

[54] **AUTOMATIC SWITCHING CONTROL SYSTEM FOR RAILWAY CLASSIFICATION YARDS**

[75] Inventors: **John J. DePaola, Penfield; Charles W. Morse, Rochester, both of N.Y.**

[73] Assignee: **General Signal Corporation, Rochester, N.Y.**

[22] Filed: **Apr. 4, 1973**

[21] Appl. No.: **348,376**

[52] U.S. Cl. .... **104/26 B, 246/1 R**  
 [51] Int. Cl. .... **B611 7/08**  
 [58] Field of Search ..... **246/182 A, 1 R; 104/26 R, 104/26 B**

[56] **References Cited**  
**UNITED STATES PATENTS**

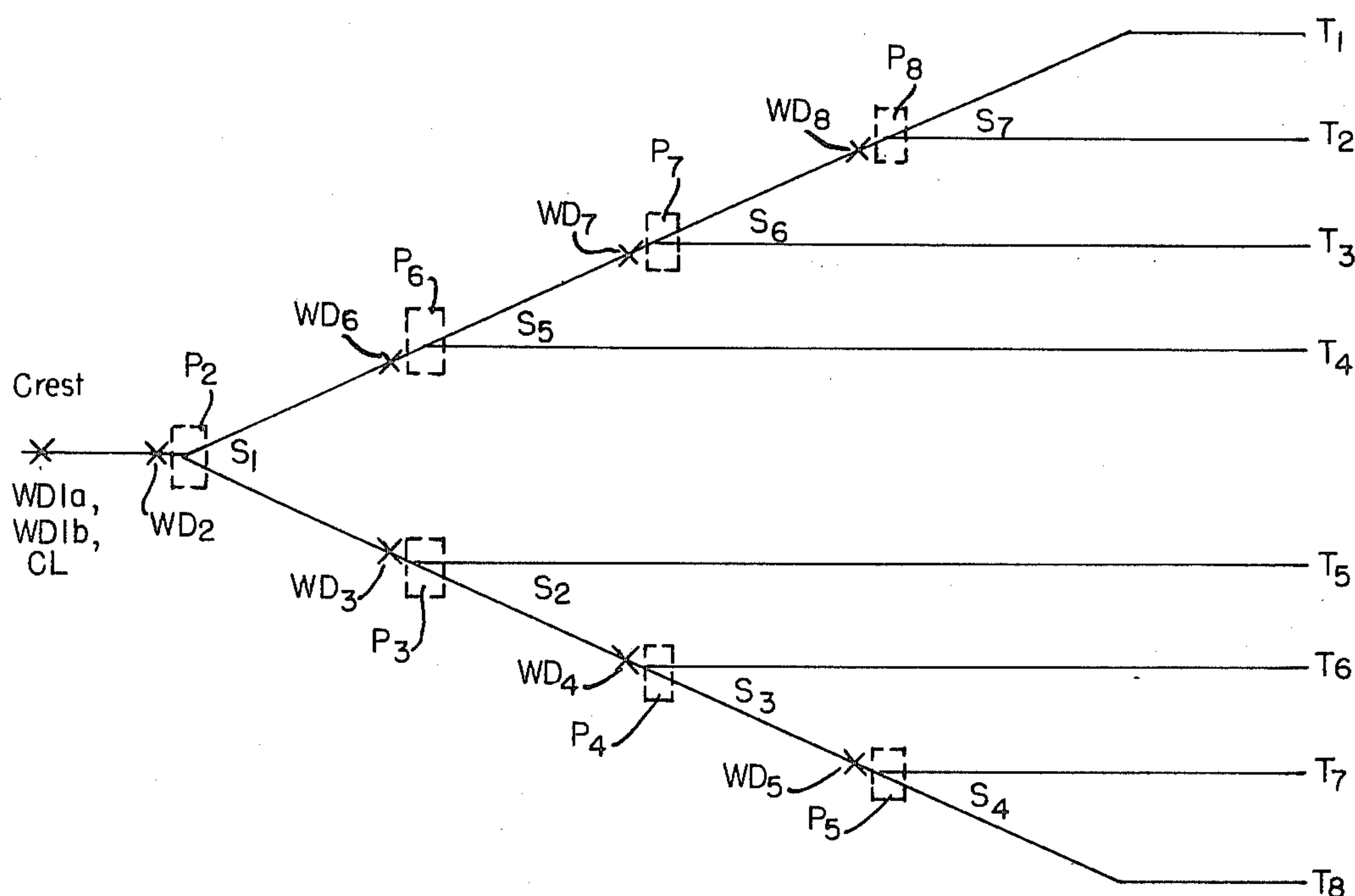
3,307,031	2/1967	Frielinghaus et al. ....	246/4
3,480,773	11/1969	McCune et al. ....	246/4
3,543,020	11/1970	McGlumphy et al. ....	246/1 R
3,736,420	5/1973	Elder et al. ....	246/182 A

*Primary Examiner*—M. Henson Wood, Jr.  
*Assistant Examiner*—Reinhard J. Eisenzopf  
*Attorney, Agent, or Firm*—Stanley B. Green; Milton E. Kleinman

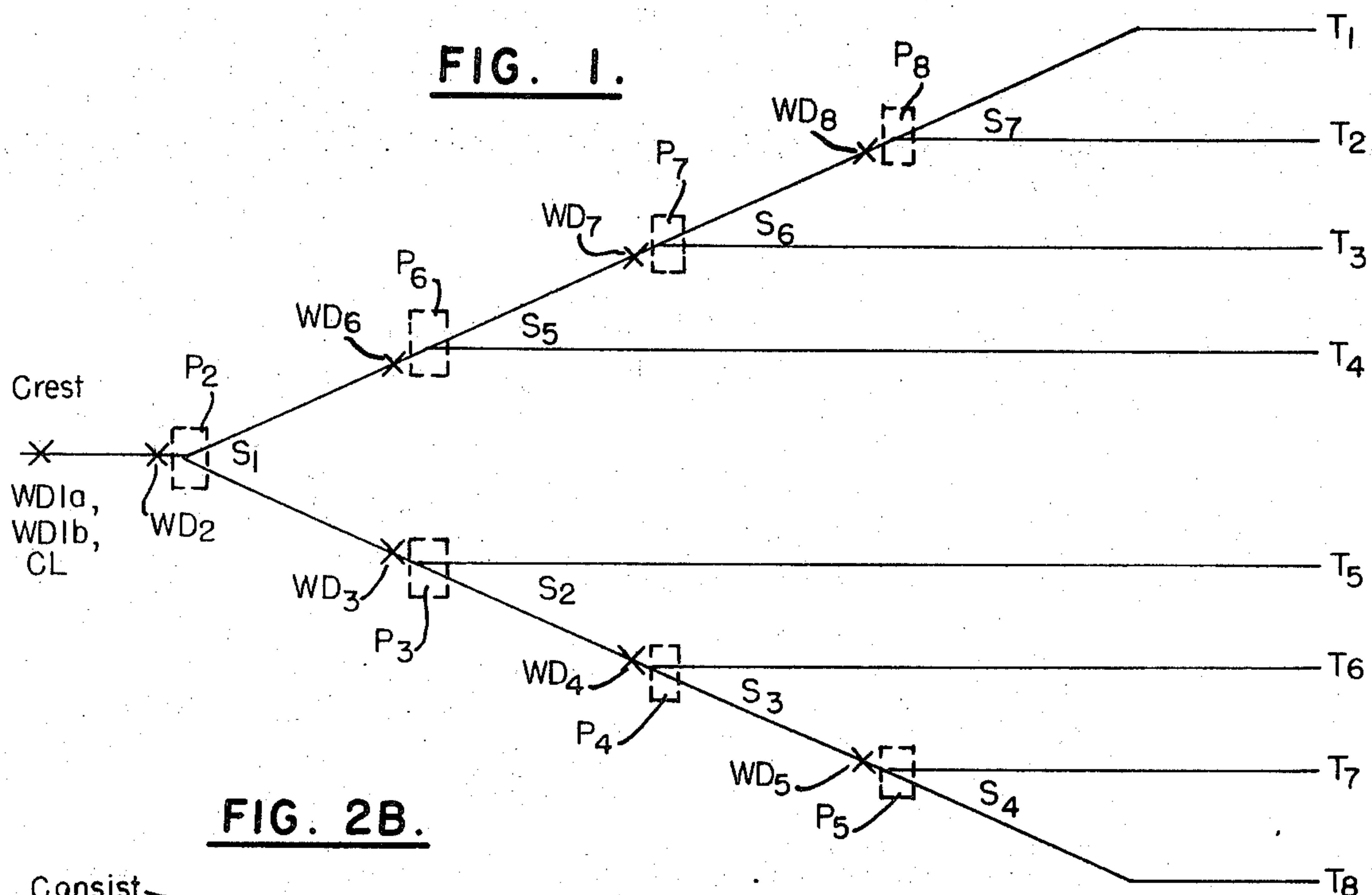
[57] **ABSTRACT**

A method and apparatus for controlling the positioning of switches in a railway classification yard so as to route successive cuts from a train, as they are uncoupled at the hump of the yard, to their respective destination tracks. A digital computer receives information respecting the location of the various cuts as they are traveling through the yard and operates the various switches in the yard to safely route each of the cuts to its destination track. Protection is provided to handle blocked and fouled tracks and also to detect stalled cars and prevent cornering. The computer is responsive to information received from a variety of sensing devices placed throughout the yard and means are further provided to detect failures of these sensing devices in order to enable the system to operate in spite of such failures. The apparatus is so arranged that it can be applied to a wide variety of classification yards with a minimum amount of modification. In particular, the yard characteristics are separately stored in a storage means. To modify the apparatus to control a classification yard with differing characteristics, it is only necessary to modify the yard characteristic data stored in the storage means and the apparatus can be operated in a virtually unchanged manner.

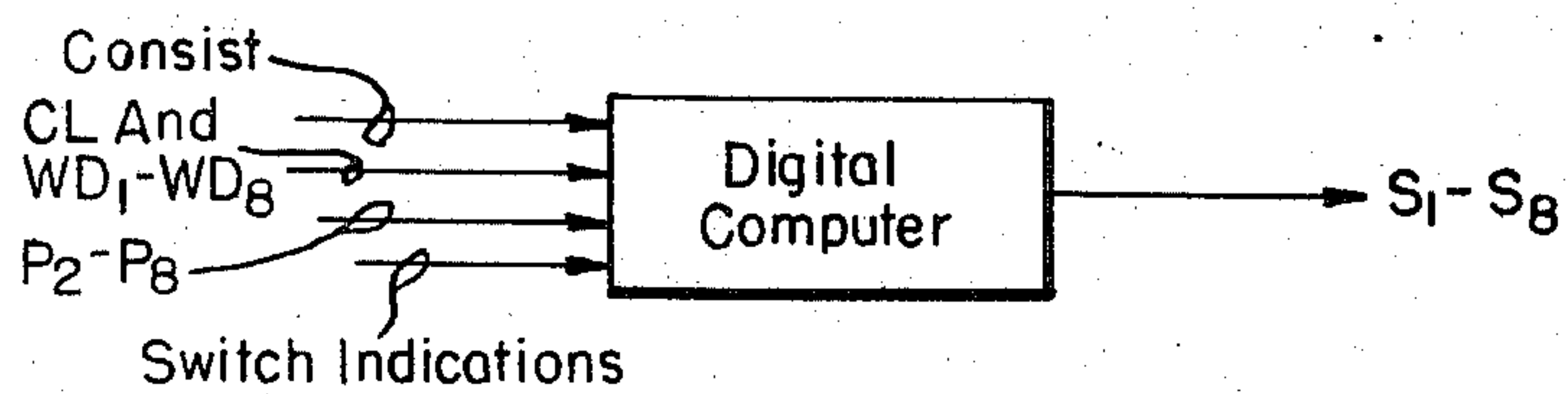
**16 Claims, 22 Drawing Figures**



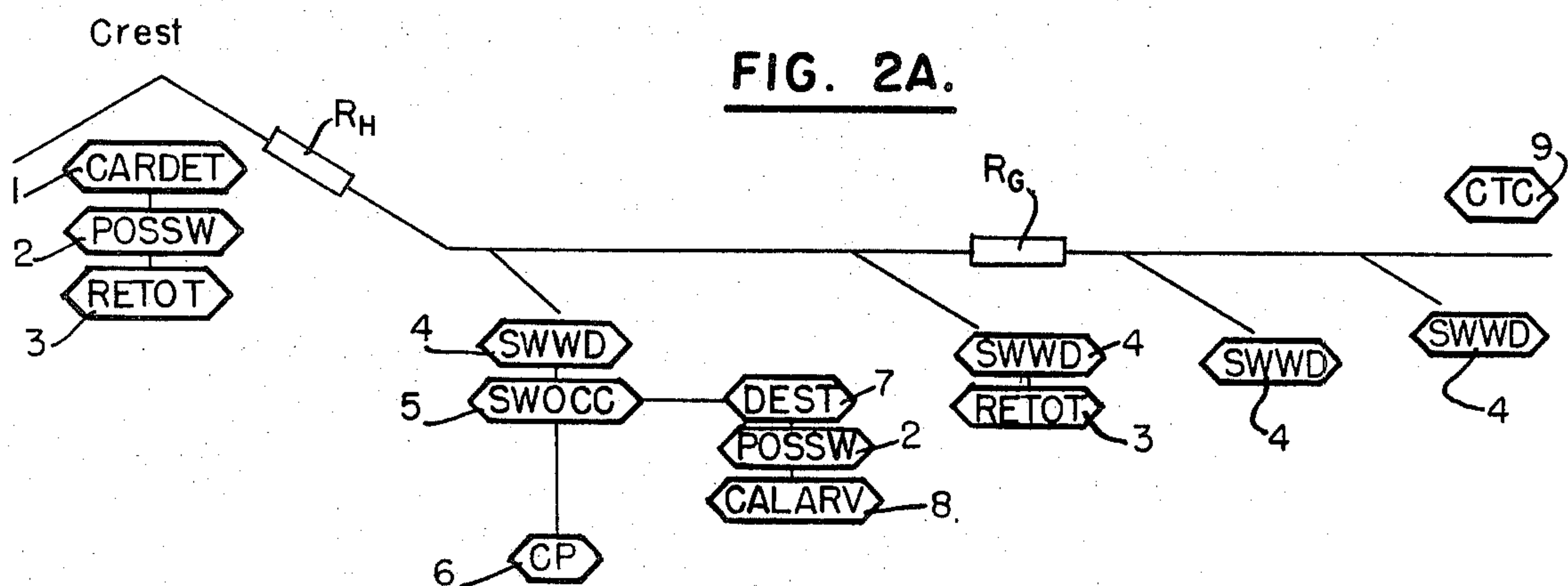
**FIG. 1.**



**FIG. 2B.**



**FIG. 2A.**



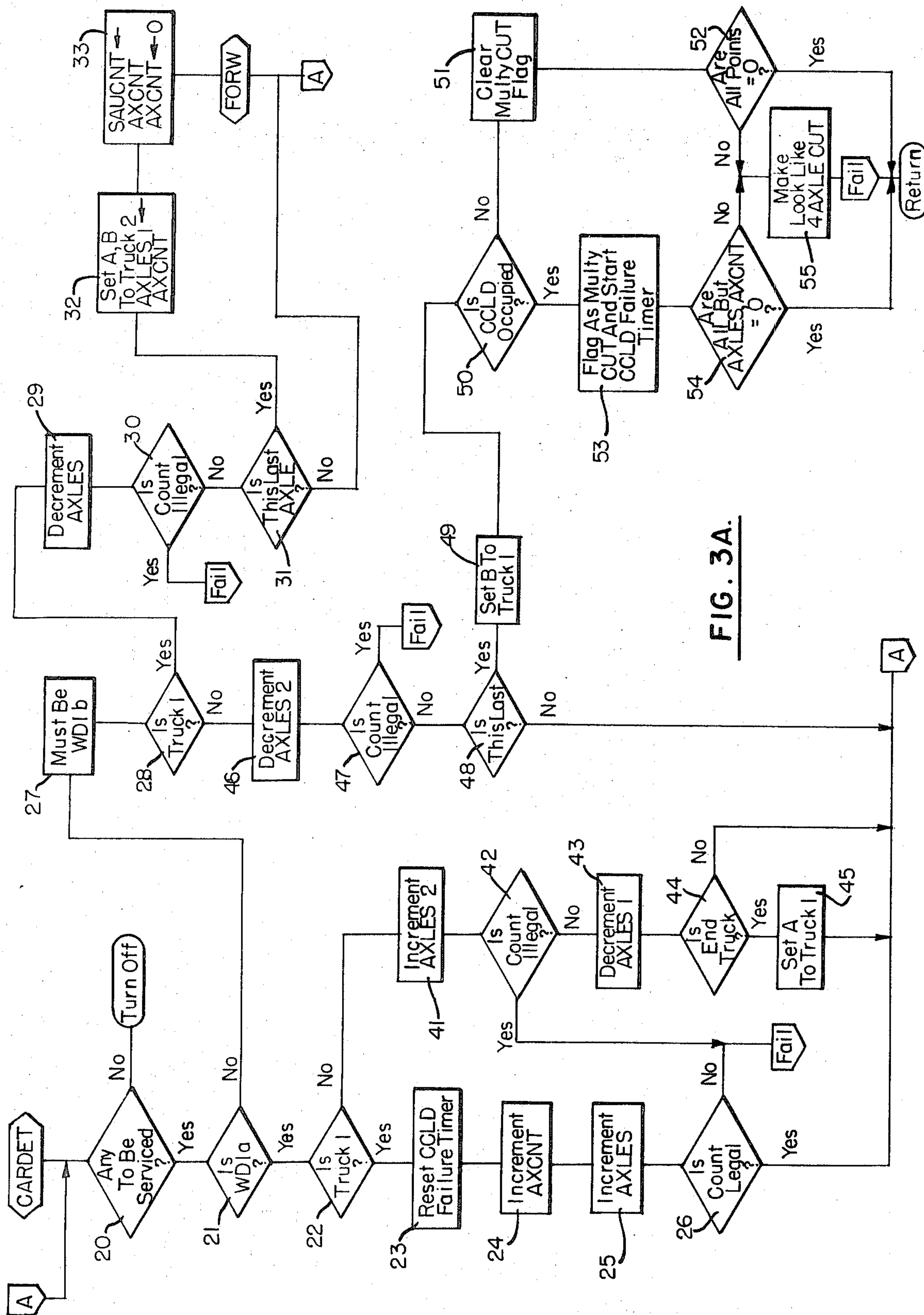
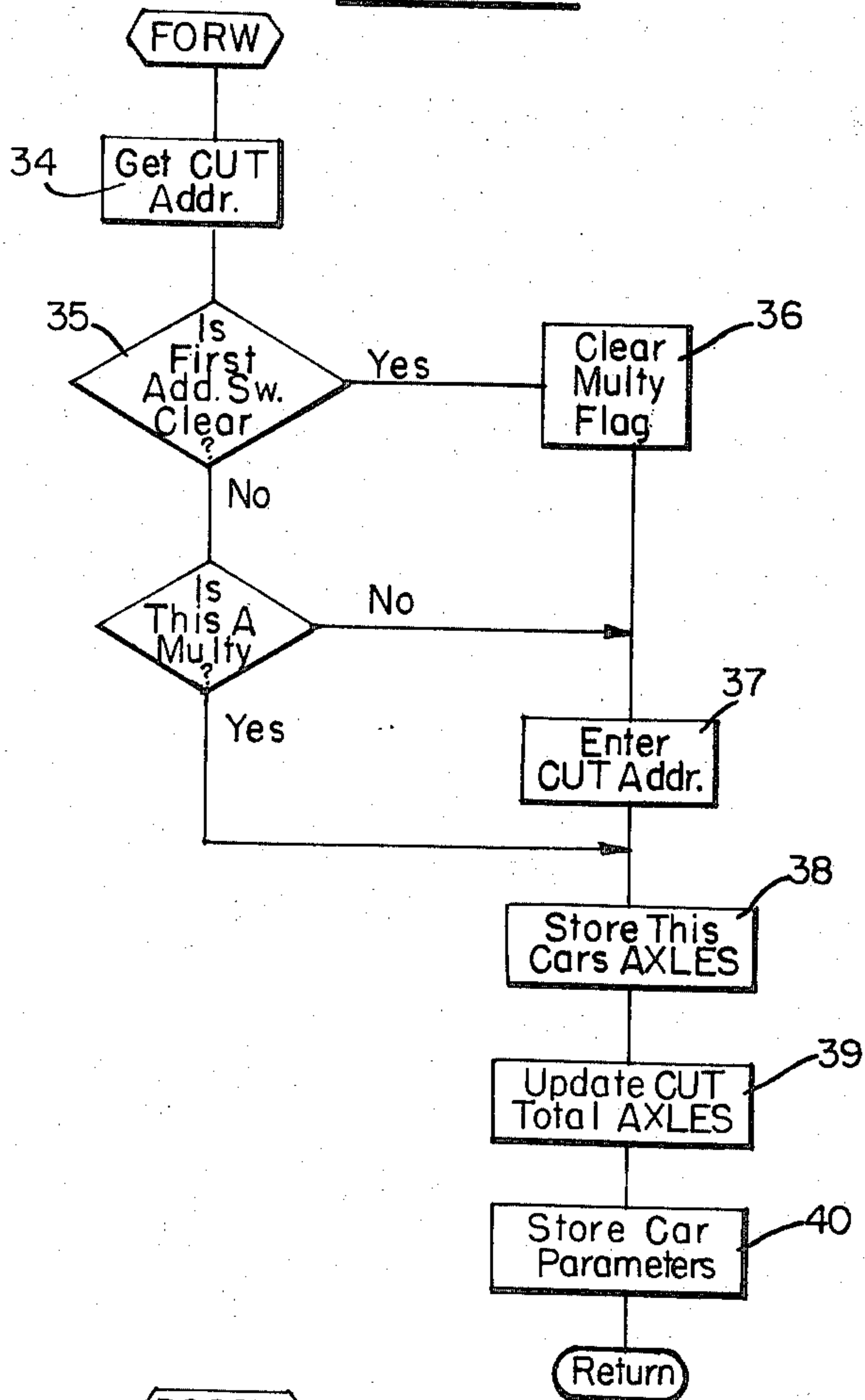


