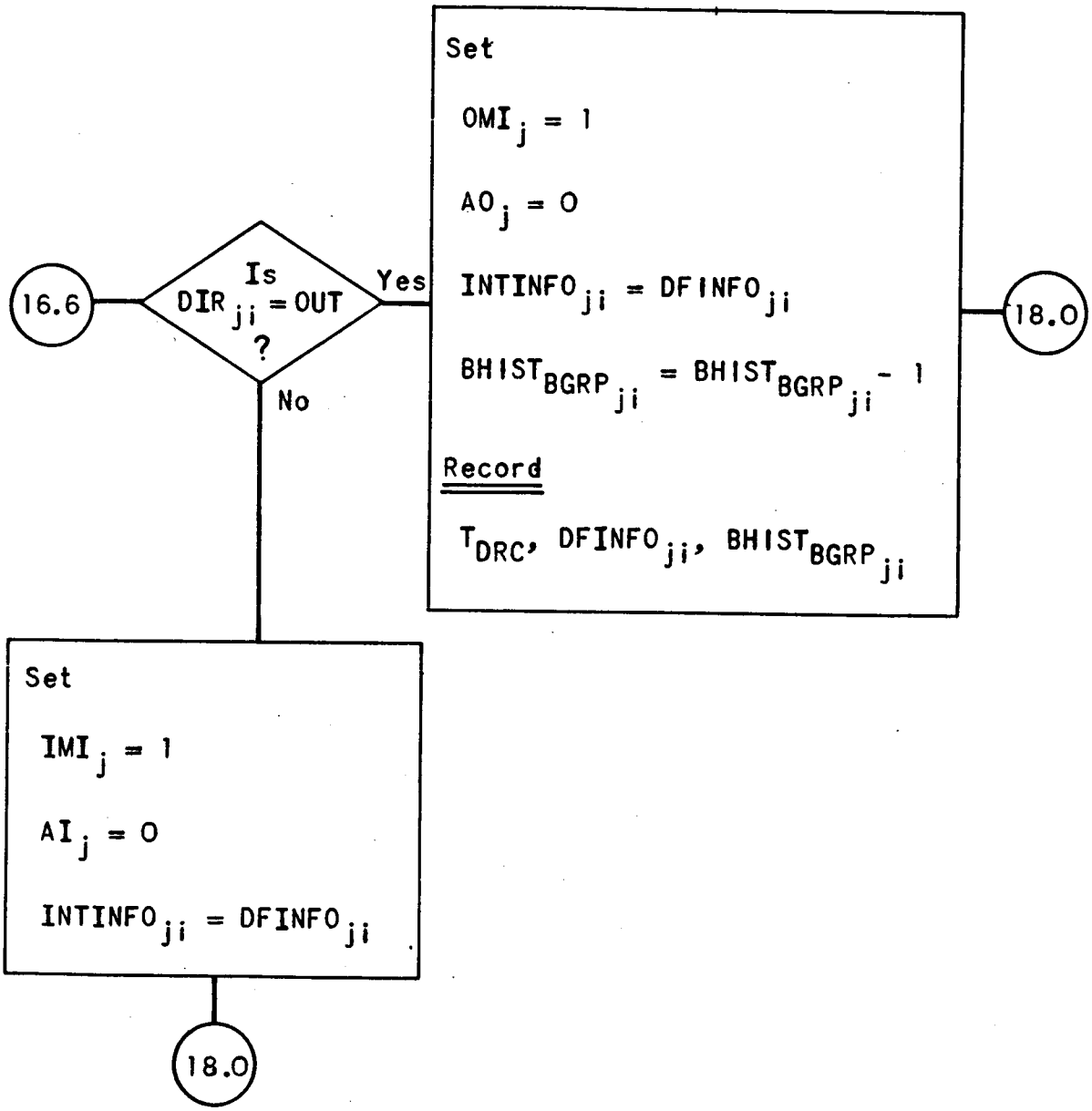
1004 Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A16-3



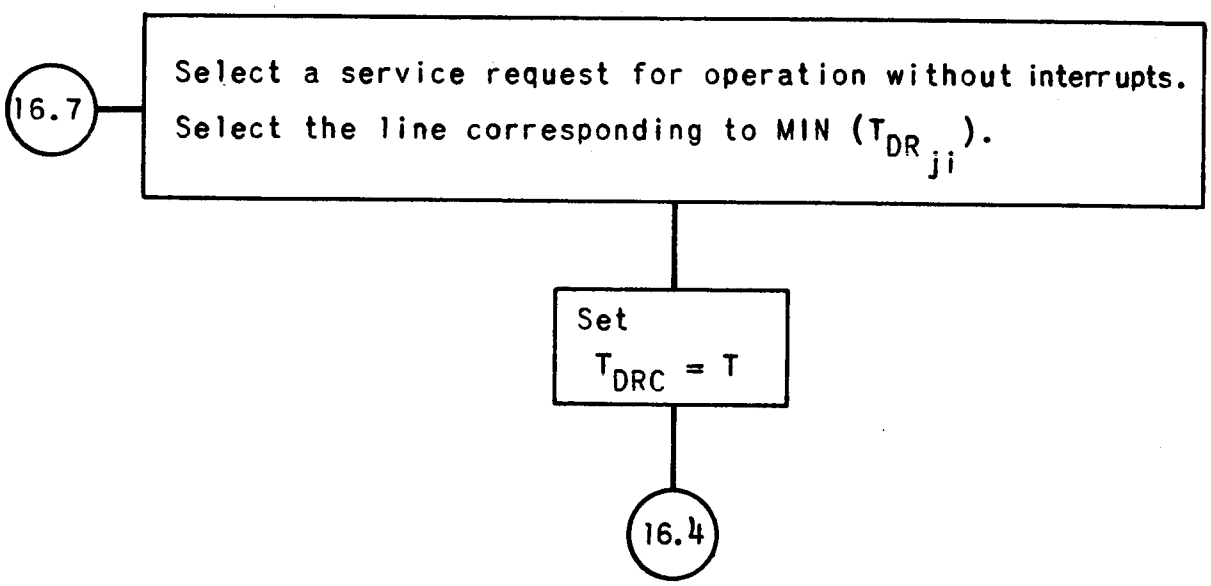
1004 Channel: Compute CAT and set the next data request.
UNIVAC 494 Algorithm
FIGURE A16-4