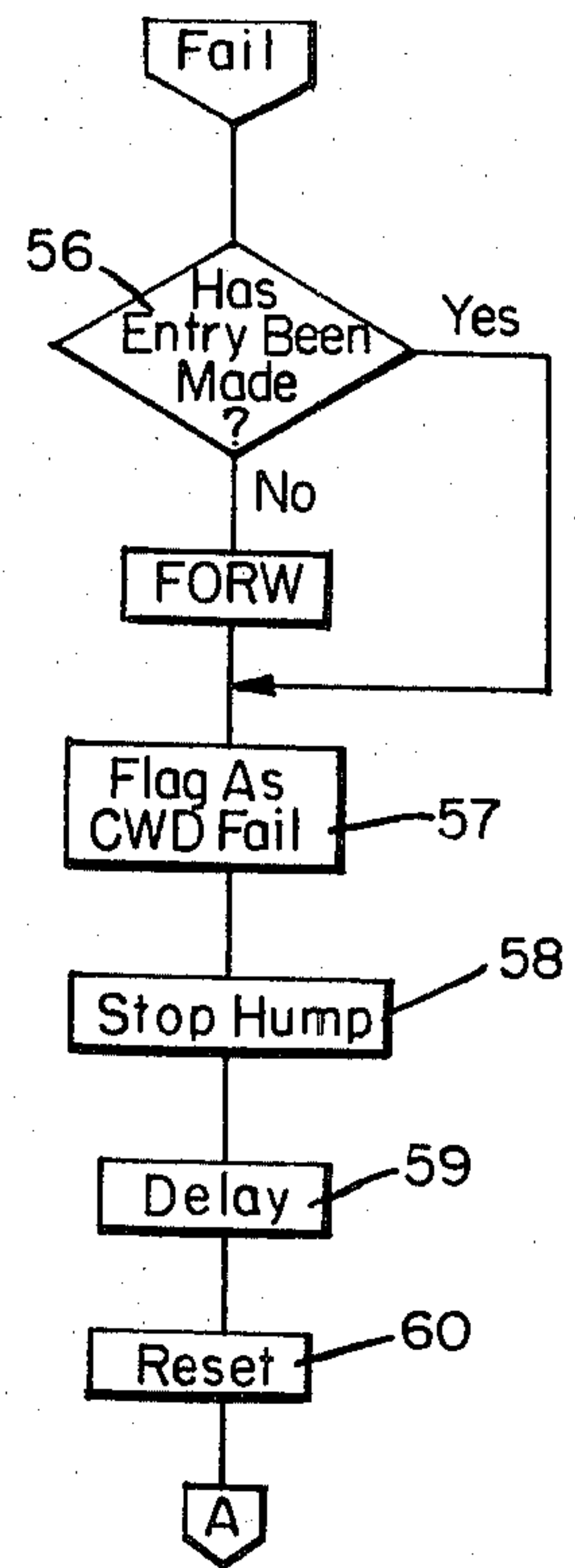
FIG. 3A.



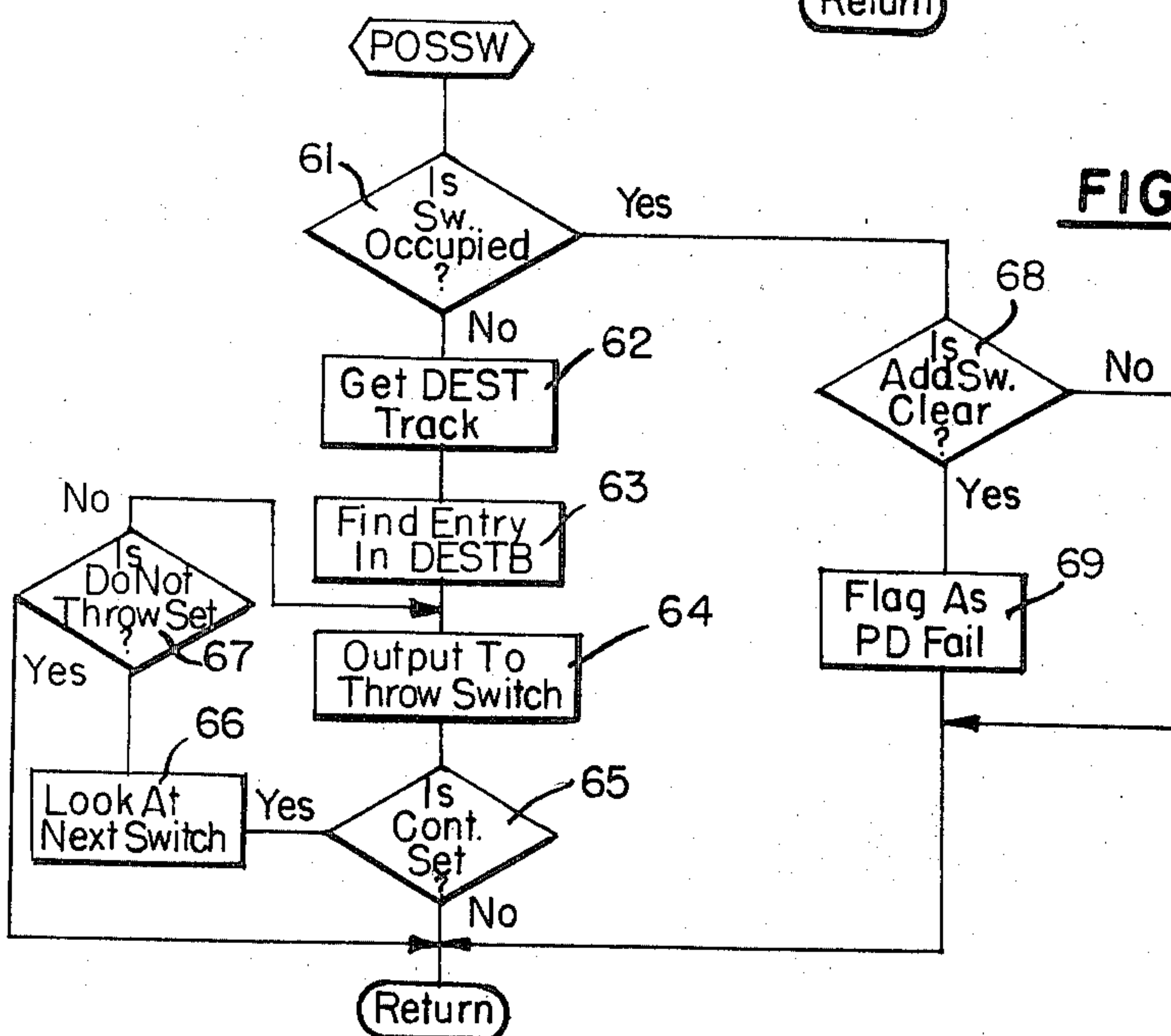
**FIG. 3B.**



**FIG. 3C.**



**FIG. 4.**



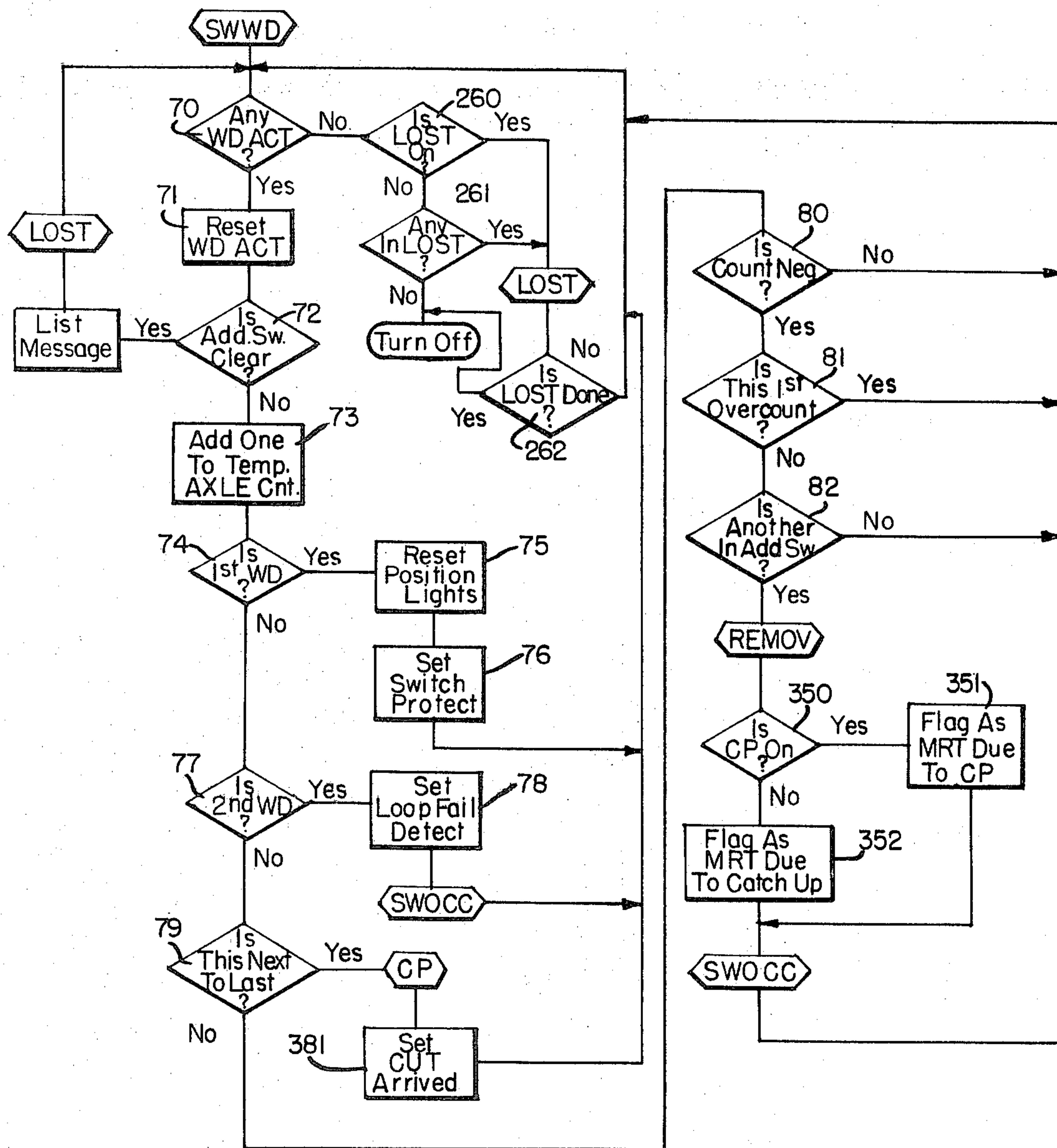


FIG. 5A.

FIG. 5B.

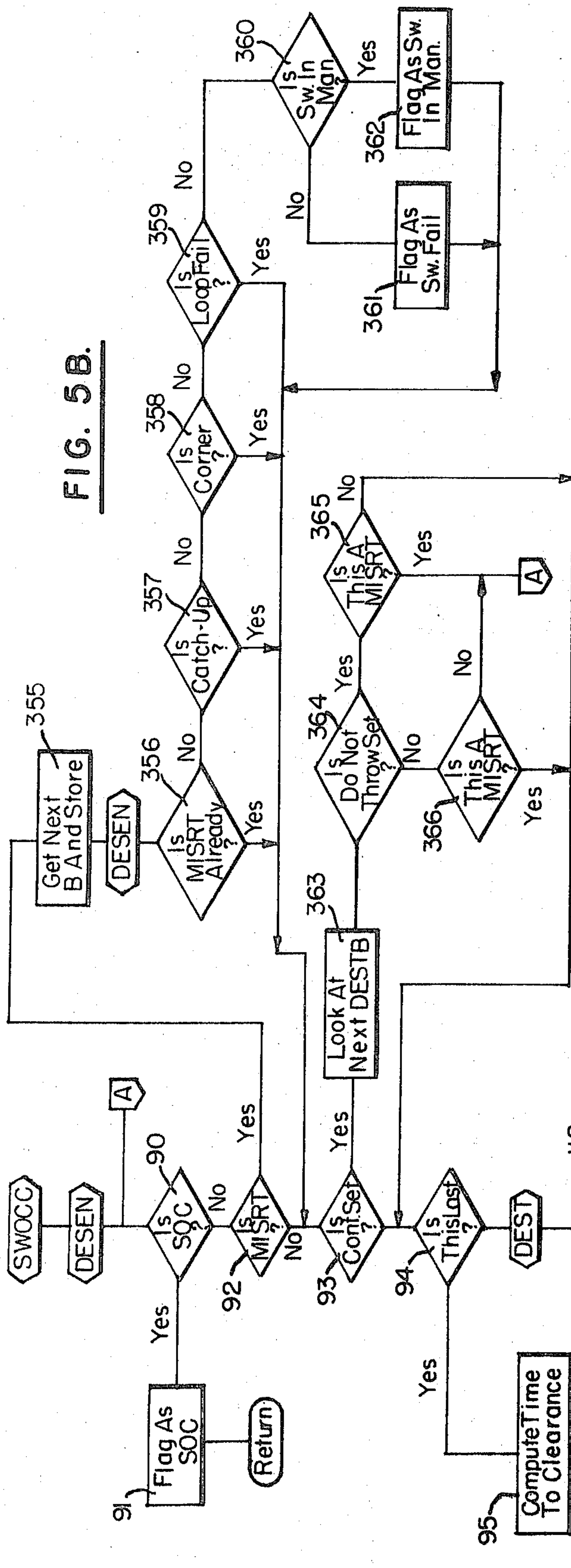
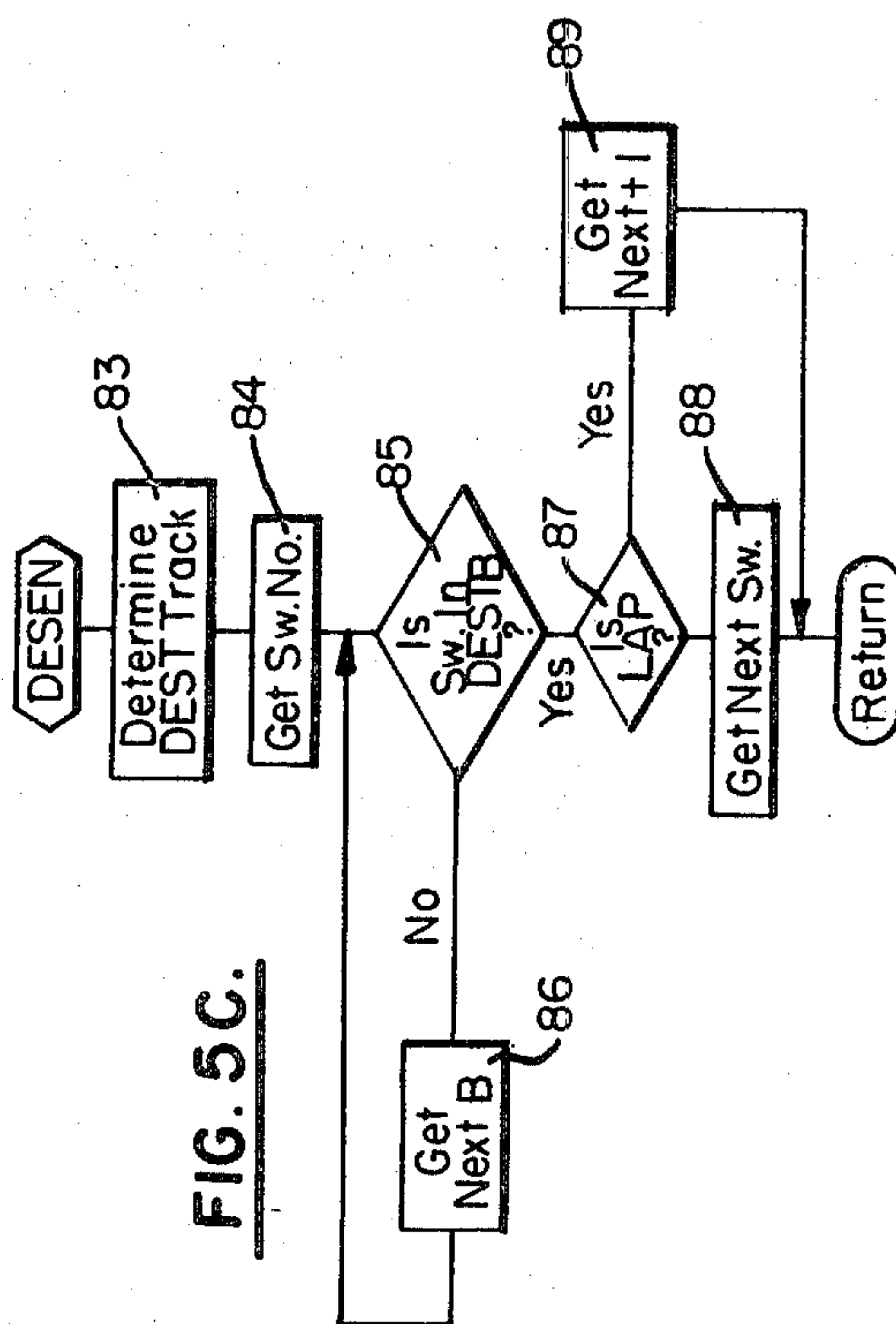


FIG. 5C.



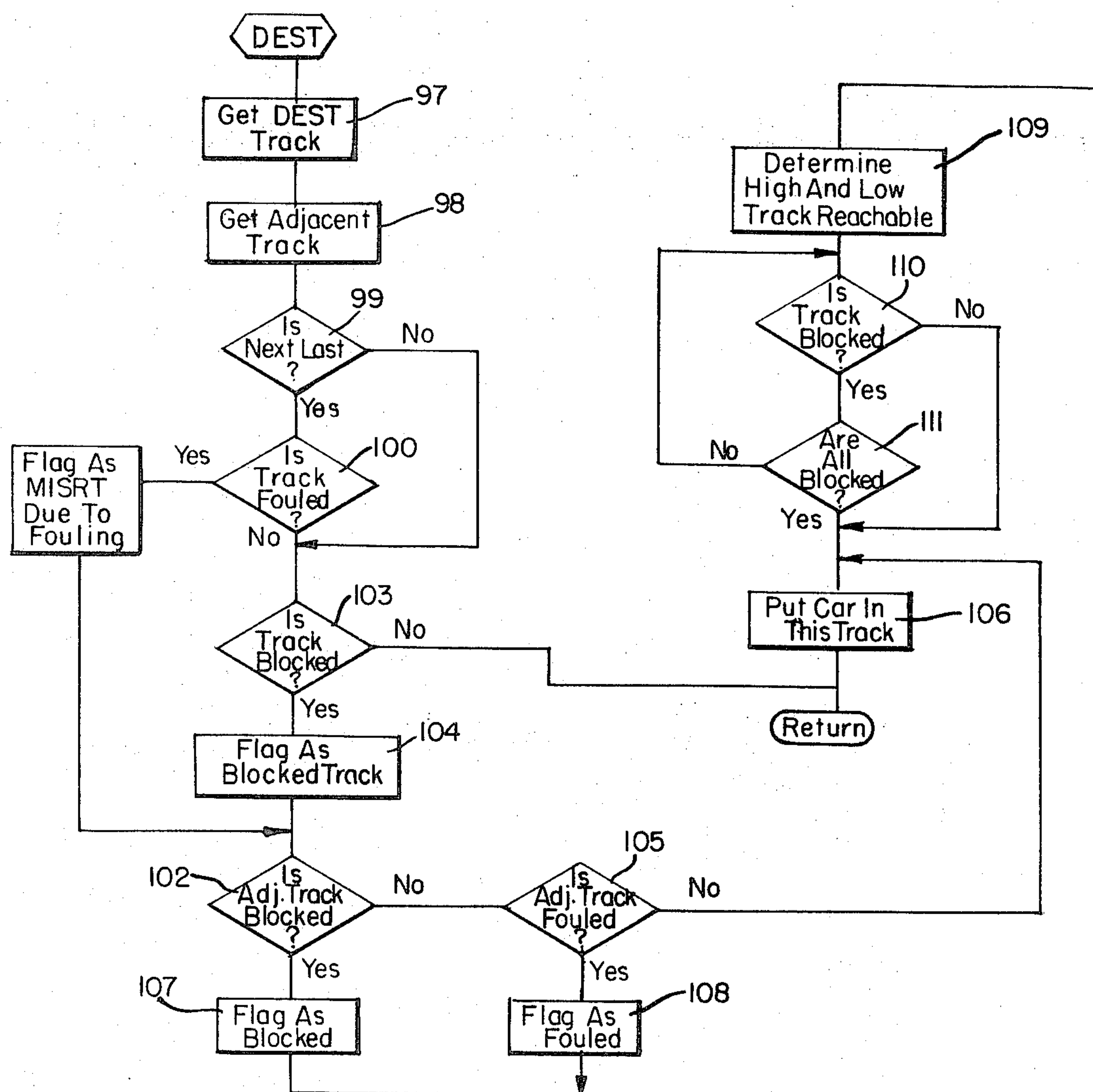
FIG. 5D.



FIG. 5C.

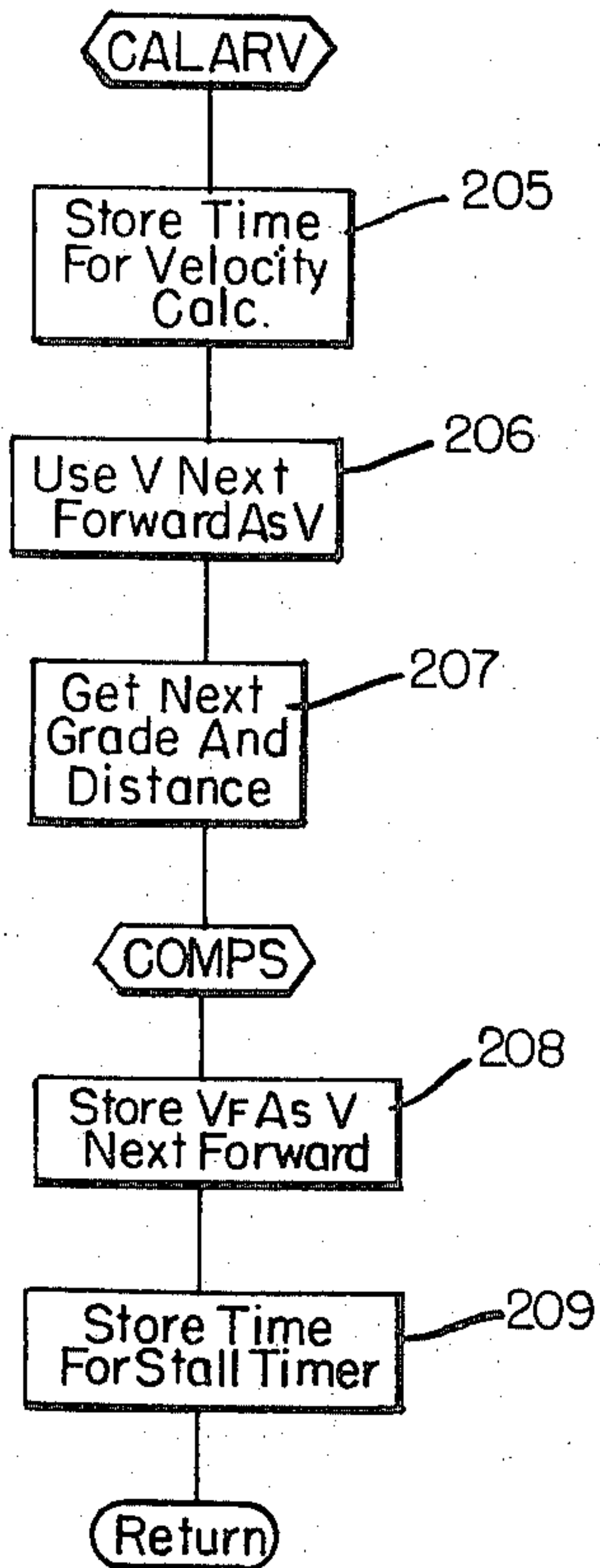


FIG. 6B.

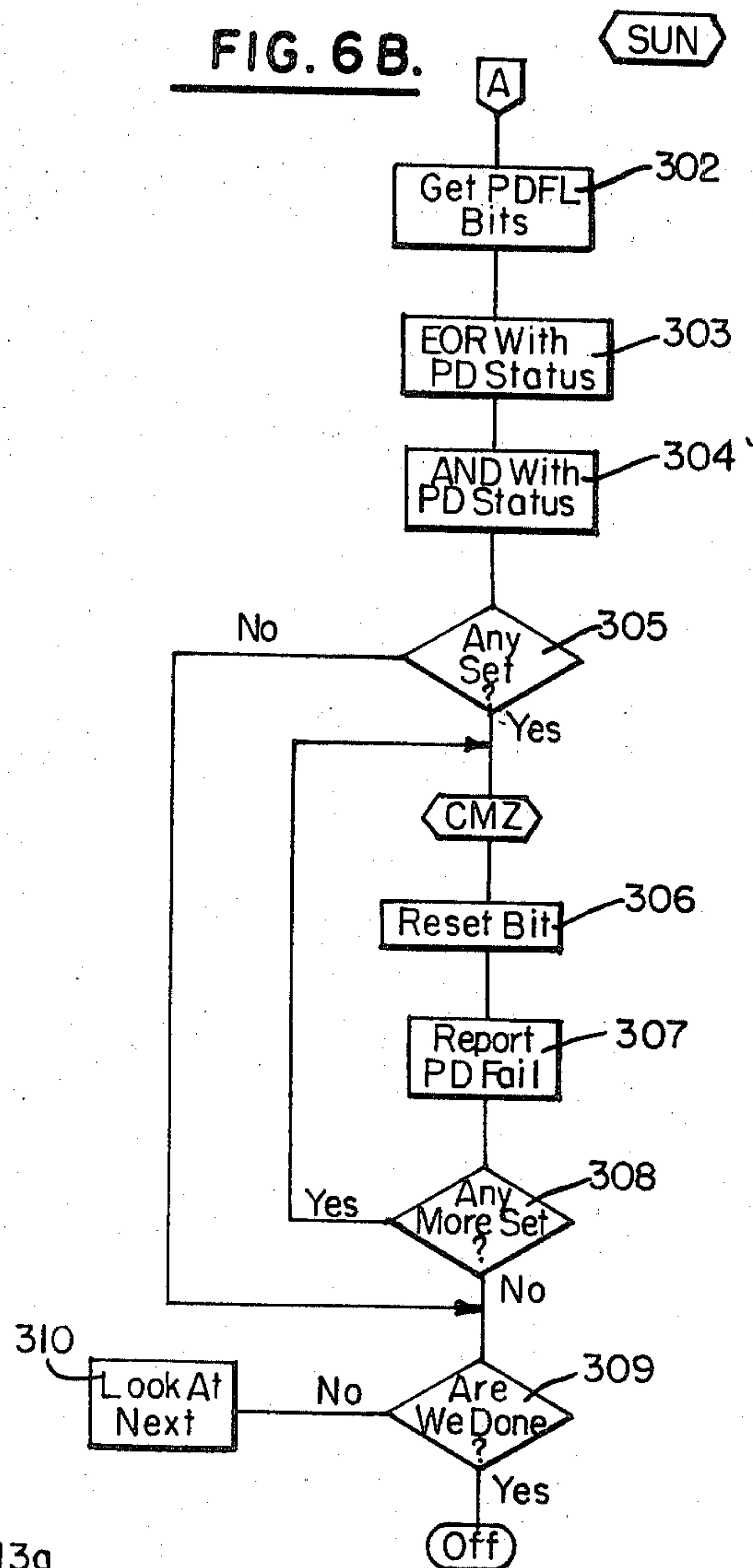
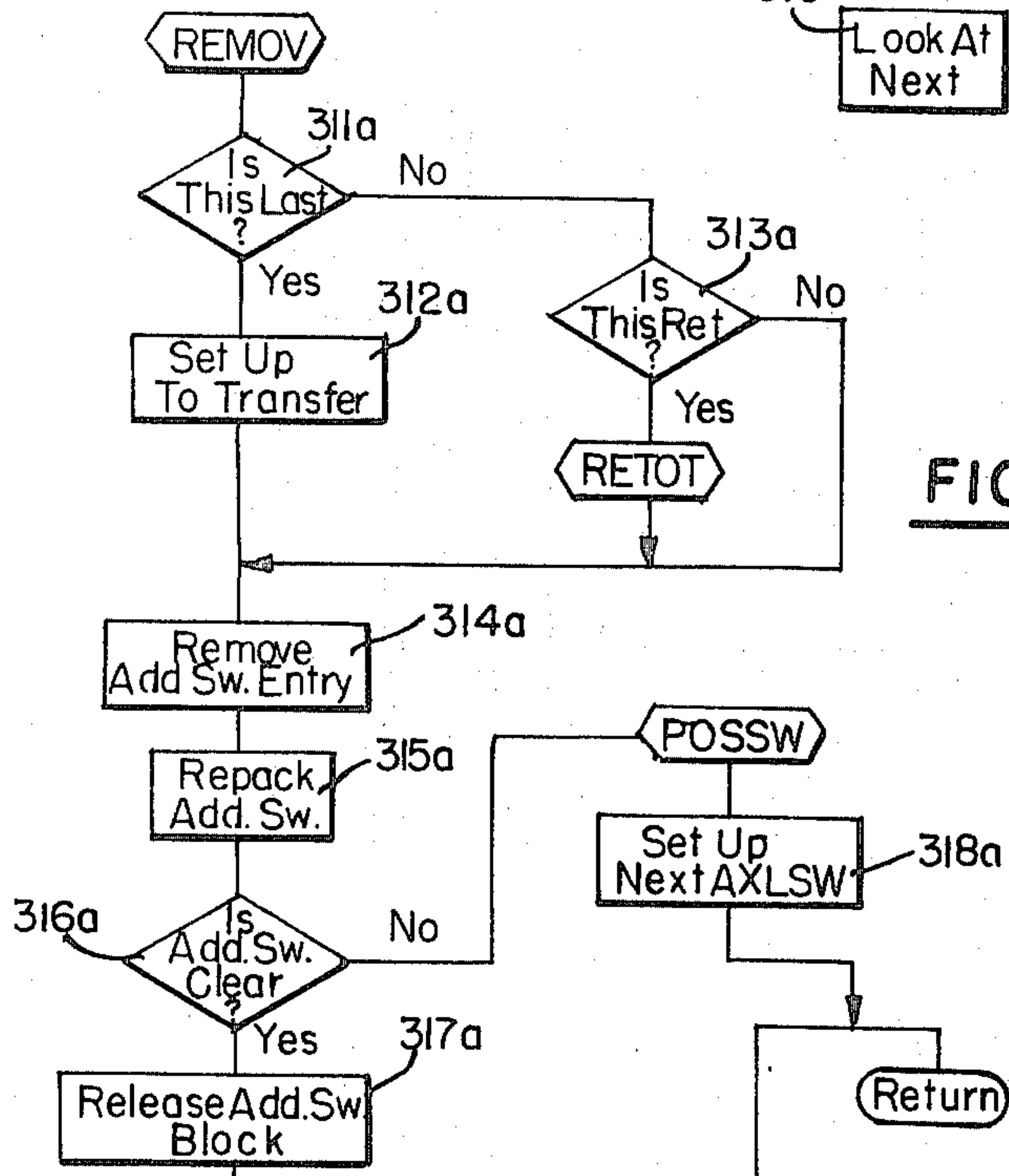


FIG. 6C.





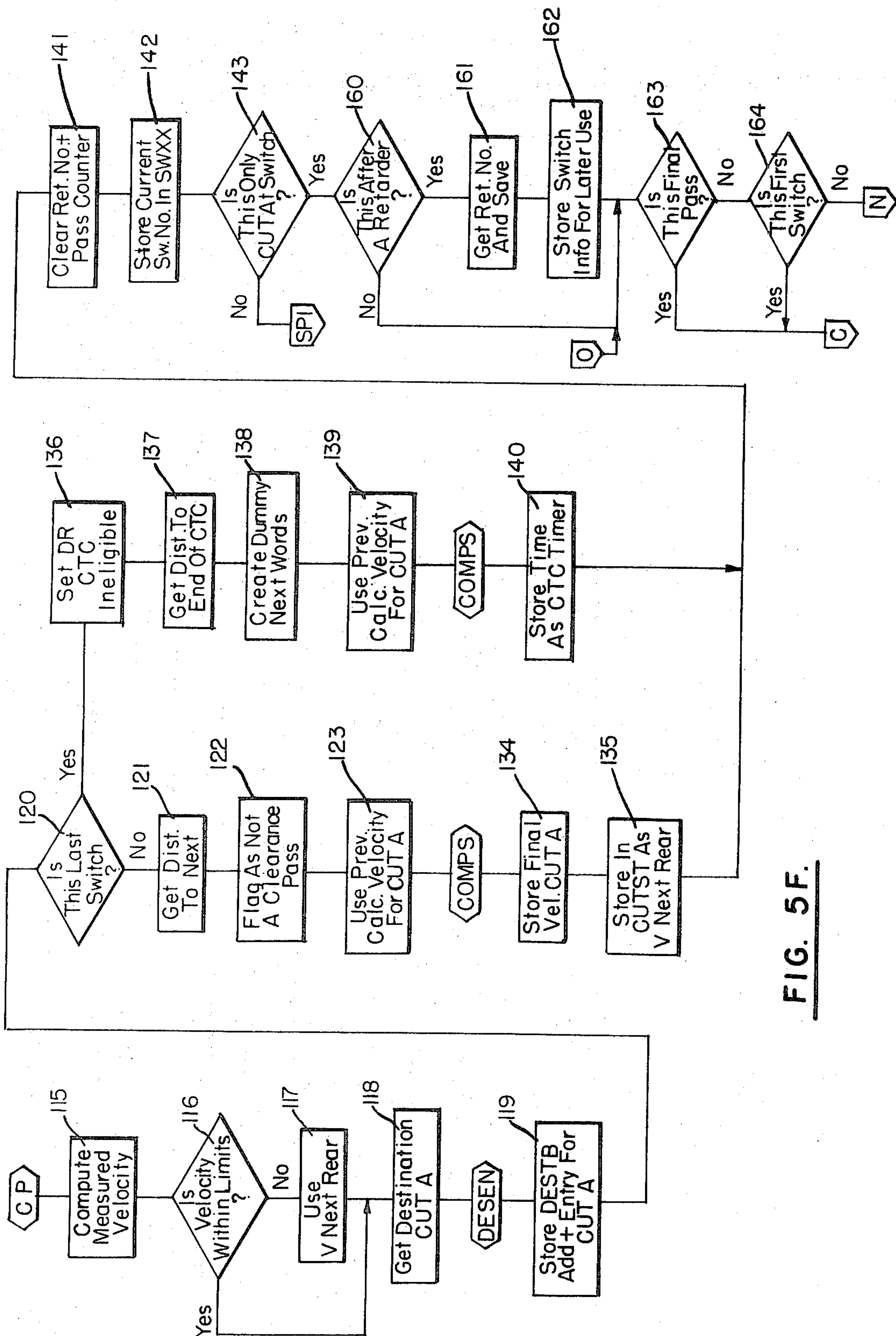


FIG. 5F.