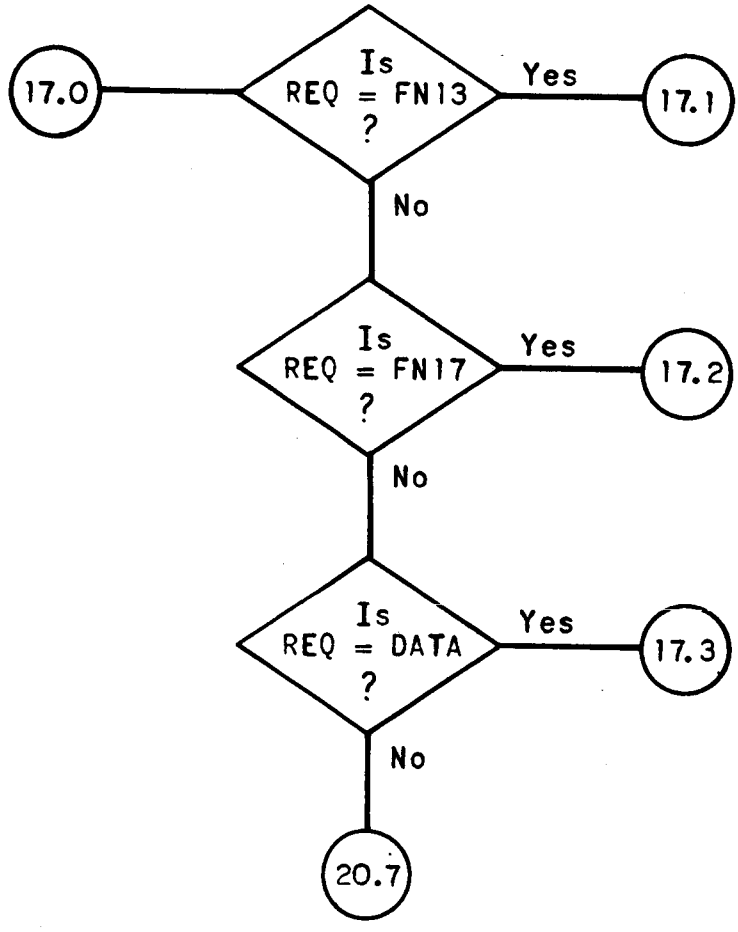
1004 Channel: Set the next data request evaluation.
UNIVAC 494 Algorithm
FIGURE A16-5



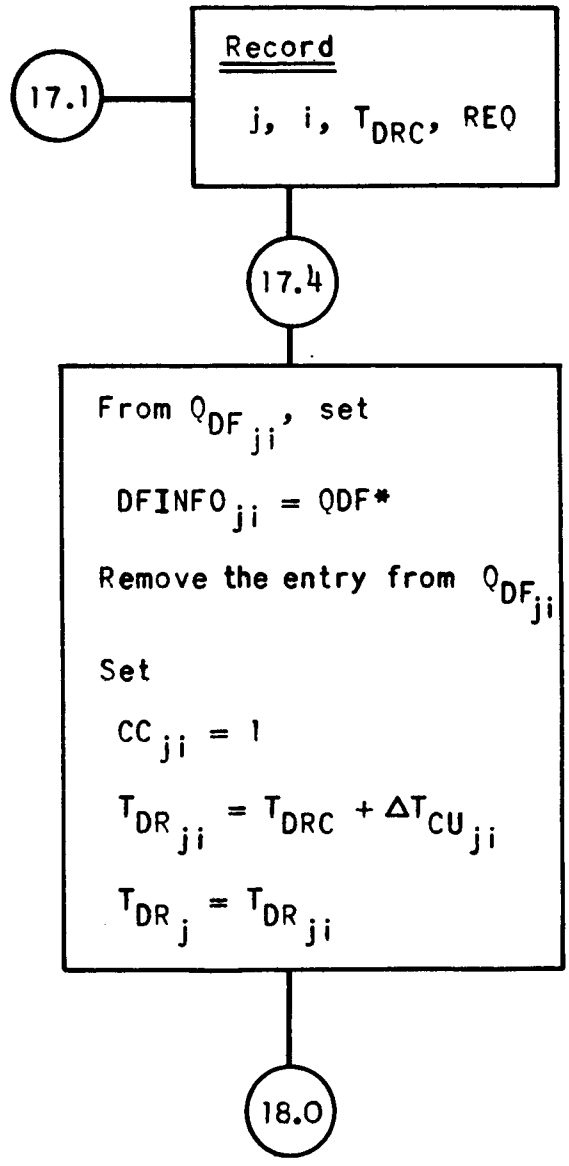
1004 Channel: Set an interrupt.
UNIVAC 494 Algorithm
FIGURE A16-6



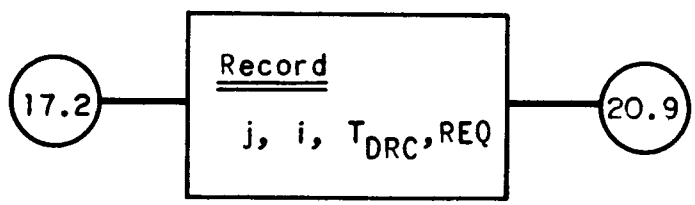
1004 Channel: Select a service request.
UNIVAC 494 Algorithm
FIGURE A16-7