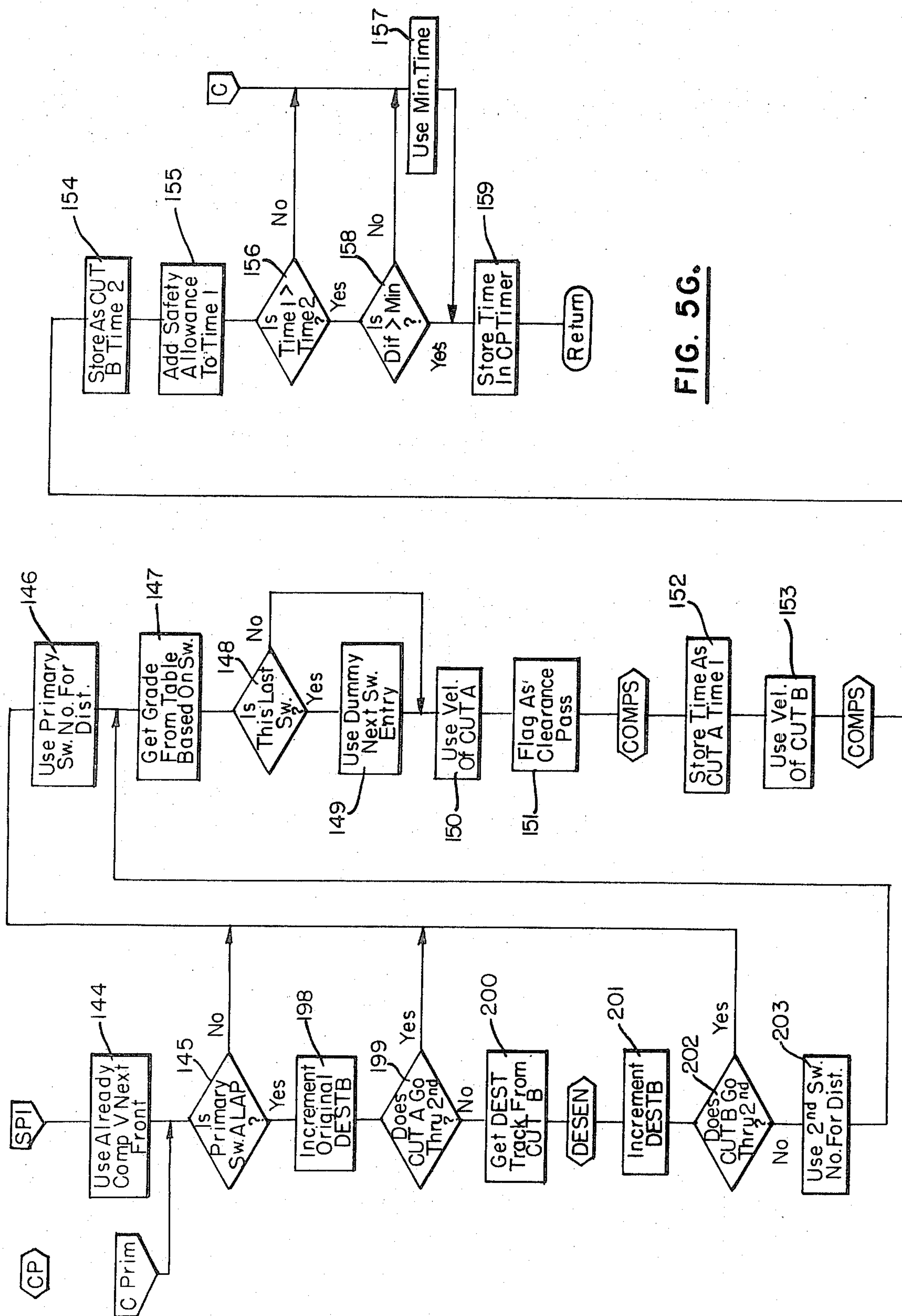


FIG. 56.

FIG. 5H.

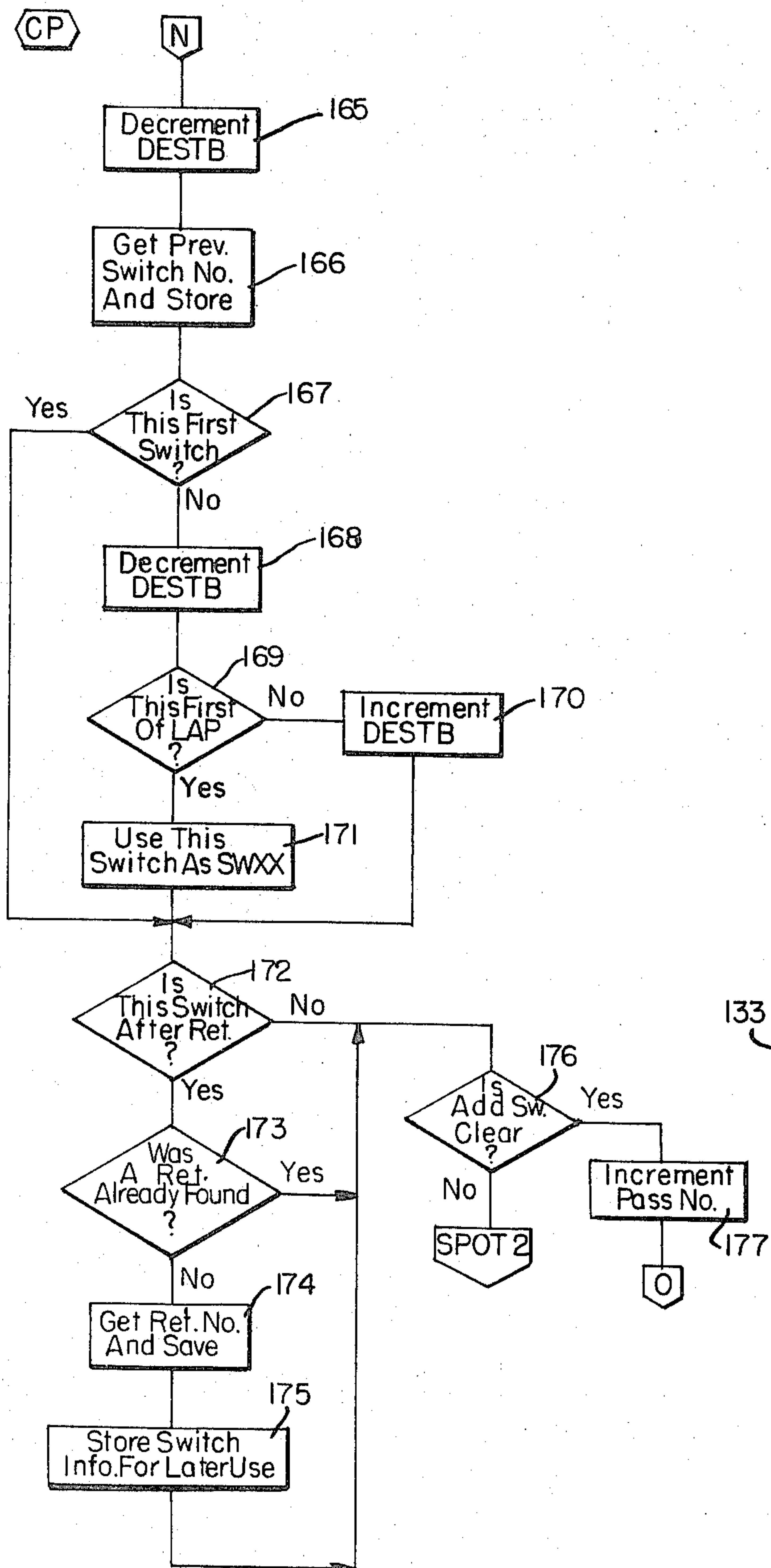


FIG. 5I

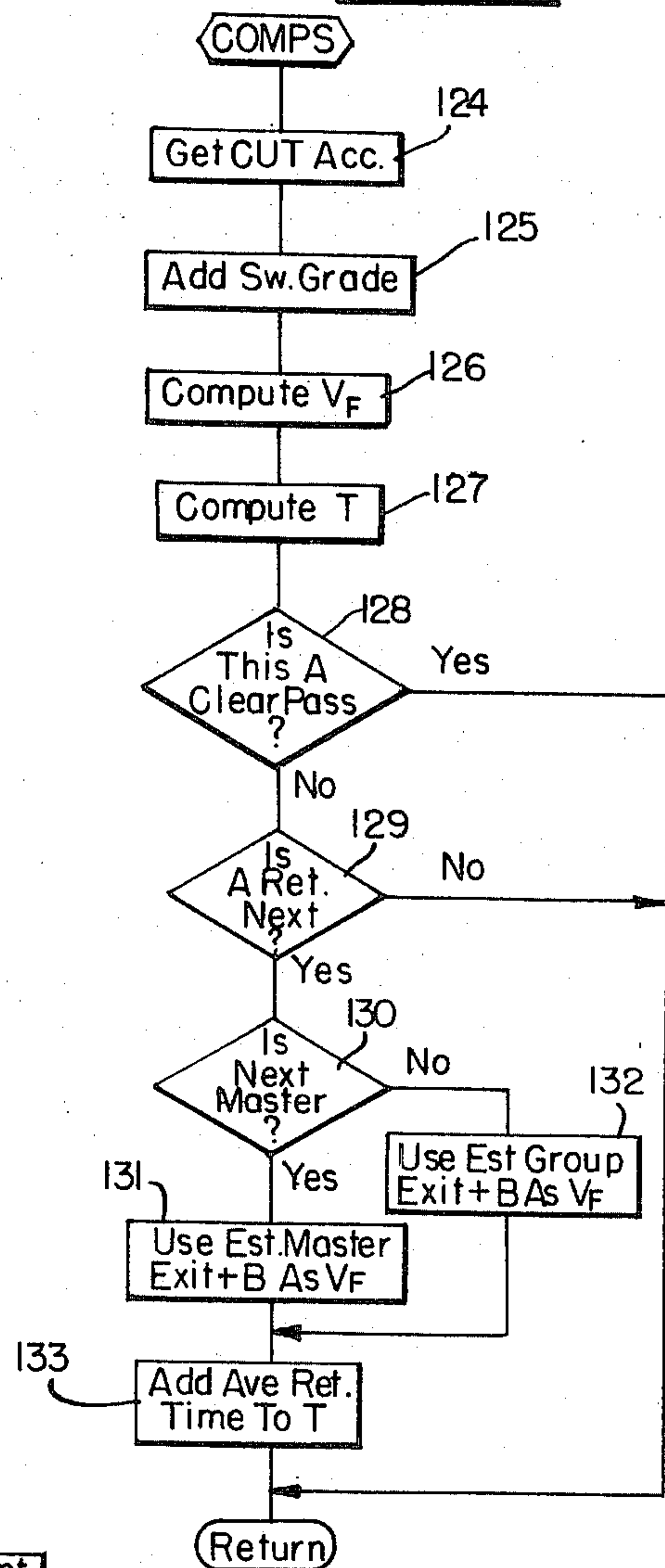
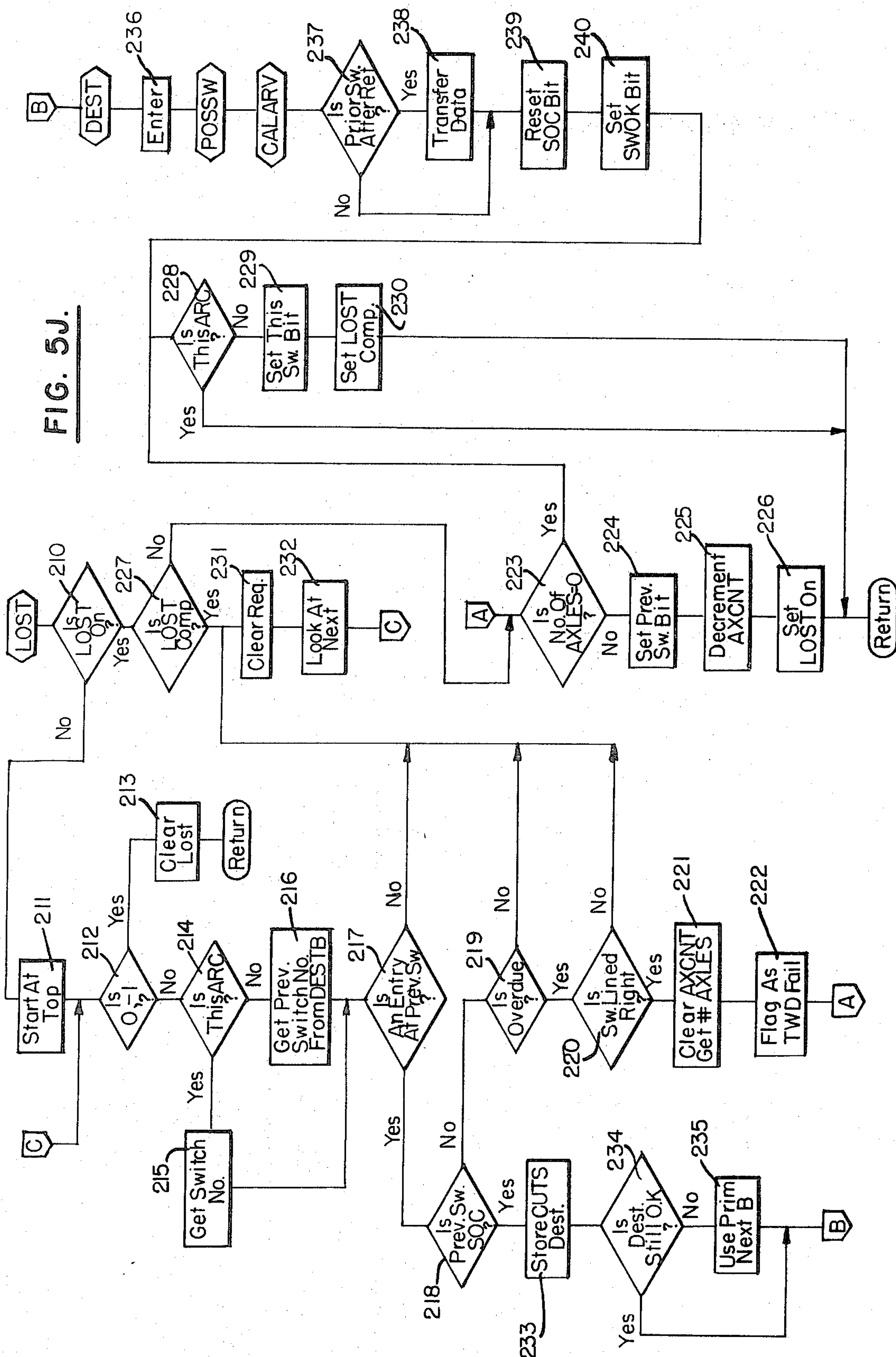
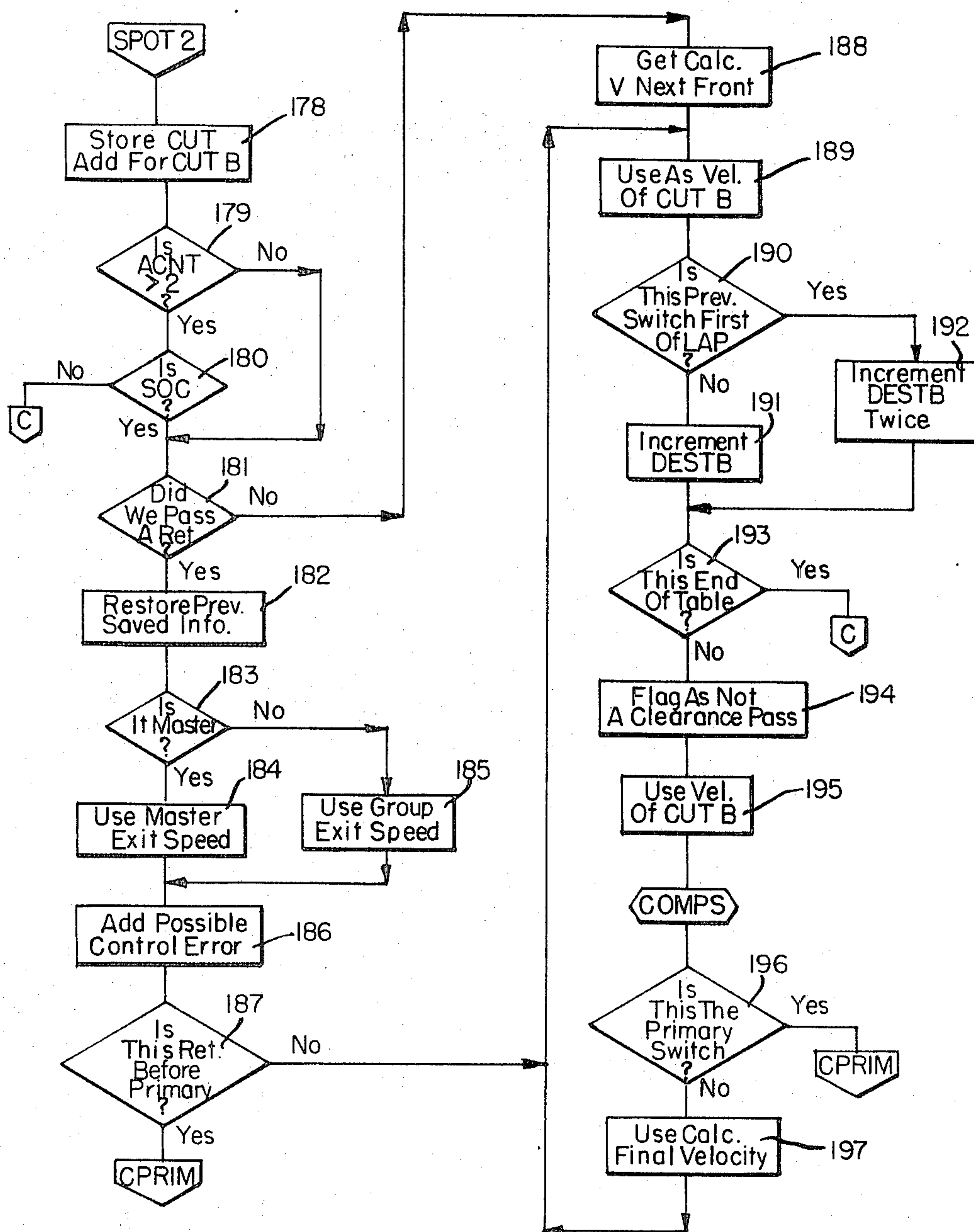




FIG. 5J.



FIG. 5K.

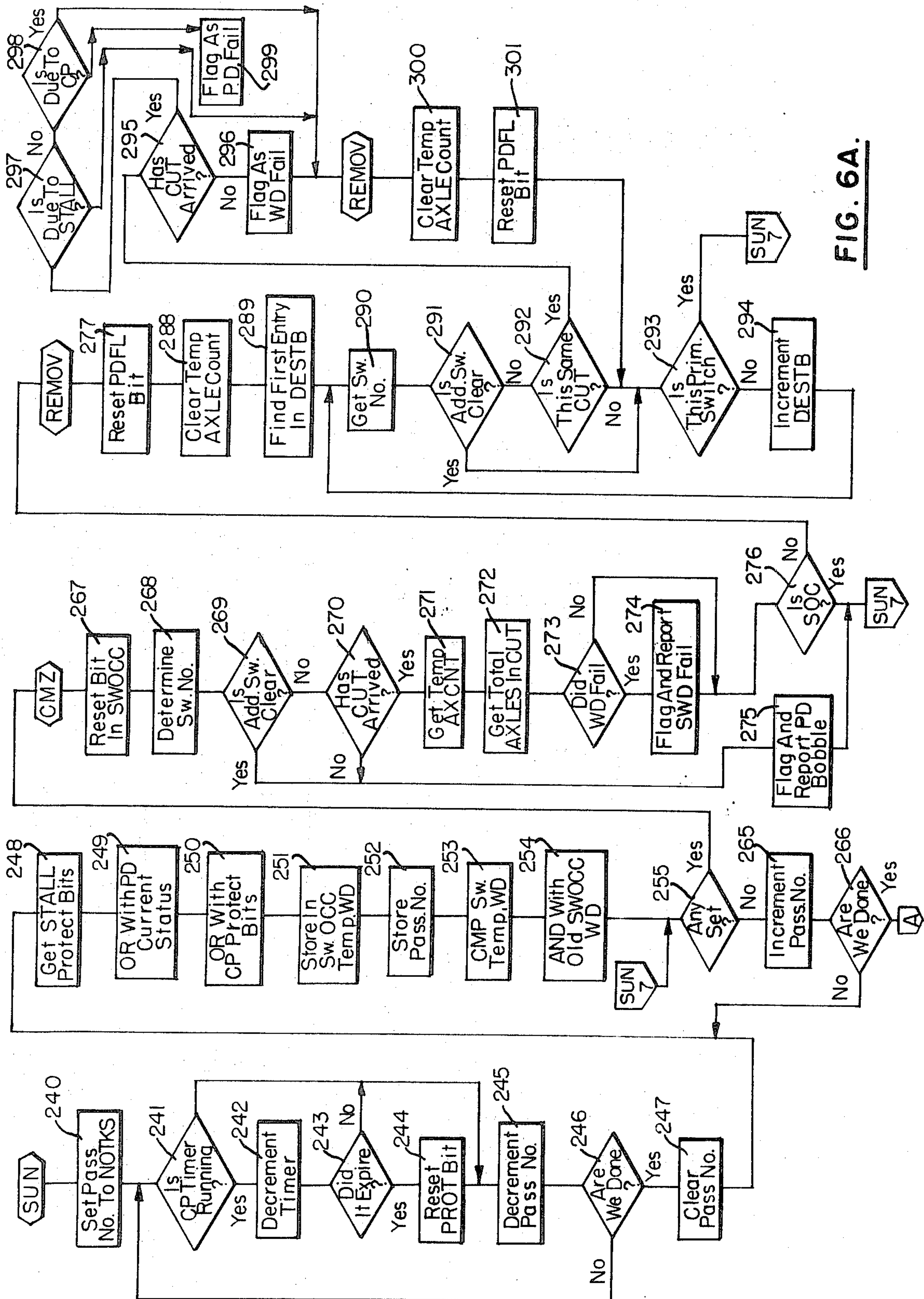
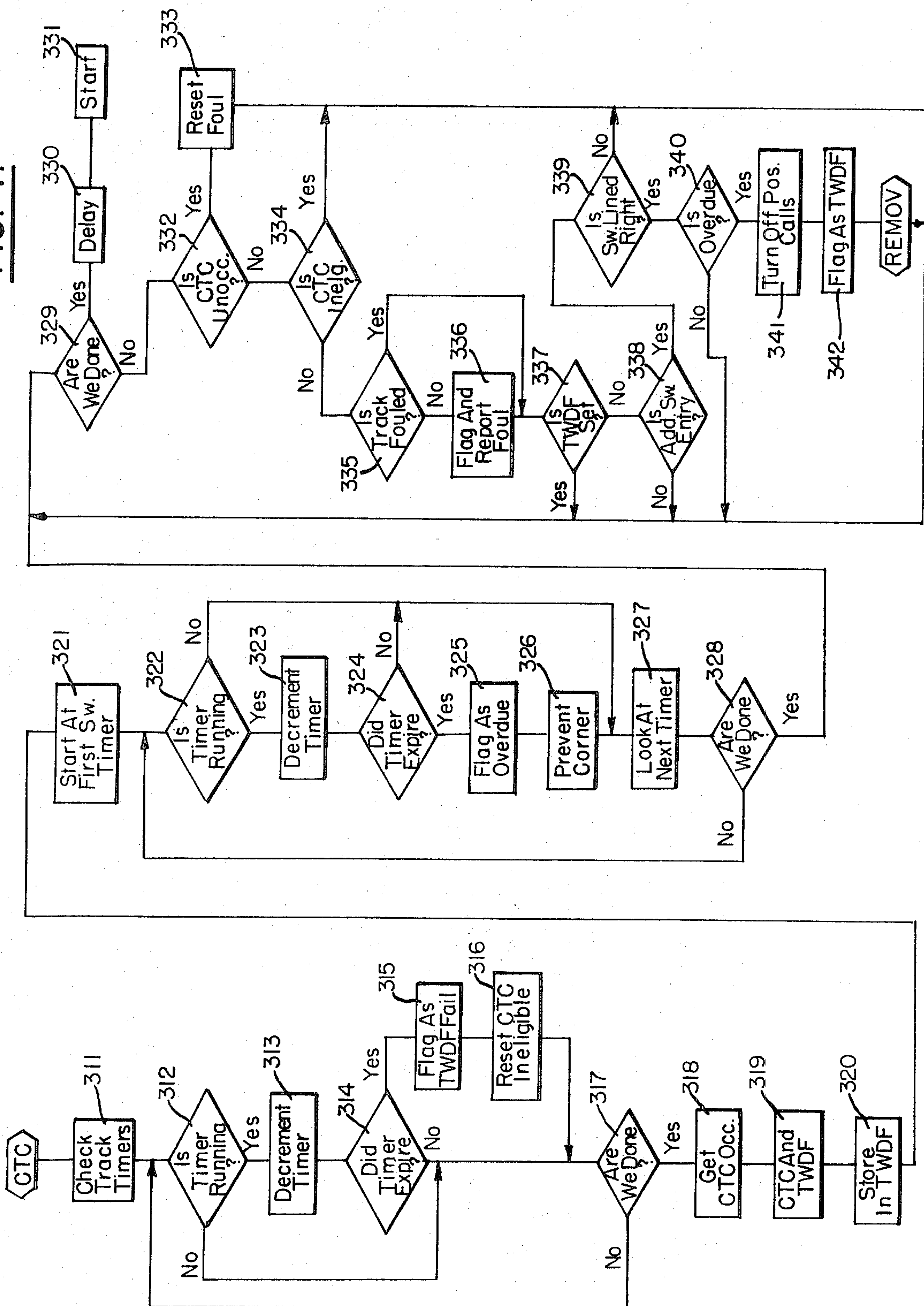




FIG. 7.





## AUTOMATIC SWITCHING CONTROL SYSTEM FOR RAILWAY CLASSIFICATION YARDS

### BACKGROUND OF THE INVENTION

Typically, railway classification yards perform a sorting function. A train of cars is received and pushed up to the hump of the yard. At the hump, the cars are uncoupled and allowed to roll down the hump and the switches in the yard are positioned so as to route each of the different cuts taken from the train, to its destination track. The cars in the train may have originated from a single location but commonly have ultimate destinations at widely separate locations. The railway classification yard segregates the train into a number of different groups, each of the groups having cars with the same or nearby ultimate destinations. A typical railway classification yard is shown, in plan view, in FIG. 1. This shows a yard with eight destination tracks so that the incoming train can be broken up into as many as eight different groups of cars. FIG. 2 shows an elevation view of the yard showing one typical route through the yard.

In the earlier days of use of railway classification yards, the switching functions were manually controlled. That is, an operator, stationed in a control tower, manually positioned the switches in the yard so as to properly route a particular cut to its destination track. However, the period of time during which a car may be traveling from the crest to its destination track can be appreciable and, in order to avoid conflicts, these manually controlled yards were used limited to only one or two cars actually rolling through the yard at the same time. In an attempt to increase the capacity of these railway classification yards, automatic control apparatus, based on relays, was provided so as to follow a particular cut through the yard and properly position the switches immediately ahead of it as necessary to route the cut to its destination track. This enables a plurality of cars to be rolling through the yard at the same time and avoids routing conflicts. However, these control units were large, costly, and complex. A prime factor in the large cost of these control systems was the fact that each one had to be specially designed to fit the characteristics of the particular yard for which it was intended. For instance, such a system for a yard with eight destination tracks could hardly be sufficient to control a yard with 16 destination tracks. Even two yards with the same number of destination tracks might not be able to use identical control apparatus due to differences in the switching arrangement within the yard.

With the advent of the digital computer and associated technology, it became apparent that application of this technology to the railway classification yard control problem could reduce the cost and the complexity of the control equipment required. Instead of tailoring making a relay control system for each yard, it would only be necessary to devise a program for the digital computer, fitted to the yard to cooperate with the sensing and control apparatus in order to properly operate the various switches so as to achieve the functions of the railway classification yard. This approach has proved fruitful and there are a number of digital computer controlled railway classification yards now in operation.

The advent of the computer, with its increased decision-making capabilities over the relay control systems

also allowed the control system to respond to a number of potentially dangerous situations and to control the apparatus in the yard to avoid these dangers. For instance, a cut may become stalled at an intermediate position in the yard. This is a condition which must be recognized and responded to by the control system so as not to route additional cuts into the same track in which another cut is stalled. Furthermore, one of the most potentially dangerous situations in a railway classification yard is cornering. Railway cars are designed to withstand substantial shocks when applied at the car coupler. However, the superstructure of many cars is not designed to withstand similar impacts, and if impacted, will be severely damaged. If one cut traverses a switch in one position and a following cut traverses the same switch in a different position, a potential cornering situation is present. If the second cut is traveling faster than the first cut, the front corner of the second cut may impact the rear corner of the first cut causing damage to both. To prevent cornering, the control system must respond to potential cornering situations by maintaining the switch in its original position for a sufficient period of time to ensure that this cornering cannot occur.

Although the advent of digital computer technology enabled the reduction in costs of fabricating a control system substantially, the complexity of the control problem, of course, was not reduced. A program devised to control the railway classification yard was still costly and complex to develop. Applicants believed it desirable to provide a system in which a digital computer controlled a railway classification yard with a program that was easily modifiable to control a wide variety of classification yards. In particular, applicants sought to devise such a system wherein the logical or decision making functions was segregated from the data which defined the yard characteristics. The system disclosed in this application accomplished this objective and is thus adaptable to almost any classification yard. That is, the logical portions of the program can be applied to any classification yard and the tables which define the yard characteristics need only be modified in accordance with the characteristics of the yard to which the program is to be applied.

The system disclosed herein also provides improved cornering protection. The cornering protection system is capable of searching through the portion of the yard behind any particular cut to search for another cut which is destined for the same switch. The second cut's present velocity is determined and then the velocity is projected so as to determine the velocity of the second cut at the switch where a potential cornering problem exists. The system then computes the times necessary for both cuts to reach a point beyond which cornering is no longer possible. The switch is then held in its original position for a time sufficient to ensure that no cornering can take place. If the second identified cut reaches the switch before the expiration of this protection time, it is routed to follow the first cut which will of course make cornering impossible.

In addition, the system is designed to respond to and safely handle failures in sensing device operation without creating dangerous conditions.



## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a plan view of a typical railway classification yard, schematically illustrating the placement of sensing apparatus of the control system;

FIG. 2A is an elevation view of a typical route in a railway classification yard shown in FIG. 1, identifying major routines which are performed when a cut is sensed at that specific location in the route;

FIG. 2B is a schematic showing of the interrelationship between the digital computer and the various sensing and control devices in the yard;

FIG. 3A and FIG. 3C, taken together, are a flow diagram for the routine entitled "CARDET";

FIG. 3B is a flow diagram for the routine entitled "FORW";

FIG. 4 is a flow diagram for the routine entitled "POSSW";

FIG. 5A is a flow diagram for the routine entitled "SWWD";

FIG. 5B is a flow diagram for the routine entitled "SWOCC";

FIG. 5C is a flow diagram for the routine entitled "DESEN";

FIG. 5D is a flow diagram for the routine entitled "DEST";

FIG. 5E is a flow diagram for the routine entitled "CALARV";

FIG. 5F, FIG. 5G, FIG. 5H, and FIG. 5K, taken together, are a flow diagram for the routine entitled "CP";

FIG. 5I is a flow diagram for the routine entitled "COMPS";

FIG. 5J is a flow diagram for the routine entitled "LOST";

FIG. 6A and FIG. 6B, taken together, are a flow diagram for the routine entitled "SUN";

FIG. 6C is a flow diagram for the routine entitled "REMOV";

FIG. 7 is a flow diagram for the routine entitled "CTC".

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In a railway classification yard, a train of incoming cars is pushed up to the crest or hump of the yard and then the cars are uncoupled and allowed to roll down the crest and through the yard to their intended destination tracks. Destination tracks for each of the cars of the train are assigned with respect to the ultimate destination of each of the cars. When adjacent cars of a train have the same ultimate destination or nearby ultimate destinations, they may be assigned identical destination tracks. In this case, the adjacent cars remain coupled and such cars are called a cut. For the purposes of this description we shall refer only to a "cut," and a "cut" may comprise one or more than one car intended for the same destination track.

At or before the time of arrival of the train into the railway classification yard, a consist will be available. The consist is merely a list of the cars in the train in the order in which they will arrive, identifying the car by its owner and serial number, the number of axles in the car, and, most importantly, its intended destination track. After a cut is uncoupled from the train and allowed to roll down the hump, the apparatus disclosed herein will properly position the switches in its path so

as to route the cut to the correct one of the destination tracks in the yard.

FIG. 1 shows an illustrative railway classification yard with eight destination tracks, T1-T8. It will, of course, be understood that the number of destination tracks in an actual railway classification yard can vary, being either more or less than the number of destination tracks shown in FIG. 1.

Located at the crest is wheel detector 1 (WD1), which actually comprises a pair of wheel detectors WD1a and WD1b. Additional wheel detectors, WD2 through WD8, are located prior to each switch in the classification yard. FIG. 1 shows seven switches in the classification yard, S1-S7. Also adjacent each of the switches in the yard is a presence detector, P2-P8. Both the wheel detectors WD1-WD8 and the presence detectors P2-P8 sense the presence of a cut. Each of the wheel detectors produces a signal whenever a wheel (also referred to as an axle) passes thereover. On the other hand, each of the presence detectors produces a signal when a cut is in the space defined by the presence detectors. Both the wheel detectors and the presence detectors provide real time information as to the location of the various cuts throughout the yard at any instant of time. In addition to detectors WD1a and WD1b at the crest, a further detector CL is also located adjacent the crest. This comprises a light source and a light detector which are located on opposite sides of the track at the height of the railway car couplers. The light detector is so located with respect to wheel detectors WD1a and WD1b that when there is only one car in a cut, the previously interrupted light path will be restored after passage of the car. However, when a cut is made up of more than one car, the light path will continue to be interrupted by the coupler of the second car.

In order to properly control the switches of the railway classification yard, the apparatus of the present invention employs a digital computer which is loaded with the program illustrated in FIGS. 3-7. The program itself and the manner in which it operates will be discussed in detail with respect to FIGS. 3-7. However, FIG. 2B shows the information flow paths both to and from the computer. The first class of information that the computer receives, after being loaded with the program and the tables defining the yard characteristics, is the train consist which, as explained above, is a sequential list of the cars of the train as they will appear at the crest along with data with respect to each of the cars. As the cars are uncoupled at the crest, detectors WD1a, WD1b and CL provide sufficient information to determine the number of cars in the cut and the number of axles in each cut.

The remaining information received by the computer consists of real time information related to the status of sensing devices throughout the yard which reflect the changing conditions of the yard. In particular, indications are received from detectors CL and WD1a and WD1b, located at the crest. Further information is received from wheel detectors WD2-WD8 and also the presence detectors P2-P8. These sensors indicate the passage of cuts and the identification of a cut as either single or multi-car. Furthermore, the computer receives switch indications from each of the switches in the yard. Each switch may be indicating either normal, reverse, or neither of the two. This latter case is possible if for some reason the switch cannot be fully thrown



to one or the other of its positions by reason of some physical blockage or failure in the switching equipment.

In accordance with the available information and based on the logic found in the program, the computer determines the required position for the switches and transmits these switch calls to the field to control the positioning of the switches.

FIG. 2A is an elevational view of the yard showing the crest at the left and the path of one typical route to a particular destination track. Four switches have been shown in the route in accordance with FIG. 1. The switches are unidentified to make the view completely general. In addition, two car retarders, RH, the hump retarder, and RG, a group retarder, are shown, for illustrative purposes, since the system disclosed herein does not control the retarders. A related application Ser. No. 339,473, entitled "Car Retarder Control System," filed Mar. 8, 1973, discloses a system that is useful for controlling retarders in a railway classification yard.