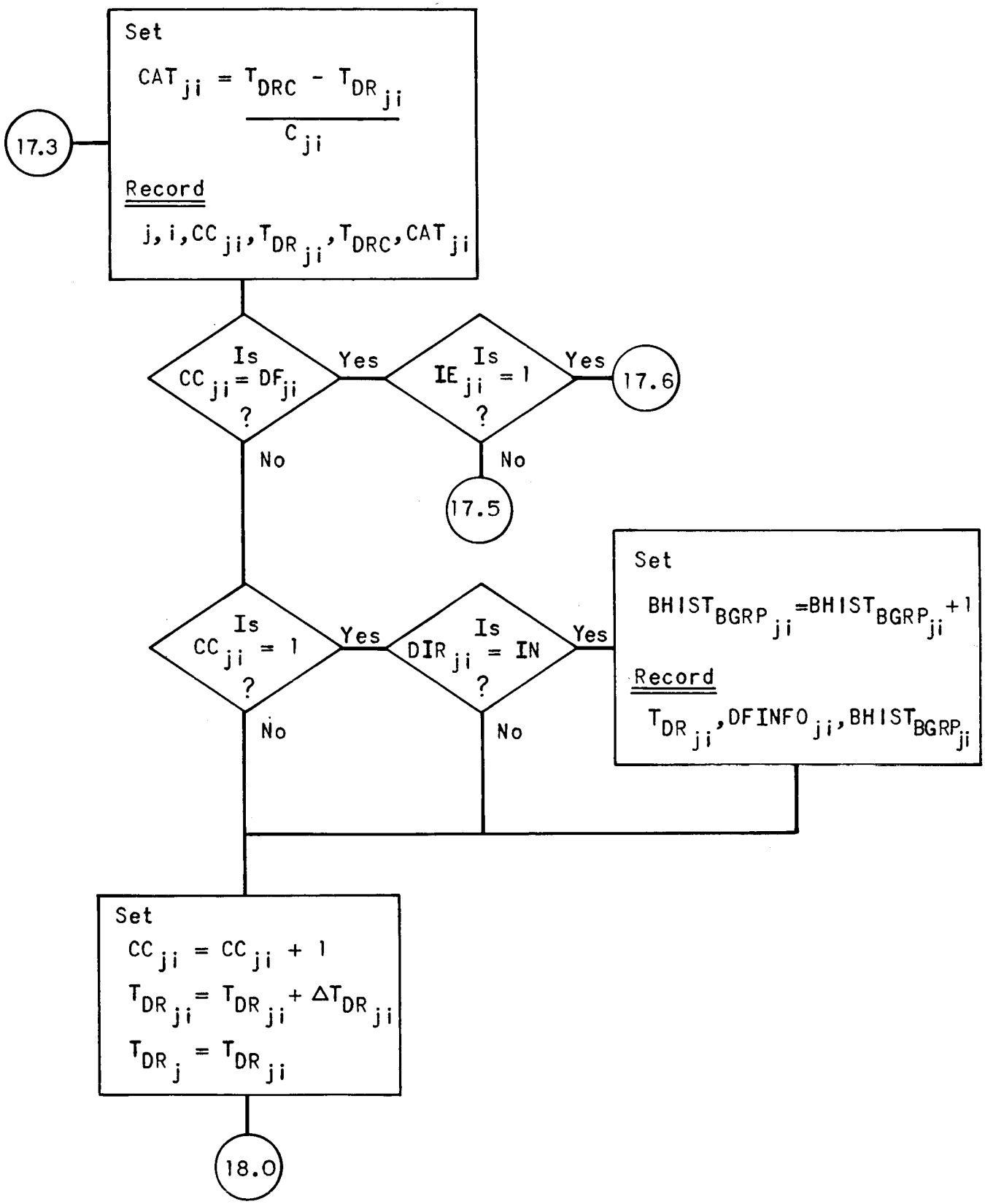
CONSOLE Channel: Determine the I/O action requested.
UNIVAC 494 Algorithm
FIGURE A17-1



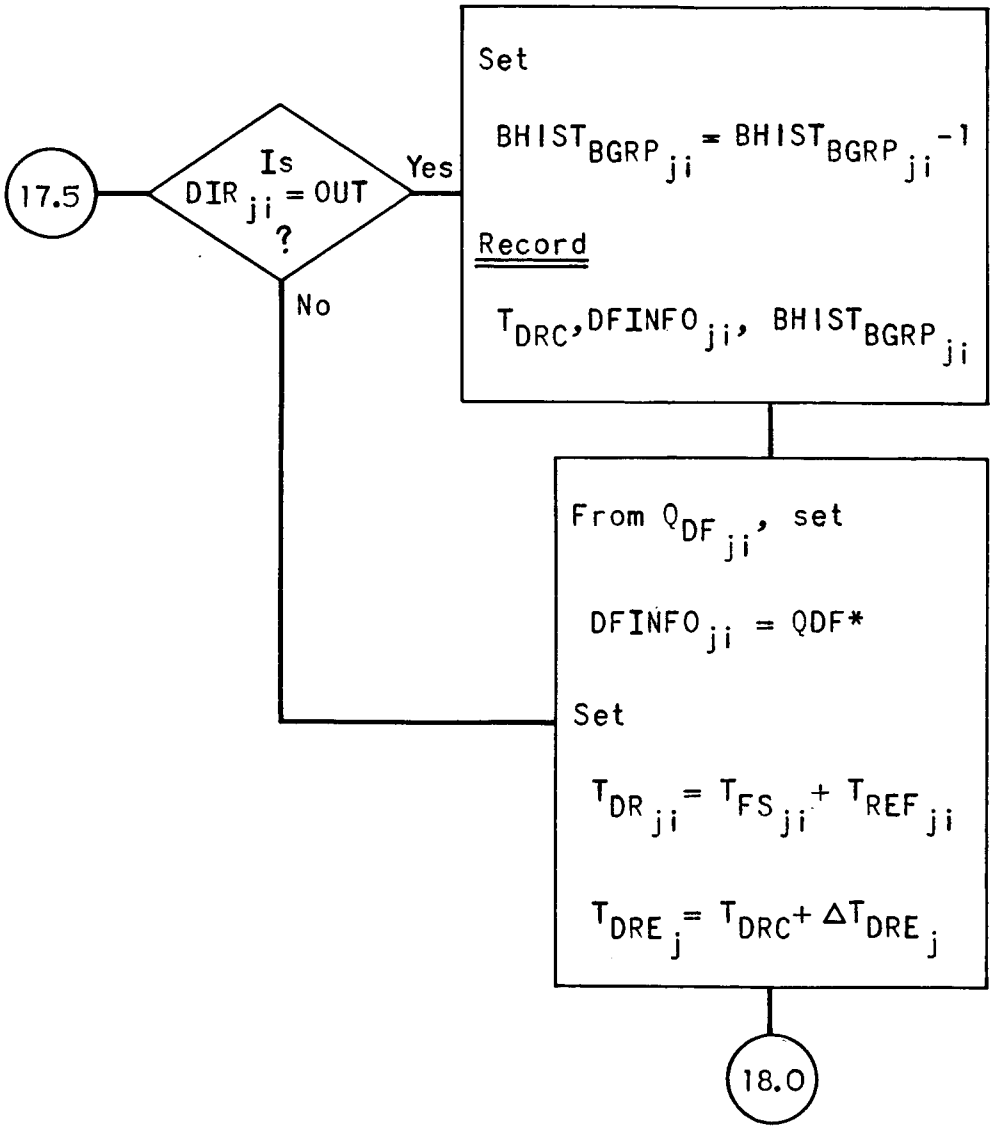
CONSOLE Channel: Perform a FN13 action.
UNIVAC 494 Algorithm
FIGURE A17-2



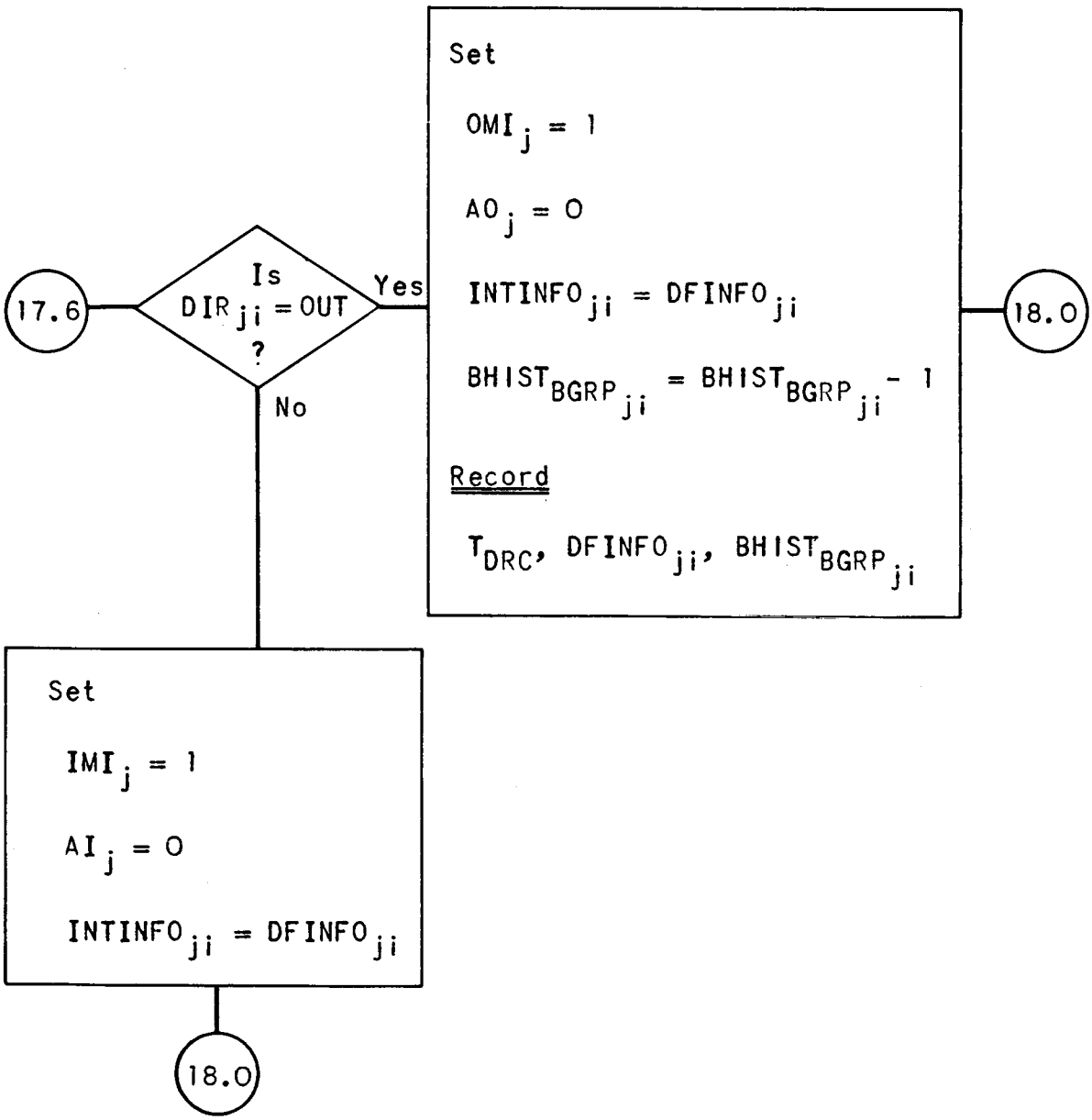
CONSOLE Channel: Perform a FN17 action.
UNIVAC 494 Algorithm
FIGURE A17-3



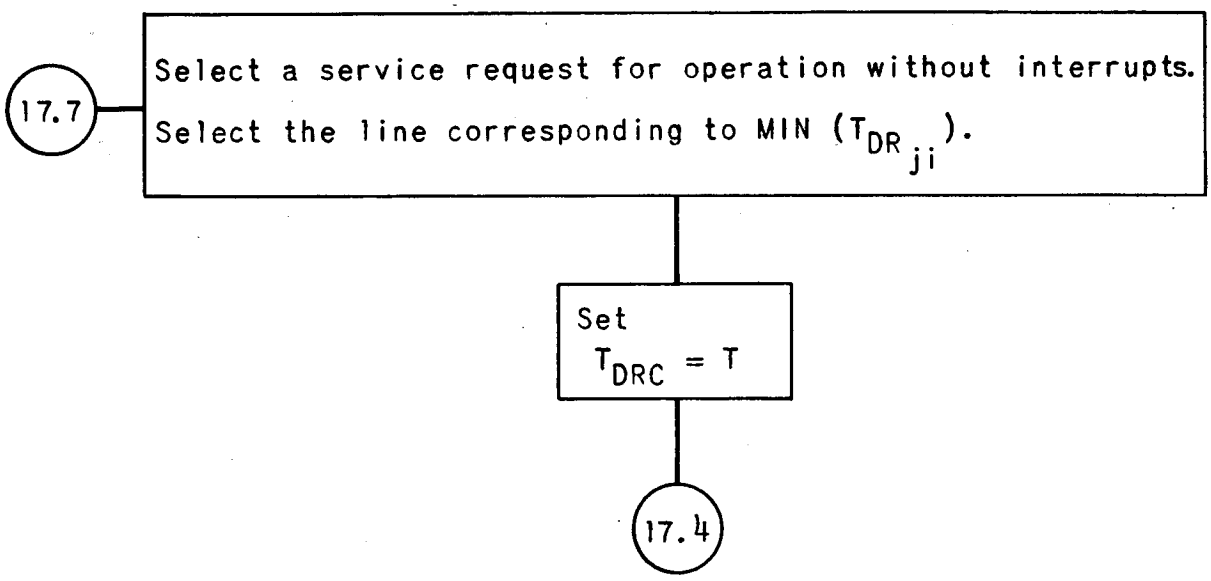
CONSOLE Channel: Compute CAT and set the next data request.
UNIVAC 494 Algorithm
FIGURE A17-4



CONSOLE Channel: Set the next data request evaluation.
UNIVAC 494 Algorithm
FIGURE A17-5