Illustrated in FIG. 2A adjacent each switch and the crest of the yard is a listing of some of the relevant routines which are called into play by the presence of a cut at the identified location. For instance, when a cut is detected at the crest, the routines CARDET, POSSW and RETOT are called into play. The routine CARDET responds to the signals produced by detectors WD1a, WD1b, and the cut light detector CL to identify the make-up of each of the cuts. This routine will be discussed in detail with reference to FIG. 3A. The next routine in FIG. 2A, POSSW, determines the positioning of the first switch in the yard in accordance with the intended destination track of the cut which is at the crest and will cause that switch to be thrown to the proper position so long as the switch can be thrown without violating any of the safety rules. The subroutine RETOT merely involves a transfer of information with respect to the cut to the car retarder operating system. Since the car retarding operation system does not form any part of the instant invention, this subroutine will not be further discussed.

Located at the first switch, the subroutine SWWD services the switch wheel detector. Upon the detection of a wheel detector actuation at any switch in the yard, the routine SWWD is called. This routine searches for an entry in a table associated with the switch at which the wheel detector is actuated in order to determine the identity of the cut approaching the switch. If no entry is found, the routine LOST is called to handle the situation. LOST is discussed with reference to FIG. 5J.

Assuming there is an entry at the table associated with the switch, upon the first wheel detector actuation, an entry is made in a software file to prohibit the throwing of this particular switch. Upon the second wheel detector actuation, at this switch, the routine SWOCC is called to determine the positioning of the switch in order to identify whether or not a misroute has occurred. If a misroute has occurred, a new destination is determined for this cut in light of existing conditions. Then, the subsequent portions of the route for the cut are checked to determine whether the intended destination is achievable, and if it is, an entry is made in the table for the next switch in the route to identify this cut. At this time, the routine POSSW will throw this next switch, if such action is achievable.

Subsequently, CALARV calculates the arrival time for the front of this cut at the next switch. This is re-

quired in order to detect stalled cuts, that is, if the cut does not arrive at the next switch within the computed time, the cut is identified as stalled and appropriate protective action is taken.

When the next to the last axle in the cut causes a wheel detector actuation, the routine CP is called which will prevent cornering. Subsequently, a further entry is made in a software file to indicate that the cut has arrived at the switch. Since there is only one remaining axle to be counted in the cut, it is assumed that all cars of the cut have arrived at the switch.

If two excess wheel detector actuations are detected at a switch, it is assumed that a catch-up condition has occurred. That is, a following cut had caught up to a preceding cut and it must be determined whether or not the following cut was misrouted in following the first cut over the switch which was positioned in accordance with the first cut's destination, and not necessarily the second cut's destination.

The same functions are performed as each cut is identified at each succeeding switch in its travel through the yard.

There are two additional routines which are performed, although neither is related to the arrival of a cut at a switch. The wheel detectors signal the arrival of a cut at a switch, the presence detectors, P2-P8, when their status changes from indicating the presence of a cut to indicating the absence of a cut, indicate that a cut has left the switch. When this occurs, it may be possible to remove the identification of the cut from the table associated with the switch whose presence detector went unoccupied. In particular, this function can be achieved when the software file indicates that the cut has arrived. This is performed by SUN which also detects wheel detector failures inasmuch as if the indicated change occurs without the previous occurrence of a number of wheel detector actuations, it can safely be assumed that the wheel detector has failed. Furthermore, when a cut has arrived at a switch, the tables for all the previous switches in the cut's route are checked to make sure they no longer contain the identification of this cut. If they do, there may have been other presence detectors or wheel detectors which have failed and appropriate action is taken.

Another routine which is performed without relation to specific indication changes is CTC. This routine checks a number of timer functions which are provided to check for stalled cuts, blocked or fouled tracks, and provide cornering protection. This routine will also detect wheel detector failures and, when such occurrence is indicated, will cause the removal of a cut's identity from tables associated with affected switches.

Before one can understand the manner in which the entire system operates, one must understand how each of the particular routines operates and how that operation is interfaced with the operation of the remaining routines. An essential starting point for understanding of the operation of the particular routines is an understanding of the information that is stored in the computer and the manner in which that information is stored. Therefore, prior to describing any of the routines in detail, we shall first describe some of the tables which are created by the program and referred to by the program to obtain essential information.

One table that is created by the program as the system operates is called the "cut statistics" (hereinafter referred to as CUTST). The basis for the cut statistics



is found in the train consist which is stored in the machine prior to the arrival of the train. As each cut is uncoupled from the train and detected as an entity by CARDET, the information from the consist for the particular cars in the cut is stored at a particular location which is thereafter referred to as that cut. The stored data comprises the identifying designation for each of the cars in the cut, the total number of axles in the cut and other information which is available from the consist including the destination track. As the cut proceeds through the yard, entries are made into CUTST with respect to that cut such as velocities and expected times of arrival at various points in the yard. At other times, the program refers to CUTST for the particular cut in question to obtain information which has previously been stored therein. For instance, if at some point it is necessary to determine whether or not all of the axles of a cut have passed a given point, it is only necessary to compare the number of signals received from a particular wheel detector with the total number of axles that are known to be in that cut. If the number of counts from the wheel detector is less than the total number of the axles of the cut, one knows that the entire cut has not yet passed that point. On the other hand, if the total number of counts from the wheel detector is greater than the number of axles contained in that cut, then the system has determined a catch-up condition in which a following cut has caught up to a preceding cut.

Another table which is relied on by the system is entitled "destination" (hereinafter referred to as DESTB). In this table are stored entries for each destination track in the yard. A portion of a typical DESTB is represented below in Table 1. The entry for each destination track is made up of a number of words equal to the number of switches in the route from the crest to the particular destination track. Thus, referring to the typical entries in Table 1, both destination tracks 5 and 6 are made up of four words and therefore there are four switches which are traversed from the crest of the classification yard to each of destination tracks 5 and 6. The last two columns in DESTB each represent multi-bit entries. Thus, the last column "switch number" would require at least 6 bits. Five bits are required to count up to 32, an additional sixth bit is required for the negative indicator which designates the end of the entry for the particular destination track. Another column with which requires multi-bit entries is the retarder number column. This identifies the retarder which precedes the particular switch. Of course, a 0 entry in this column indicates that no retarder precedes this switch.

The "normal" column in DESTB indicates whether or not that particular switch should be reversed or normal in order to route the cut to its particular destination track. Thus, assuming an entry of 1 designates a normal switch and an entry of 0 designates a reverse switch, the route for destination track 5 would consist of, in sequence, switch 8 normal, switch 32 reverse, switch 6 normal, and switch 5 normal. The "retarder" column indicates, when a 1 is present, that the switch follows a retarder.

TABLE 1

Desti- nation	Continue	Normal	Retarder	Do Not	Retarder	Switch
Track				Throw	No.	No.
5	1	1	1	0	1	8
	0	0	0	1	0	32
	0	1	1	0	2	6
	0	1	0	0	0	5
6	1	1	1	0	1	8
	0	0	0	1	0	32
	0	1	1	0	2	6
	0	0	0	0	0	5
						-1

An explanation of the meanings of the "continue" and "do not throw" columns requires an explanation of the use of lap switches in classification yards. Where two switches are so close together that it would be unsafe to throw one while the other is occupied, the switches are considered lap switches. The one bit in the continue column for a switch indicates that the switch is the first switch of a lap switch. The one bit in the "do not throw" column indicates that the switch is the unused second switch of a lap switch. From DESTB we can determine that switch 8 is the first switch of a lap switch and, when it is in the normal position, switch 32 is the unused second half of the lap switch and it has been set to the reverse position.

The -1 in the switch number column indicates the end of an entry for a particular destination track. DESTB is one of the tables which defines the characteristics of the yard.

An additional table which is related to the characteristics of the yard is entitled "NEXTB" for next best. For various reasons, either related to a simple malfunction, or the prohibition against throwing a switch when such action is unsafe, a cut may be misrouted at any point in the route. If this occurs, it is important for the system to recognize that a cut has been misrouted. It is equally important to determine a new destination track for the particular cut so that it may be supervised in its travel through the remaining portion of the classification yard. In order to select a new destination track for a misrouted cut, the table NEXTB is provided. For each destination track, there is an entry for each switch in the route and related thereto a different destination track. Therefore, if a cut becomes misrouted, it is only necessary to determine the original destination track and the switch at which it became misrouted to determine, from the table NEXTB, the next best destination track for that cut. Other tables related to the yard characteristics define, for each switch, the distance and grade to each next possible switch in a route. Since the make-up of these tables is not vital, they are not further described here.

Through the use of these tables, one of the important objects of the present invention is achieved. The entries in these tables uniquely define the classification yard characteristics. Thus, to apply the system disclosed in the present application to a classification yard different than the one for which it was designed, it is only necessary to modify these tables in accordance with the characteristics of the yard to which the system is to be applied. The remaining portions of the system are applicable to any classification yard.



For each switch in the classification yard, there is an additional table entitled ADDSW. These tables are created as the system operates and, when the classification yard is not in operation, each of these tables is empty. Each table is capable of holding a number which designates the address of the location in CUTST where information respecting an approaching cut can be located. When the first cut is detected at the crest of the classification yard, the address of CUTST for that cut is loaded in the ADDSW for the first switch in the yard. When the first cut reaches the first switch, the address which had been in ADDSW with respect to the first switch is transmitted to the next ADDSW table for the switch next to the route to the destination track for that cut. As the cut proceeds through the classification yard, the CUTST address for that cut is transmitted from ADDSW table to ADDSW table preceding the cut. Thus an entry is made to an ADDSW table when the cut associated with that entry reaches the preceding switch. Because more than one cut may be traveling between two particular switches, each ADDSW table is capable of holding a multiplicity of entries. The sequence of the entries identifies the sequence of the cuts approaching the particular switch. It is from this address, stored in ADDSW, that the system can determine the destination track for a particular cut so that the switch may be properly positioned and the next switch in the route identified for receipt of the address identifying the location in CUTST where information respecting that cut may be found.

References to FIGS. 3A and 3B shows the subroutine CARDET. There are three real time inputs to the system related to this program. Two are received respectively from WD1a and WD1b, located at the crest. These wheel detectors are spaced approximately eight feet apart. Since the inter-axle spacing on railway trucks is approximately 5 to 5½ feet, WD1a will be actuated by the second axle before WD1b is actuated by the first. For a two axle truck the wheel detector sequence will be WD1a, WD1a, WD1b and WD1b. Furthermore, since the minimum inter-truck spacing of a railway car is 12 feet, the last axle of the first truck will actuate WD1b before the first axle of the second truck actuates WD1a. In a like manner, in a multi-car cut, the last axle of the first car will actuate WD1b before the first axle of the second car actuates WD1a. The third real time input to the system comes from CL or cut light. As is explained above, this is employed to determine the presence of a multi-car cut in combination with the signals from the wheel detector.

Before discussing the logic of CARDET in detail, attention is called to the fact that the routine sets up five registers entitled respective AXCNT, AXLES, AXLES1, AXLES2, and SAVCNT. The routine also controls the conditioning of two "flip-flops" and at different portions of the routine responds to the conditions of these "flip-flops." One is entitled A and can indicate either truck 1 or truck 2 depending on its condition; the other is entitled B and also can indicate either truck 1 or truck 2 depending on its condition.

When CARDET is called, the first decision point 20 determines whether or not any wheel detector actuations have to be serviced. Since the subroutine itself contains a loop it is possible to reach decision point 20 when no wheel detector actuations are awaiting service. In that case, the subroutine is turned off. Assuming there is a wheel detector actuation to be serviced,

decision point 21 determines if the wheel detector that has been actuated is WD1a. If it is, then a reference is made to the flip-flop A, by decision point 22, to determine whether or not it is the first truck which has actuated the WD1a. The manner in which the flip-flops A and B are set and reset to indicate truck 1 or truck 2 will be indicated as the discussion of FIG. 3A proceeds. It is only necessary to know that after the passage of a car, both A and B are set to truck 1. Assuming that it is truck 1 which has actuated WD1a, then function 23 resets the CCLD failure timer. After the passage of a car has been recorded, the CL detector is used to determine whether or not the cut is a multi-car cut or not. If CL does not detect a light signal, then the system assumes the reason is that a car coupler coupling the first and second cars of a cut has prevented the light from reaching the detector. However, the system is also designed so as to detect the failure of the light detector and uses a timer for this function. The timer is set when a car has passed the wheel detectors and no signal is detected. If the axles of a second car are detected, the timer is reset by function 23. If no second car of a cut is detected, then the system determines that the CCLD has failed.

Function 24 then increments the count in AXCNT, which had previously been set to 0. Function 25 increments the count in AXLES and decision point 26 determines if the count is legal. This is a logic function test on the operation of the system. Under proper system operation, the value stored in AXCNT will never exceed the value in AXLES by more than 2. Assuming the count is legal, the subroutine has recorded the first axle of a cut and proceeds to await further actuations to be serviced. At this time the register AXCNT has a count of 1 and so does the register AXLES, the other registers maintain their 0 counts, and the flip-flops are still set at truck 1.

The second axle of the first truck, when detected by WD1a, traverses the identical portions of the subroutine, the only difference being that at the conclusion, AXCNT has a count of 2 as does AXLES.

The next wheel detector to be actuated will be WD1b, caused by passage of the first axle of the first truck. Thus, decision point 21 will determine that WD1b has been actuated and not WD1a. This routes the routine through function 27 and the decision point 28 determines whether or not this is still truck 1. Since it is, function 29 decrements the count in AXLES and decision point 30 determines if this count is legal. Since the difference between the AXCNT count of 2 and the AXLES count of 1 is less than 2, the count is legal and decision point 31 determines if this is the last axle. The manner in which this is accomplished is merely a subtraction of the AXCNT from AXLES and since the difference is not 0, it is not the last axle and the subroutine proceeds to service further actuations. Assuming that no other wheel detector actuations remain to be serviced, decision point 20 will cause the subroutine to turn off. The next wheel detector actuation will be the second axle of the first truck actuating WD1b. As before, functions 20, 21, and 27 will bring the subroutine up to decision point 28 which will determine that this is still truck 1 and the count AXLES will be decremented by function 29. Decision point 30 determines whether or not this count is illegal. At this time, Axles has a count of 0 and AXCNT has a count of 2. Since the difference is not greater than 2, the count is not ille-



gal and decision point 31 will determine if this is the last axle. Since the count in AXLES is 0, it is clear that this is the last axle and function 32 sets A and B to truck 2 and transfers the count from AXCNT to AXLES1. Thus, AXLES1 contains the number of axles in truck 1. Function 33 then transfers the count in AXCNT to the register SAVCNT and then sets AXCNT to 0. At this point then, the register AXCNT registers 0, AXLES registers 0, AXLES1 contains a count of 2, AXLES2 contains a count of 0, and SAVCNT contains a count of 2 and both flip-flops A and B are set to truck number 2. From this point the subroutine proceeds to a portion entitled FORW which is shown in FIG. 3B.

In FIG. 3B function 34 obtains the cut address in CUTST for this particular cut. Decision point 35 determines if the first ADDSW table is clear. This is the table for the first switch in which will be stored the addresses in CUTST related to cuts approaching the switch. Assuming this is the first car over the crest, it will be clear and function 36 clears the multi-car cut flag. This is a software flag indicating whether or not the particular cut is a multiple-car cut. Since this flag is not reset after a multi-car cut has been detected, it is cleared at this point, after the first axle of the next cut has been detected. Function 37 enters the address of the portion of CUTST pertaining to this cut into the ADDSW table for the first switch in the yard. Functions 38, 39 and 40 store various car parameters, particularly the axle count for this car and the total axles for the cut in the CUTST table, as well as other parameters. In addition to the car sensing devices already mentioned there is normally a weighing device and bulk-measuring device at or near the crest which records the various other parameters with respect to the car. Function 40 stores these parameters in the portion of CUTST related to this cut. The program then returns and awaits further wheel detector actuations.

The next wheel detector actuation will be WD1a responding to the first axle on the second truck and as before decision points 20, 21 and 22 will be traversed. However, decision point 22 will determine that this is not truck 1 and therefore function 41 will increment the count in AXLES 2. Decision point 42 will determine if this count is illegal, and assuming it is not, function 43 will decrement the count in AXLES1. At this point AXCNT will contain a count of 0, AXLES will contain a count of 0, AXLES1 will contain a count of 1 since its previous count of 2 has been decremented by function 43. AXLES2 will contain a count of 1, having been incremented by function 41. SAVCNT will retain its count of 2. Decision point 44 determines if this is the end of truck 2. This is simply a comparison of the count in AXLES1 with AXLES2, since the routine assumes that the trucks in a car will be symmetrical. Since the counts are not equivalent, this is not the end of truck 2 and the program returns to await further actuations to be serviced.

The next wheel detector actuation will again be WD1a actuated by the second axle in the second truck. The same routine as that encountered for the first axle of truck 2 will be traversed, and after completing function 43, the count in AXCNT will still be 0, as will the count in AXLES. The count in AXLES1 will be 0, its previous count of 1 having been decremented by function 43. The count in AXLES2 will be 2 since its previous count of 1 was incremented by function 41. Deci-

sion point 44 then determines whether or not this is the end of truck 2. This merely requires reference to AXLES1 which will contain a count of 0 if it is the end of truck 2. Assuming it is, the function 45 sets WD1a back to track 1 ready to respond to the next car's actuation of WD1a.

The next wheel detector actuation will be WD1b in response to the first axle of the second truck. As before, decision points 20 and 21 will be traversed as will function 27, and at decision point 28, reference to "B" will determine that this is not truck 1, and therefore function 46 will be performed to decrement the count in AXLES2. Decision point 47 determines if this count is illegal, and assuming it is not, decision point 48 determines if this is the last axle. This merely requires reference to AXLES2 which contains a non-zero count indicating that this is not the last axle. The program then returns to await further wheel detector signals. The next wheel detector signal will be WD1b actuated this time by the second axle of the second truck. Decision points 20 and 21 will direct the program to function 27 and thence to decision point 28 which will again determine that this is not truck 1 by reference to B. Function 46 will then again decrement AXLES2 which will then arrive at a count of 0. Decision point 47 will determine that this count is not illegal and decision point 48 will determine that this is indeed the last axle. This will direct the program to function 49 to set B back to truck 1 to being counting axles of the next car. Decision point 50 then determines whether the detector CCLD is occupied. If this is a single car cut, then the detector will not be occupied directing the program to function 51 to clear the multi-car cut flag. Subsequently, point 52 will determine if all the counters are registering 0, as they should be at the end of a cut. If they are, then program returns.

Assuming, however, that the detector CCLD is occupied the function 53 will set the multi-car cut flag and start the CCLD failure timer in operation. This is the timer that was referred to with respect to function 23. Subsequently, decision point 54 determines whether or not all counters but AXLES and AXCNT are equal to 0. Assuming this is a multi-car cut, AXLES1 and AXLES2 should be set to 0, assuming no failures. The program will then return. If this is a single car cut and all counters are not 0, or if it is a multi-car cut and one or more of the counters other than AXLES and AXCNT are not equal to 0, function 55 will force this to look like a 4-axle cut by removing the previously stored number of axles from CUTST and substituting four in its place. This indicates a wheel detector failure and so the routine is directed to a failure routine.