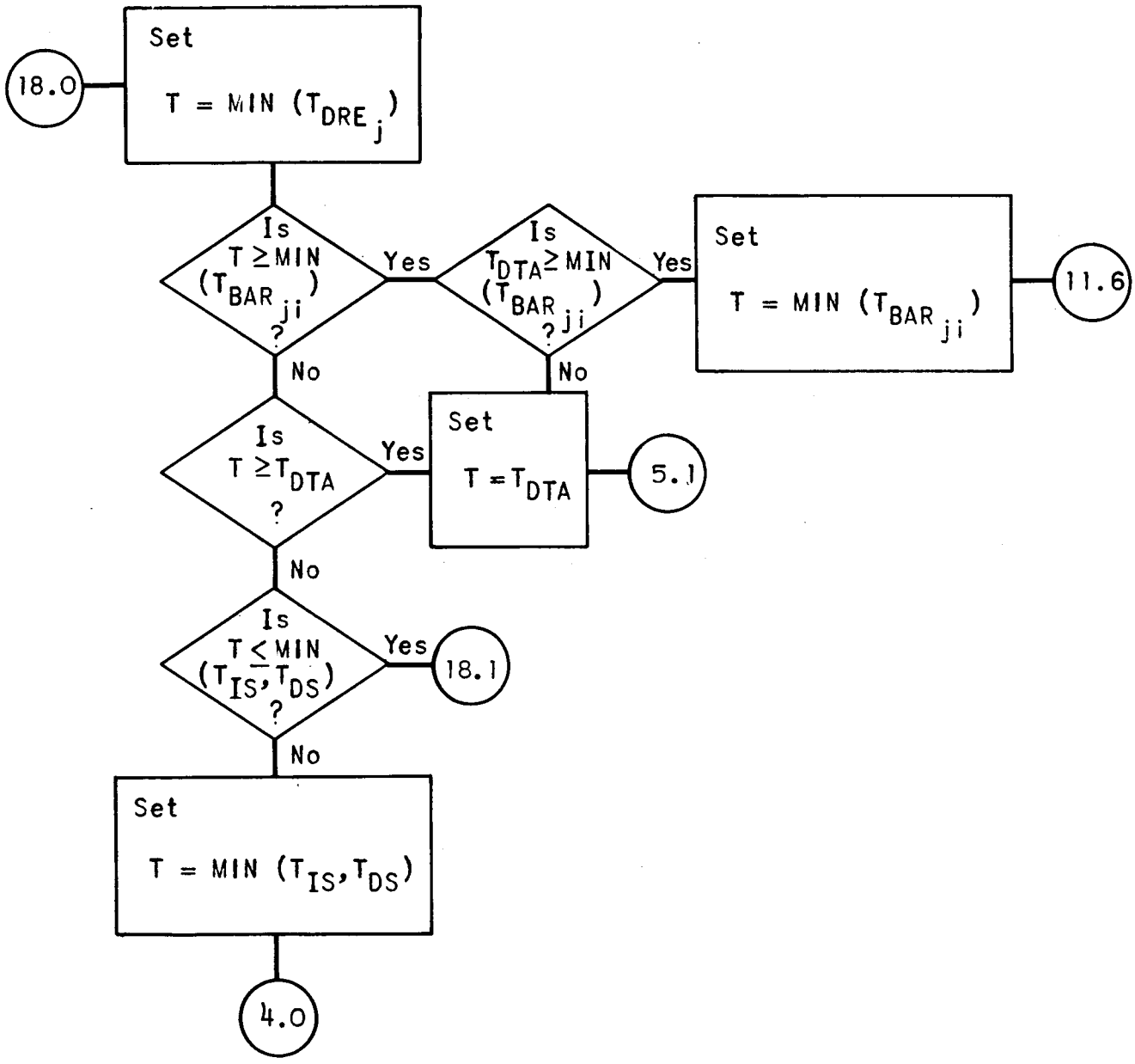
CONSOLE Channel: Set an interrupt.
UNIVAC 494 Algorithm
FIGURE A17-6



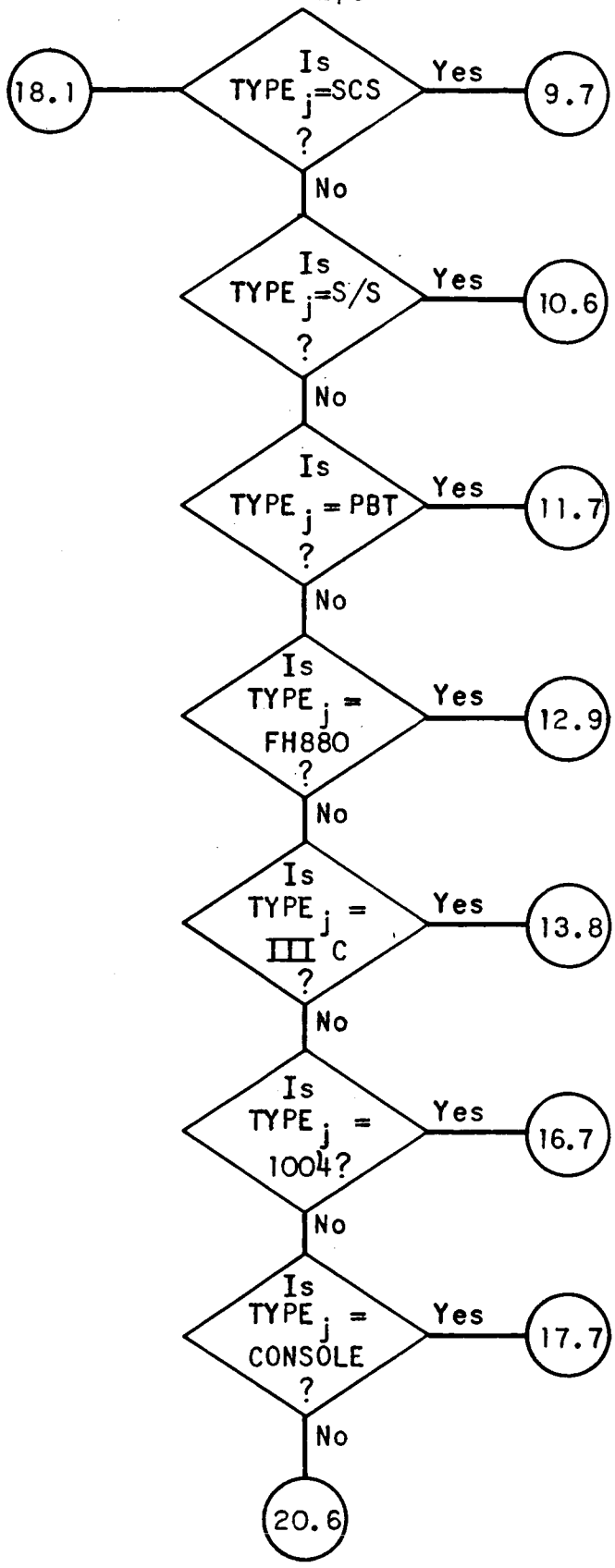
CONSOLE Channel: Select a service request.

UNIVAC 494 Algorithm

FIGURE A17-7



Time Housekeeping: Find the next running time event.
UNIVAC 494 Algorithm
FIGURE A18-1



Time Housekeeping: Select the correct I/O model.
UNIVAC 494 Algorithm
FIGURE A18-2

19.0

For every data transfer, we have recorded:
- Channel identification,
- Device identification,
- Number of the data request,
- Initiation time of the data request,
- Completion time of the data request,
- CAT of the data request,
- Delay of the data request stream (S/S only).

For every FN13 or FN17, we have recorded:
- Channel identification,
- Device identification,
- Completion time of the data request.

For every interrupt, we have recorded:
- Channel identification,
- Device identification,
- Type of the interrupt,
- Interrupt scan detection time of the interrupt

For every subinstruction of special interest, we ha
- Type of the subinstruction,
- Execution time of the subinstruction,
- Identification of the program segment containi subinstruction,
- Identification of the data frame for which the tion is executed,
- Other variables pertinent to each particular s

For every input data frame, we have recorded:

Identification of the data frame,

Initiation time of the data frame,

The number of buffers in use for the buffer group used by the data frame.

For every output data frame, we have recorded:

Identification of the data frame,

Completion time of the data frame,

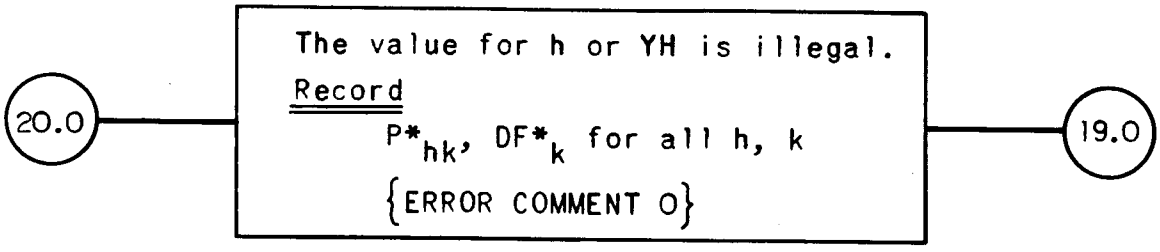
The number of buffers in use for the buffer group used by the data frame.

If the first request in a data frame or the entire data frame is lost by excessive interrupt service duration, the interaction effect is specially denoted.

From the information which has been recorded, abstract the desired information and generate a resource utilization report for transfer, processing, and storage.

STOP

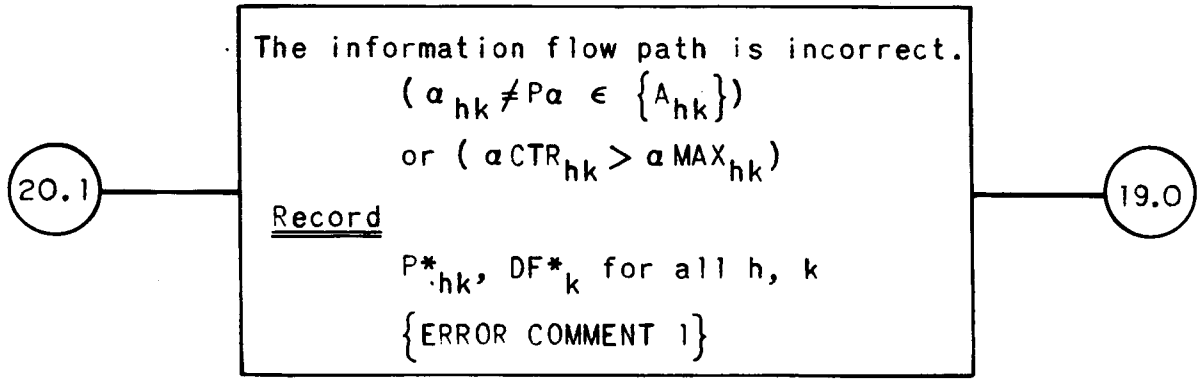
Results Computation.
UNIVAC 494 Algorithm
FIGURE A19-1



Error Housekeeping: Stop after h error.

UNIVAC 494 Algorithm

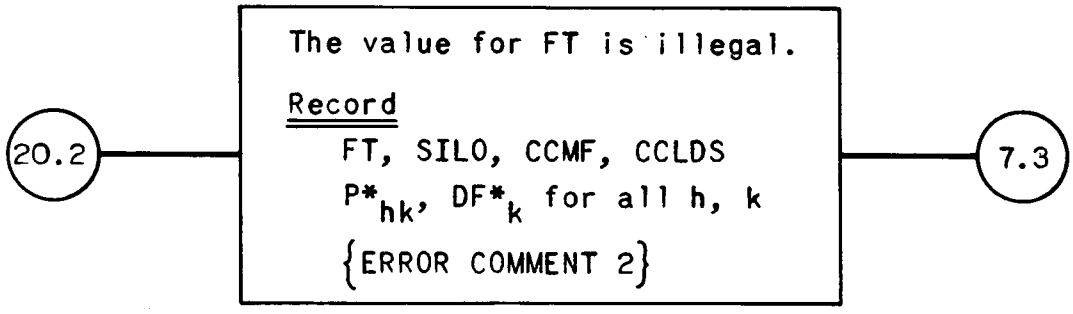
FIGURE A20-1



Error Housekeeping: Stop after information flow path error.