The only remaining portion of CARDET to be explained is the failure mode. The most common cause of a failure in this portion of the routine would be a wheel detector which failed to signal a count. The decision points 26, 42, 47, and 30 directly monitor the wheel detector counts by monitoring the value in the registers and testing for an illegal combination. Decision points 52 and 54 also check the same registers under specified conditions to also detect a failure. If a failure is detected, the program is directed to the failure subroutine shown in FIG. 3C. Decision point 56 determines whether or not an entry has been made. If the failure occurred prior to the last axle of the first cut actuating WD1b, then the program never would have performed FORW (FIG. 3B). In this case an entry would not have



been made and decision point 56 would direct the program to this subroutine to make the entry so as not to lose the particular cut. On the other hand, if the failure is detected subsequent to the time an entry has been made in ADDSW by the FORW routine, decision point 56 will immediately thereafter perform function 57. This will flag the occurrence of a crest wheel detector failure and then via function 58 will stop the humping operation, and after a predetermined delay caused by function 59 will reset the system parameters in function 60 and then proceed to service further actuations. The routine POSSW is shown in FIG. 4. The purpose of this subroutine is to throw a switch for an approaching truck when it is safe to do so. This is called when an entry is made to the ADDSW table for the switch.

When entered, decision point 61 determines whether or not the switch is occupied. As shown in FIG. 1 each switch has a wheel detector preceding it and a presence detector protecting the area just subsequent to the switch. When a wheel detector is actuated, the switch is considered occupied until the presence detector indicates that the area defined by the presence detector is free of any cars. In this pass through POSSW, since it has been called by CARDET, the switch to which decision point 61 has reference is the first switch in the yard. Assuming the switch is not occupied, function 62 now obtains the destination track. The previous routine FORW (FIG. 3B) entered the cut address into the ADDSW table for the first switch. By referring to the portion of CUTST to which that address refers the destination track for the cut is available. Function 63 then finds the entry in DESTB corresponding to that switch. As shown in Table 1, assuming a destination track was 5, the second bit in the first word would be the entry to which function 63 is directed. Function 64 outputs the selected bit to throw the switch to correct position. Subsequently, decision point 65 determines if continue is set. Reference to Table 1 for destination track 5 shows that the continue bit, the first bit in the first word, is set. As was explained with reference to Table 1, this continue bit, when set, indicates the switch is the first switch of a lap switch. If continue is set, function 66 looks at the next switch, and this has reference to merely looking at the next word in the destination table for that particular destination track. Decision point 67 determines if do not throw is set. As shown in Table 1, for destination track 5 do not throw is set in the second word and therefore the program would proceed to return. If do not throw is not set, then decision point 67 refers the program back to function 64 to throw that switch as well. Since this is a lap switch, the two switches are thrown at the same time. For the same reason, there is no need to determine if the switch is occupied since we have determined at decision point 61 that the first switch was not occupied and therefore the second would not be. Outputting that instruction would refer the program then to decision point 65 to determine if continue was set. Reference to Table 1 for destination track 5 showed that continue was not set and therefore the program would return.

Assuming that the switch was occupied, decision point 68 would determine if ADDSW is clear. Since the ADDSW would only be clear if a switch is unoccupied and no cuts are approaching, if decision point 68 determines the ADDSW is clear, function 69 flags this as a presence detector fail. This will occur if a presence detector fails to indicate, after a wheel detector actuation,

that the cut has cleared the switch. If the ADDSW table is not clear as determined by decision point 68, then the program returns and will attempt to perform POSSW at some further time in order to correctly position the switch.

As shown in FIG. 2A after POSSW has performed, the next routine to be performed is RETOT. This merely transfers data with respect to the cut to the automatic retarder control portion of the program to properly control the hump retarder RH. Since that program forms no portion of the instant invention, no further mention will be made of RETOT.

If a second cut reaches the crest before the preceding cut reaches the WD2, then the same routine will be performed for this second cut which will result in a second entry in the ADDSW table for the first switch.

Upon the arrival of the first cut at WD2, the subroutine SWWD will be called, as shown in FIG. 2A. SWWD is shown in FIG. 5.

The purpose of this subroutine is to check for lost cars, to set the axle count of the cut arriving at a particular switch and to protect the switch from being thrown.

By the time WD2 or any other wheel detector preceding a particular switch is actuated by a cut, that switch should be in position to route the cut successfully to its destination track. If it is not, then the cut is misrouted because the switch will not be thrown after a wheel detector actuation until the presence detector, subsequent to the switch, indicates that the cut has cleared that switch location. The first decision point 70 in SWWD determines if there are any wheel detector actuations, and if there are not, the decision point 260 determines if LOST is on. If it is, or if decision point 261 determines that LOST entries have been made, then LOST (FIG. 5J) is performed. After LOST is performed decision point 262 determines if LOST is done, if not SWWD is reentered. If LOST is done, or if LOST is not on and there are no entries in LOST, the routine is completed. The purpose of LOST will become clear as the description proceeds. If there are the wheel detector actuation flag is reset by function 71 so it will not be serviced twice and function 72 determines if the ADDSW table for that particular switch is clear. If it is, then the program LOST is entered. The appearance of a cut at a switch when the ADDSW table for the switch is clear means that the cut is no longer identified with a particular portion of the CUTST table and is in effect lost. The subroutine LOST, shown in FIG. 5J, handles this occurrence.

Assuming that the ADDSW table is not clear, then function 73 adds a count to a temporary axle count for that cut arriving at that switch. Since the CUTST table contains information with respect to the number of axles in a cut, this temporary axle count table can determine when the entire cut has arrived.

Function 74 then determines, from the temporary axle count table, whether or not this is the first wheel detector actuation for this cut. Assuming it is, function 75 resets the position lights to indicate on a control panel the position of the cut, and sets a software switch protection to disable this switch from being thrown, and then returns the program to decision point 70 to await further wheel detector actuations.

On the second wheel detector actuation, the same routine would be followed except that decision point 74 would determine that this is not the first axle and de-



cision point 77 would determine that this is the second axle. Thereafter, function 78 would set the loop failure detector, which is a timer. A predetermined time after the second axle passes the wheel detector, if the loop presence detector fails to detect the cut, then it is signaled as a loop failure. Subsequent to function 78, subroutine SWOCC is performed. This subroutine is shown in FIG. 5B and will be discussed later. Generally, this subroutine determines if the track destination is achievable and safe, and, if it is, puts in a call for throwing the next switch in the route. The program then returns to decision point 70 and awaits further wheel detector actuations.

Subsequent wheel detector actuations will repeat the foregoing routine up to decision point 77. This will determine that it is not the second wheel detector actuation and decision point 79 will determine whether or not this is the next-to-last wheel detector actuation to be expected. If it is not, decision point 80 will determine if the count is negative, that is, if more axles had been counted than had been expected. If the count is not negative, then the program returns and waits for further wheel detector actuations.

On the next-to-last wheel detector actuation, decision point 79 will call the subroutine CP for corner prevention. This subroutine is shown in FIG. 5F, 5G and 5H. After the completion of that routine, function 81 sets the cut arrived indicator and returns to await further wheel detector actuations. When the next-to-last axle has arrived, the system considers the entire cut has arrived and reacts accordingly. The next wheel detector actuation will run through a similar routine except that at decision point 79 the actuation will no longer be the next-to-last and therefore decision point 80 will determine whether or not the count is negative, which it will not be and therefore the routine will loop back and wait for further wheel detector actuations. The remaining portions of SWWD react to wheel detector actuations over and above the number of axles expected, and will be discussed later.

Before discussing the remaining portions of SWWD, it will be helpful to first discuss the routine SWOCC which is entered on the second wheel detector actuation from function 78 and also the routine CP which is entered on the next-to-last axle from decision point 79.

The routine SWOCC determines if the destination track assigned to the cut is achievable and safe and then enters a call to the next switch. This routine is shown in FIG. 5B. When entered, the first order of business is a further routine DESEN. This routine is shown in FIG. 5C and its function is merely to determine what the next switch is. In FIG. 5C the first function 83 is to determine the destination track for that cut. The ADDSW table will refer to the address in CUTST in which is located the information required by function 83. Function 84 obtains the switch number at which the cut has actuated the wheel detector for the second time. It will be recalled, as shown in FIG. 5A, that the second wheel detector actuation at a particular switch, determined at decision point 77 called SWOCC, shown in FIG. 5B, which in turn called DESEN. Decision point 85 determines if this switch appears in DESTB for the particular destination track to which the cut should be routed. If it is not, then function 86 obtains the next best destination track for this cut at the particular switch. Whether or not a reference to NEXTB is necessary, decision point 85 will register

that the switch number is in DESTB for either the original destination track or the one obtained from NEXTB. Decision point 87 then determines whether or not the particular switch is a lap switch. Of course, this can be determined from DESTB by checking the "continue" bit in the table. If it is not a lap switch, then function 88 obtains the next switch number in DESTB for this particular route. If it is a lap switch, then function 89 obtains switch NEXT plus 1, also from DESTB. At that point the routine terminates and returns to SWOCC (shown in FIG. 5B).

Decision point 90 determines whether or not the switch at which the wheel detector actuation was recognized is out of correspondence. There are three possible switch indications, normal, reverse, or neither of the two. If the switch is in the latter category, it is recognized as out of correspondence and function 91 flags this and returns to SWWD (FIG. 5A).

Assuming, however, that the switch is in correspondence, then decision point 92 determines if we have a misroute. Whether or not a misroute has occurred can be determined by comparing the DESTB entry for this switch with the switch indication.

Assuming there is no misroute, decision point 93 determines if continue is set, and, assuming it is not set, that is, the particular switch is not the first switch of a lap switch, decision point 94 determines if this is the last switch in the route. This can be determined from reference to DESTB which would have a -1 entry, indicating the end of an entry, subsequent to the last switch in the route. Assuming it is the last switch in the route, functions 95 and 96 compute the time required for the cut to clear the switch and sets up protective devices to prevent fouling if the cut does not clear the switch. In the case for the last switch in a route, this concludes SWOCC.

However, if this is not the last switch in the route, then decision point 95 calls the subroutine DEST, shown in FIG. 5D.

The function of DEST is to see if the destination for the cut is fouled or blocked, and if it is, then DEST determines an alternate route for the car.

Referring to FIG. 5D, the first function 97 in DEST is to obtain the destination track of the cut. As before, a reference to ADDSW will disclose the address in CUTST for this cut. Reference to CUTST will determine the destination track. The next function, 98, determines the adjacent track from reference to a yard characteristic table. DEST will check both the destination track and the adjacent track to determine whether either are fouled or blocked. It is necessary to check the adjacent track to determine whether either are fouled or blocked. It is necessary to check the adjacent track because if it is fouled, it may prevent the cut from reaching the destination track. Decision point 99 determines whether or not the next switch is the last switch, by reference to DESTB. If it is, decision point 100 determines if the track is fouled and if it is, function 101 flags this situation as a misroute due to fouling and advances to decision point 102 to determine if the adjacent track, selected at function 98, is blocked.

If the next switch is not the last, there is no necessity to check for a fouled track and decision point 103 is subsequently checked to determine if the track is blocked. If it is not, then DEST has determined that the destination track is neither blocked nor fouled and it has completed its function.



Assuming, however, that the track is blocked, then function 104 is performed subsequent to this determination to flag this track as blocked. Then, decision point 102 determines whether the adjacent track is blocked. If it is not, then decision point 105 determines if the adjacent track is fouled and if it is not, function 106 determines a new destination track for the cut as the adjacent track which had been determined at function 98.

If the adjacent track is blocked then function 107 would flag it as blocked. If the adjacent track is fouled, then function 108 would flag it as fouled. In either case, the next function, 109, determines the tracks which are available for the cut from that point, eliminating the original destination track and its adjacent track, since the program will only reach this point if both these tracks are unavailable. These tracks are sequentially checked by decision points 110 and 111 until one is determined which is available and then function 106 selects the track as available. If all tracks are blocked, then one of them is selected by function 106 since to properly route a cut, a destination must be available. In any event, the conclusion of DEST is a selection of a destination track, the original one if possible, and adjacent one if the original one is blocked or fouled, or some other available track, or if none are available, one of the blocked tracks is selected.

This destination track is then stored, by function 112, in CUTST for the particular cut. Subsequently, function 113 enters the cut address in the ADDSW table for the next switch, which had been determined above from DESEN.

At that point subroutine POSSW is entered (shown in FIG. 4) to set up to throw the switch.

After completing POSSW the next routine entered is CALARV. The function of this subroutine is to calculate a time of arrival of the cut at the next switch. If the cut does not arrive within the computed time, it is assumed to be stalled and appropriate protective action is taken.

This routine is shown in FIG. 5E and begins with function 205 which stores the time between wheel detector actuations for a velocity calculation. Function 206 obtains from CUTST the previously calculated  $V_{next\ forward}$  for this cut and employs this as the velocity figure in the calculation. The manner in which  $V_{next\ forward}$  is computed and stored will become clear later. Function 207 obtains the subsequent grade and distance between the switch at which the cut was detected and the next switch in the route for that cut. Knowing these factors, COMPS is entered to actually perform the computations. COMPS is discussed with reference to FIG. 5I and will not be discussed further here. It is only necessary to recognize that the routine produces a velocity figure which is the projected velocity of the cut at the next switch in the route and also produces a time figure which is the time required by the cut to reach the next switch in the route.

Function 208 stores the just computed projected velocity  $V_f$  as  $V_{next\ forward}$  in CUTST with respect to this cut. Function 209 stores the computed time for the cut to reach the next switch in a stall timer. We will discuss subsequently the CTC routine which detects expired timers and flags these as stalled cuts. This is the completion of the CALARV routine.

At the conclusion of CALARV, decision point 114 determines if a retarder is next, so that if it is, a transfer

can be made to the retarder operating system of the current cut parameters. If no retarder is next, or after the transfer is made, the routine is completed and returns to SWWD. The portions of SWOCC which have not yet been discussed are pertinent to the repeat of SWOCC subsequent to function 381 in SWWD. Therefore, we will postpone discussion of the remaining functions until that time.

Thus far, the presence of an axle detection at a switch calls SWWD. Each axle actuation causes a different function in SWWD to be performed. The first sets switch software protection to disable the switch from being thrown, the second causes SWOCC to be traversed to check switch occupancy and to determine if the original destination track is achievable and safe and puts in a call to the next switch and calculates an arrival time at that switch. When the next-to-the-last axle of the cut is detected, at decision point 79, then the subroutine CP is entered. CP is the cornering prevention routine and is shown in FIG. 5F, 5G and 5H.

Before discussing the cornering prevention routine in detail, it will be worthwhile to discuss what cornering prevention is and the manner in which the routine provides cornering prevention.

If the separation between two cuts which traverse the same switch in two different positions is not great enough, the front of the first car in the second cut may come into contact with the rear of the last car in the first cut. Although railroad cars are designed to withstand impacts, the design philosophy is based on the assumption that the impact will occur at the coupler and coupling mechanism and at no other location. For this reason the coupler and coupling mechanism of railroad cars are designed to withstand relatively severe impacts. The remaining portions of a railroad car are not so designed and can be severely damaged by impacts or contacts with stationary objects or other cars. If the separation between the two cuts discussed above is not great enough to avoid impact, it should be readily apparent that the impact will arise due to the corner of one car impacting the corner of another car; just the location where railroad cars are not designed to withstand impacts. Thus, this situation is called cornering and is a situation which is to be avoided. For each switch a clearance distance is determined as the distance the rear of a cut must be beyond the switch points to clear another cut on an adjacent track. For railroad cars which are 10 feet in width, a reasonable clearance distance may be on the order of 100 feet.

In discussing the CP routine we will refer to a primary switch and also to cut A and cut B. Reference to FIG. 5A shows that when the next to the last axle of a cut actuates any switch wheel detector, the routine CP is called. The cut which has actuated this wheel detector is cut A. The switch at which the wheel detector is located is the primary switch. In order to prevent cornering, the subroutine CP in effect looks back at the yard to cuts approaching the primary switch. When it finds such a cut, which is called cut B, it will compute the time it will take for cut A to reach the clearance point and it will also compute the time it will take for cut B to travel from the primary switch to the clearance point. The routine then locks the switch into the position that it is in as cut A traverses the switch for a predetermined period of time. This period of time ensures that if cut B reaches the switch before the expiration of this period, it will be routed to follow cut A and thus



prevent cornering since it will then be on the same track. At the expiration of this period of time, the separation between cuts A and B are sufficient so that cut A will reach the clearance point before cut B can reach the clearance point thus ensuring cornering prevention.

The first function 115 computes the measured velocity for what is cut A at the primary switch. This computation is based upon the time elapsed between a pair of wheel detector actuations along with knowledge of the wheel base of the car. All this information is available is CUTST. Decision point 116 determines whether or not the velocity computed at function 115 is within allowable limits. Since it is always possible for a wheel detector to fail to count an axle, this ensures that the velocity computed at function 115 is a realistic velocity. A wheel detector failure will result in an unrealistic velocity computation. If the velocity computed is not within the proper limits, then function 117 obtains the quantity  $V_{next\ rear}$  from the table CUTST. The manner in which this quantity is inserted in CUTST will become clear as this description proceeds. If the computed velocity is within the proper limits, or after  $V_{next\ rear}$  is obtained from CUTST, function 118 gets the destination of cut A, again from the CUTST table. Then the routine DESEN is performed. This is shown in FIG. 5C and has been explained above. Its function is merely to obtain from DESTB the number of the next switch in the route for the cut. Subsequently, function 119 stores the DESTB address and the entry for cut A. This has reference to the number of the next switch in the route for this cut as determined by DESEN. At that point decision point 120 determines whether or not this is the last switch in the route for cut A. Assuming it is not, function 121 determines the distance to the next switch. From the classification yard characteristics, which are stored in a table, this distance between switches is readily available. Function 122 flags this particular computation as not a clearance pass. The reason for this will become clear as the description proceeds. Function 123 then obtains the proper velocity for cut A. This is either the velocity computed at function 115, if it is within the limits as determined by decision point 116, or else it is the velocity determined by function 117. At this point, the routine COMPS is performed. This routine computes the velocity of cut A at the next switch in the route and also computes the time it will take cut A to reach the next switch in the route.