UNIVAC 494 Algorithm

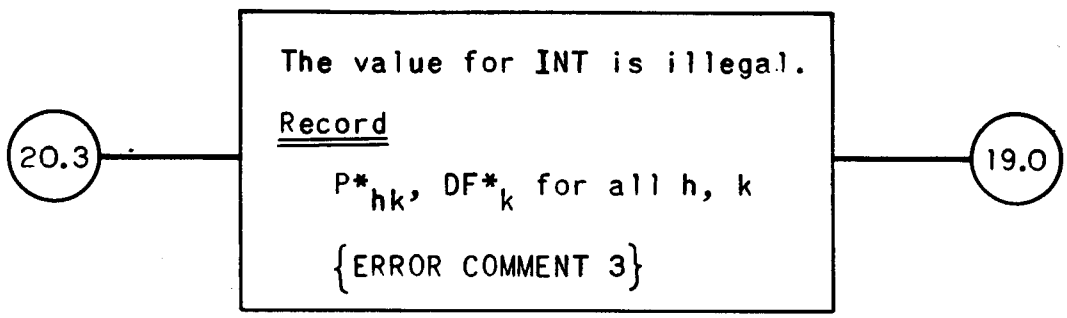
FIGURE A20-2



Error Housekeeping: Attempt to continue after FT error.

UNIVAC 494 Algorithm

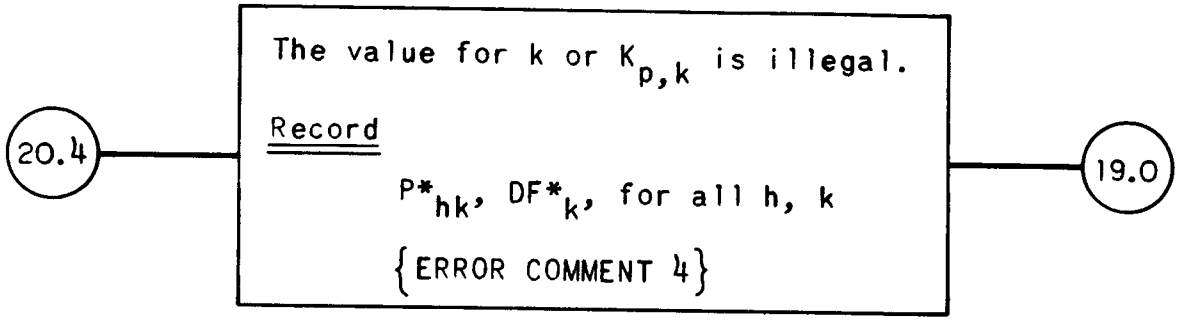
FIGURE A20-3



Error Housekeeping: Stop after INT error.

UNIVAC 494 Algorithm

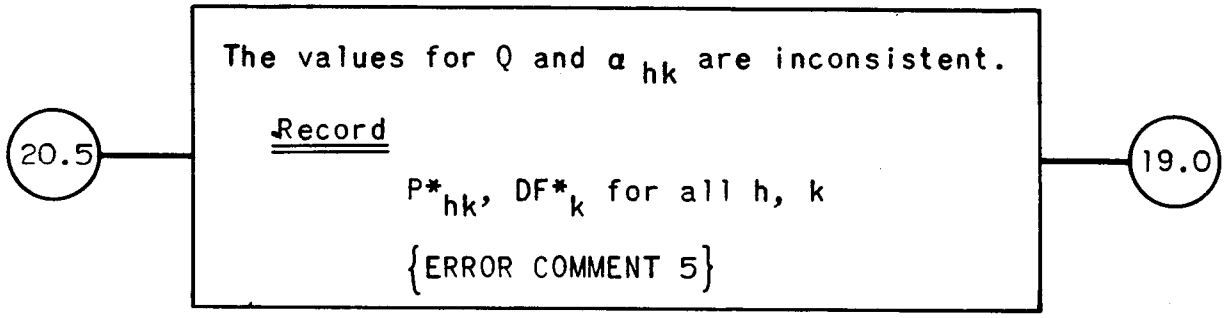
FIGURE A20-4



Error Housekeeping: Stop after k error.

UNIVAC 494 Algorithm

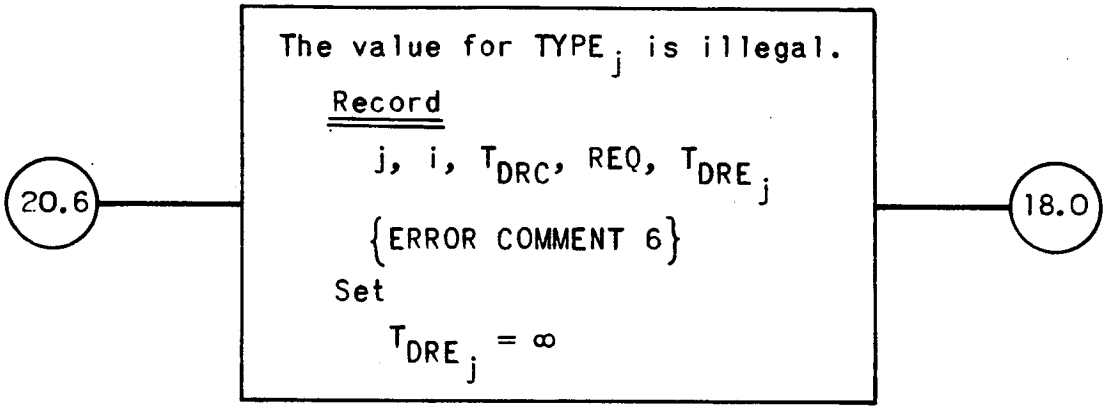
FIGURE A20-5



Error Housekeeping: Stop after Q error.

UNIVAC 494 Algorithm

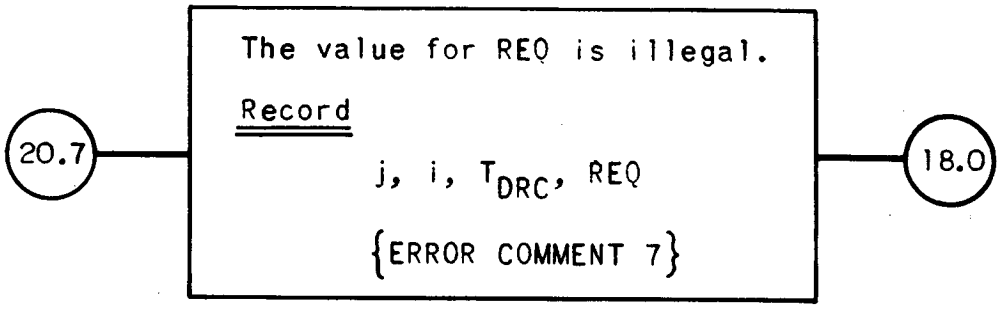
FIGURE A20-6



Error Housekeeping: Attempt to continue after TYPE$_j$ error.

UNIVAC 494 Algorithm

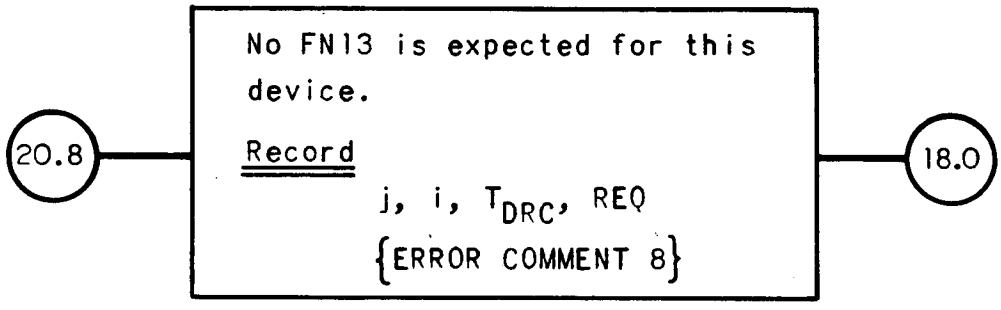
FIGURE A20-7



Error Housekeeping: Attempt to continue after REQ error.

UNIVAC 494 Algorithm

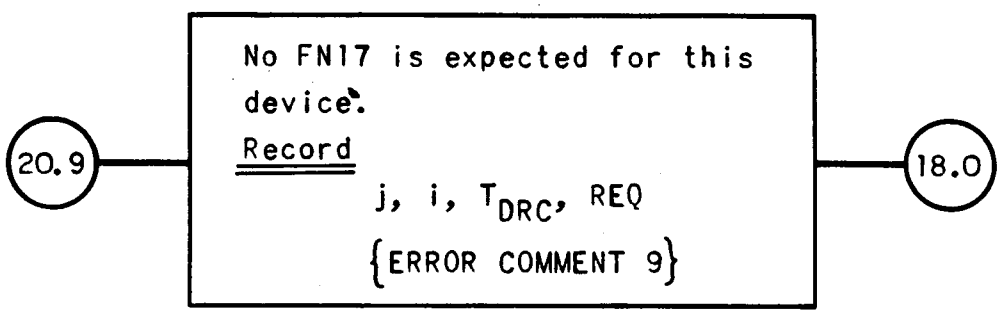
FIGURE A20-8



Error Housekeeping: Attempt to continue after FN13 error.

UNIVAC 494 Algorithm

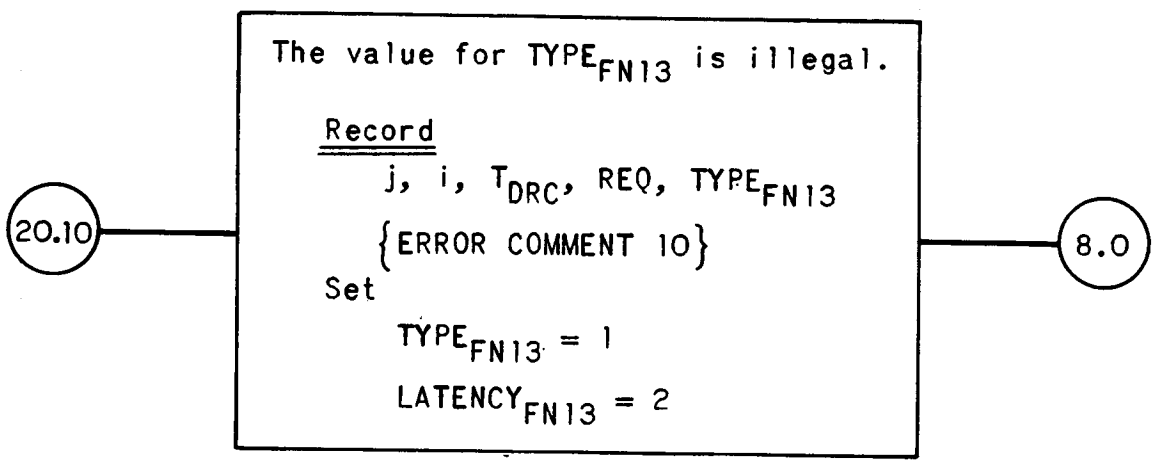
FIGURE A20-9



Error Housekeeping: Attempt to continue after FN17 error.

UNIVAC 494 Algorithm

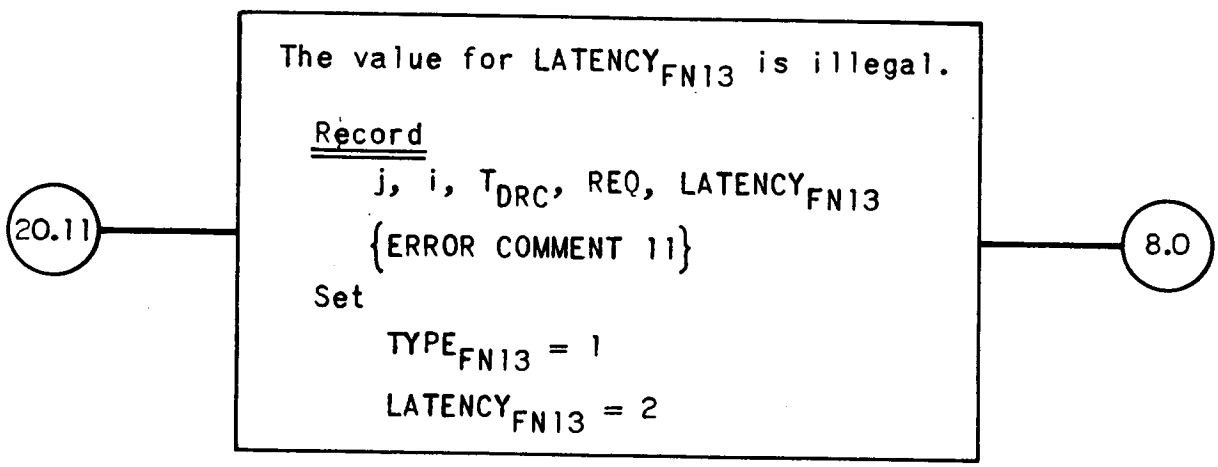
FIGURE A20-10



Error Housekeeping: Attempt to continue after $TYPE_{FN13}$ error.

UNIVAC 494 Algorithm

FIGURE A20-11



Error Housekeeping: Attempt to continue after LATENCY$_{FN13}$ error.

UNIVAC 494 Algorithm

FIGURE A20-12