COMPS is shown in FIG. 5I. As shown there the first function 124 is to obtain the acceleration for cut A. This is obtained by the information available to the automatic retarder control program. The cut acceleration is essentially its rolling resistance. The computation of this parameter is referred to in the copending application referred to earlier. Function 125 then determines the grade at the primary switch. As was the case with the distance between switches, the grade of the track at each switch is located in a table characteristic of the classification yard. Reference to the table, at function 125, determines the grade at the switch and adds this to the cut acceleration. Function 126 then computes the velocity of cut A at the next switch in the route. This is computed by utilizing the formula  $V_f = \sqrt{V^2 + 2AS}$  where  $V^2$  is the velocity of cut A at the primary switch,  $A$  is equal to the sum of the cut acceleration and the additional acceleration which may be added due to the grade of the track, and  $S$  is the distance between switches. Function 127 then computes the time it will

take cut A to reach the next switch in the route from the formula  $T = V_f - V/A$ , where  $V_f$  is the velocity previously computed at function 126,  $V$  is the present velocity which is either the velocity computed at function 115 or that selected by function 117 and  $A$  is the composite of the present cut acceleration and whatever additional acceleration may be added by the grade of the track. Decision point 128 then determines whether or not this is a clearance pass. Since function 122 set this as not a clearance pass, then the program will proceed to function 129 to determine whether or not a retarder is next, that is, whether a retarder appears between the primary switch and the next switch in the route. If one does not, the program continues with the CP routine shown in FIG. 5F. Assuming, however, that there is a retarder in the path, the previously computed values will have to be adjusted. The computations carried out at functions 126 and 127 were based on the assumption that no retarder was present. The presence of a retarder will obviously affect the velocity of the cut at the next switch, since the velocity of the cut will be materially reduced through the retarder, and it will also affect the time it will take the cut to reach the next switch as the time it takes a cut to traverse a retarder is materially longer than the time it would take the cut to traverse the same distance if not occupied by a retarder. Therefore, assuming a retarder is next, function 130 determines if that retarder is the master retarder in which case function 131 would determine the estimated exit velocity from the master retarder, add to it a predetermined quantity delta and use this as  $V_f$  instead of the value computed at function 126. If decision point 130 determined this was not the master retarder which was next, then function 132 would use the estimated exit velocity from the group retarder plus the predetermined quantity delta as  $V_f$ . The quantity delta is an empirical value based upon the change in velocity of a cut from the exit of the retarder to the next switch in the route. At this point function 133 would add to the previously computed time another empirical quantity which is the average time it takes a cut to traverse a retarder. This will conclude COMPS and bring us back to CP (FIG. 5F).

Function 134 then stores the final velocity of cut A, as determined in COMPS, and function 135 stores in CUTST, for the particular cut, the same velocity as  $V_{next\ rear}$ . Subsequent to function 135, function 141 clears the retarder number and pass counter. In the routine subsequent to function 141, the program will keep track of the number of passes it makes and also will keep track of the retarders it examines. The necessity for this will become clear as the description proceeds. However, at decision point 120 we had assumed that this was not the last switch in the route and thus discussed functions 121 through 135. However, if this is the last switch in the route, then functions 136 through 140 will be performed instead.

Function 136 sets DR and CTC as ineligible. DR is an automatic retarder control function which is related to distance to coupling measurements. This routine is initiated when the cut is at the last switch and this is adjacent to the measurement point. At the time the cut is rolling over the measurement point, accurate measurements cannot be made and therefore measurement at this time is prohibited. For the same reason, as the cut enters the clearance circuit, the presence detector associated therewith may exhibit intermittent operation



and setting CTC ineligible ensures that this operation will not be interpreted as a presence detector failure. Basically this is an internal record keeping function. Function 137 obtains the distance to the end of the clearance track circuit for the last switch. The routine COMPS, which we discussed with reference to FIG. 5I requires a number of items of information such as the distance from one switch to the next and the grade between two switches. However, when the primary switch is the last switch in the route, such information will not be available. Therefore, function 138 creates dummy information to simulate these items of data. The information is provided on the basis of the length of the clearance track circuit and the grade of the track in the clearance track circuit. Function 139 then duplicates function 123 in selecting a previously calculated velocity for cut A and then routes the routine through COMPS. Since we have already discussed COMPS with reference to FIG. 5I no necessity is seen for a repetition of that discussion. Basically, the important quantity computed by COMPS during this pass will be the time computed by function 127. This time will be stored, by function 140, as the time in the CTC timer. The function of the CTC timer will become clear with a description of the subroutine CTC. In any event, function 141 is then performed.

Function 142 stores the primary switch number in a specified location and decision point 143 determines if this is the only cut at this switch. This will require reference to the ADDSW table for the particular primary switch. It will be recalled that the second wheel detector actuation at any switch calls the subroutine SWOCC and that this routine will cause the CUTST address to be entered into the next switch in the route for that cut. Therefore, if a cut approaching the primary switch has had at least two wheel detector actuations at the previous switch, then the primary switch ADDSW table will contain the CUTST address for that cut. Therefore, function 143 will determine that this is not the only cut at the switch and direct the routine directly to spot 1, SP1 (FIG. 5G). If cut A is the only cut at the switch, that is, if the ADDSW table contains no other entry for that switch, then the program is directed to decision point 144. The portions of the routine between decision point 143 and SP1 search for the next subsequent cut approaching this switch in order to ensure cornering prevention. The reason that decision point 143 can direct the routine to SP1 if another cut is at the switch is that it is no longer necessary to search for the directly subsequent cut as its identity is already known.

Beginning at SP1 (in FIG. 5G), function 144 obtains the velocity  $V_{next\ front}$  for the calculations to be carried on with respect to cut B. It should be recalled that function 144 directly follows decision point 143 which determined that there was another cut at the switch. It is the second cut in the ADDSW table, which is hereinafter referred to as cut B. Since the cut address is located in the ADDSW table for the primary switch, this velocity value is known to be stored in the CUTST table for this particular cut. Since the cut address is in the ADDSW table for this primary switch, routine SWOCC (shown in FIG. 5B) has been completed when cut B was detected at the preceding switch. The cut address is only entered into the ADDSW table for the next switch by function 113 at that time and directly following that function is the subroutine CALARV which

computes this specific velocity value  $V_{next\ front}$  and places it in the CUTST table for cut B. Therefore, function 144 can obtain this velocity value.

Decision point 145 then determines if the primary switch is a lap switch, and this is determined from the DESTB table. Assuming it is not, then function 146 uses a distance value which is associated with the primary switch. This distance value is the length of the clearance track circuit associated with the primary switch. Subsequently, function 147 obtains from the yard characteristics table the grade value associated with the primary switch. Decision point 148 then determines if this is the last switch in the route. If it is, function 149 obtains a dummy word to simulate a subsequent switch. The reason for this is that COMPS operates upon distance between switches. Since we have identified the last switch in the route, there is no further switch from which to obtain a distance measurement. However, the distance used is the length of the clearance track circuit and therefore function 149 uses this distance to simulate a subsequent switch. Function 150 then obtains the velocity of cut A, which has previously been computed. Function 151 then flags the particular computation as a clearance computation. Reference to FIG. 5I, which shows the routine COMPS, will show the reason for this. This routine then computes a time value, at function 127, and then decision point 128 terminates the routine COMPS since this is a clearance pass.

Function 152, subsequent to the completion of COMPS, stores the time value computed by COMPS, as time 1 with reference to cut A. This is the time it will take cut A to travel from its present position, at the switch, to the clearance point associated with the switch. Function 153 then obtains the velocity of cut B, which in this example had been determined by function 144. Again, COMPS is performed to compute a time 2, associated with cut B. This is the time it would take cut B to travel from the switch to the clearance point. Function 154 stores this time 2. Function 155 then adds an empirical safety allowance to time 1. This, in effect, makes cut A appear to be a slower traveling cut than it actually is and provides a safety factor for the remaining portions of the computation. Decision point 156 determines if time 1 is greater than time 2. That is, will it take cut A a longer time to reach the clearance point than it will take cut B? If it is not, that is, if cut A is a faster traveling cut than cut B, then function 157 is performed. Function 157 merely obtains an empirical time value which is a minimum time during which the switch is protected from being thrown to ensure that the entire cut can get through the switch. In effect, decision point 156 has determined that there is no cornering potential, with respect to cut A and cut B at this particular switch and therefore no special precautions need be taken. However, if time 1 is greater than time 2, then decision point 158 compares the difference between time 1 and time 2 with the minimum empirical time. If the difference is greater than the minimum, then the difference time is used and stored, by function 159, as a beginning value in the cornering prevention timer.

An example will suffice to illustrate the manner in which the CP timer prevents cornering from occurring. Assume that the time determined for cut A to reach the clearance point is 20 seconds and the time determined for cut B to reach the clearance point is 10 seconds.



Therefore, if cut B were directly behind cut A, it would reach the clearance point some 10 seconds before cut A and surely prior to the expiration of the next 10 seconds, the two cuts would collide. However, the program would store the value of 10 seconds, the difference between time 1 and time 2 in the CP timer. Therefore, for 10 seconds after cut A had crossed the switch, the switch could not be thrown. If cut B appeared at the switch in that 10-second period of time, it would be misrouted and follow cut A. This would prevent cornering since the two cuts would be on the same track. At the expiration of that 10-second period of time, cut A has in effect a 10-second lead on cut B and both cuts would reach the clearance point at the same time, that is the rear of cut A would reach the clearance point at the same time as the front of cut B. The safety value added to time 1 would ensure no cornering.

Thus far, the CP routine has been discussed with respect to the case where cut B's identity is found by reference to the ADDSW table at the primary switch. However, this is not necessarily always possible and the program provides the capability to search the cuts behind cut A for a potential cut B, that is, the next cut to arrive at the primary switch.

The functions 160 through 177 form the logic which searches back through the classification yard for the next succeeding cut directed to the primary switch. If such a cut is found, or if the search proceeds clear up to the first switch and no such cut is found, this loop terminates. There is, however, an additional manner in which this loop can terminate which may be used, as follows. For each path through the loop, one preceding switch may be examined for the presence of a cut. If there are a number of preceding switches, between the primary switch and the first switch in the yard, this continual looping process may consume an amount of time which is not warranted. For instance, if a cut is finally found which is directed to the primary switch but which is some six switches back in the yard, there is very little likelihood that this cut can present a cornering problem. Therefore, the routine makes provision for a pass counter which is incremented each pass through this particular loop. When the pass counter reaches a predetermined number of passes, then this loop will also be terminated. With the function of this portion of the routine now explained, we will see how this is accomplished.

Functions 160 through 162 determine if there is a retarder immediately preceding the primary switch and if so, this information is retained because it will be useful in a later loop in the routine. Decision point 163 determines if this is the final pass. This requires a comparison between the pass counter and the arbitrary number of passes which is selected as the maximum for this loop. If it is, then the routine skips to point C (shown in FIG. 5G). If not, decision point 164 determines if the switch presently being looked at is the first switch. Of course, since function 142 set the current switch number to be equal to the primary switch, this will usually not be a first switch in a yard and therefore generally the routine will continue to N (FIG. 5H). If a number of switches are looked at, each one on a different pass through this loop, and we eventually reach the first switch in the yard without finding a potential cut B, then this decision point will direct the routine to point C (FIG. 5G) terminating this loop.

Assuming, however, that this is not a final pass, and the switch number in the current switch storage location is not the first switch in the yard, then function 165 will decrement DESTB. It should be recalled that the DESTB table includes in sequential order each of the switches for the route to a particular destination track. Since there is only one route to reach any particular switch or destination track in the yard, it is only necessary to search back through the switches located in DESTB to determine whether or not a cut is present at any one of these switches in order to locate the cut succeeding cut A which is routed to the primary switch. Therefore, function 165 decrements DESTB and locates the switch preceding the primary switch in the route. Function 166 obtains this switch number and saves it and decision point 167 determines if this is the first switch. If it is, the routine jumps to decision point 172. Functions 168, 169, 170, and 171 test the switch number to determine if it is a lap switch. If it is, the switch determined at function 166 would be the second switch in the lap switch and since all lap switches are referenced by their first switch number, we must decrement DESTB once more to obtain the first switch of the lap switch. Therefore, if decision point 167 determines the current switch is not the first switch, function 168 decrements the DESTB once again and decision point 169 determines if this is the first switch in the lap switch and if it is, function 171 uses this switch number. However, if it is not, then function 170 increments DESTB again to restore the switch number determined at 166, and this switch number is used. Decision point 172 determines if this switch follows a retarder. If it does, then this information will be necessary at a later point in the program. If this switch does follow a retarder decision point 173 determines if the retarder has already been identified. If it has not, functions 174 and 175 obtain the retarder number and store the retarder and switch number for later use.

If the switch does not follow a retarder or if the retarder has already been found, or once the retarder number and switch information have been stored, then decision point 176 determines if the ADDSW table for that switch is clear. If it is, that means no cut is approaching that switch and function 177 increments the pass counter, for we now have completed one pass in the loop and reenters the loop at point O (FIG. 5F). This looping is continued each time the loop is traversed the particular switch that is looked at is the one preceding the switch looked at on the prior loop. This looping process will terminate if the number of passes reaches the arbitrary limit, determined at this decision point 163 or if we reach the first switch in the yard without finding an ADDSW table entry, determined at decision point 164. The only other exit from this loop is if decision point 176 determines that an ADDSW table is not clear at which point we jump to spot 2, SP2, in the routine (FIG. 5H).

The only time SP2 will be reached is when a preceding cut B has been identified at some preceding switch in the yard. Since an ADDSW table entry has been found for cut B, the CUTST table will contain, for cut B, its projected velocity at the switch at which it is found. However, there is no assurance that the switch for which the velocity computation has been made is the primary switch under consideration. Therefore, the functions 178 through 197 project the velocity of cut B forward from switch-to-switch until we obtain the



projected velocity for cut B at the primary switch, which is the velocity we will need in order to make the computation shown in FIG. 5G, which has been discussed above.

Function 178 stores the cut address for the now located cut B and decision point 179 determines whether or not the axle count for the switch at which we have located cut B is greater than 2. If it is not, we cannot tell if cut B will be routed to the primary switch or not and for safety purposes assume it will. If the axle count is greater than 2, then decision point 180 checks if the switch is out of correspondence, that is, if the switch indication is not the same as the switch call. If the axle count is greater than 2, and the switch is in correspondence, then we skip to point C in the program and use the minimum time, in response to function 157 for the CP timer. This logic determines that the identified cut B will not arrive at the primary switch. Since the axle count is greater than 2, had the cut been directed to the primary switch, its address would have been found in an ADDSW table associated with a switch in DESTB at which the axle count was less than 2. Since this was not the case, and the switch is in correspondence, that it is lying in the position to which it is called, we can be assured that the cut B will not be routed to the primary switch. However, if the axle count is less than 2 or the switch is out of correspondence, then it is possible for cut B to be routed to the primary switch and therefore the computations that follow are necessary for cornering protection.

The first decision point, 181, determines if a retarder was found between the primary switch and the switch at which cut B was located. If no retarder was found, then the computations can proceed by computing the theoretical velocity of the cut from switch to switch taking into account the grade of the yard. However, if a retarder was found between the primary switch and the switch at which cut B was located, then reference must be made to the retarder exit velocity in order to obtain an accurate calculated velocity for cut B at the primary switch.

Assuming, however, that a retarder was not located, then function 188 obtains the  $V_{next\ front}$  from the CUTST. Function 189 assigns to cut B this velocity and decision point 190 determines if this switch is a first of a lap switch. If it is, then function 192 increments the DESTB twice so as to skip over the second switch of the lap switch. On the other hand, if the previous switch is not the first of the lap switch, then function 191 increments DESTB once. Decision point 193 determines if this is the end of the DESTB table. An affirmative answer indicates an error since the primary switch was apparently overlooked and therefore point C is called to set the CP timer with a minimum time. However, if this is not the end of the table, then function 194 flags, for the COMPS computation which is to come, the fact that this is not a clearance pass. Function 195 directs the succeeding COMPS routine to use the velocity of cut B. The COMPS routine then computes a time and velocity for the cut to traverse the distance between the switch at which it was found and the next switch in the route. Decision point 196 determines if this next switch is the primary switch. If it is, then the computation is completed and we have a value of velocity for cut B at the primary switch and the routine jumps to  $C_{prime}$ , CPRIM (shown in FIG. 5G). Assuming, however, that the next switch after the switch at which cut B was lo-

cated was not the primary switch, function 197 uses the calculated velocity as an initial velocity for another computation and jumps to function 189 to repeat the loop. In this fashion, the velocity of cut B is determined at each switch in the route between the switch at which it was located and the primary switch until a point is reached at which decision point 196 determines that we have computed the velocity of cut B at the primary switch and jumps to CPRIM.

The functions 182 through 187 modify the above computations in the situation where a retarder was found between the primary switch and the switch at which cut B was located. Function 182 restores the previously saved information, that is, the retarder number and switch information which was saved when the retarder was located, either at functions 174 and 175 or functions 161 and 162. Decision point 183 determines if this is the master retarder and depending upon the outcome, either the master retarder exit speed is used, function 184, or the group retarder exit speed is used, function 185. In any event, function 186 adds a possible control error to allow for a change in speed between the retarder exit and the switch. Decision point 187 determines if the retarder is directly prior to the primary switch and if it is the routine jumps to CPRIM. If it is not, then the computation or loop, including functions 189 through 197, is entered to project the velocity of cut B from this retarder to the primary switch.

Decision point 145 is entered with the knowledge of the velocity of cuts A and B at the primary switch. Decision point 145 determines whether or not the primary switch is a lap switch. If it is, then the logic of functions 198 through 203 is necessary to determine what distance will be used as the distance to the clearance point where the computation is performed by COMPS. Therefore, function 198 increments the original destination table, only if the primary switch is a lap switch and decision point 199 determines if cut A goes through the second switch in the lap switch.

Function 200 obtains the destination track for cut B and a routine DESEN (FIG. 5C) is performed. Function 201 then increments DESTB so that decision point 202 can determine if cut B goes through the second switch of the lap switch. If either cut A or cut B goes through the second switch, then the distance from the primary switch to the clearance point is utilized whereas if neither cut A nor cut B does through the second switch, then function 203 selects the distance from the second switch to the clearance point as the distance to be utilized in the subsequent computations.

The functions then performed, 146-159, have already been discussed and there is no reason to repeat the discussion.

When a switch wheel detector actuation is recognized, SWWD (FIG. 5A) is called and in that routine, function 72 determines if the ADDSW table for that switch is clear. Of course, with the cut arriving at the switch, the ADDSW table should not be clear. In the event that it is, however, the routine LOST (FIG. 5J) is called in order to attempt to identify the cut.

The first decision point, 210, determines if LOST is on. Assuming this is the first unrecognized wheel detector actuation by SWWD, LOST would not be on and function 211 would reference the first switch number at which the LOST routine is called. It is entirely possible that a number of calls are made to LOST at or about the same time by a number of different switches.



Therefore, these are sequentially listed and function 211 obtains the one at the head of the list. Decision point 212 determines if the entry is zero, minus 1, meaning the end of the list. If it is, function 213 clears LOST and then returns to the routine which called it.

However, assuming that the entry is not zero, and minus 1, at decision point 212, then decision point 214 determines whether ARC, the automatic retarder control program called LOST. This merely provides for the interface between the automatic switching control program which is disclosed herein and another program to control the retarders in the classification yard. If it was the automatic retarder control program which called LOST, then function 215 obtains the adjacent switch number. Assuming, however, that this routine was not called by ARC, then function 216 gets the previous switch number from DESTB. Decision point 217 determines if there is an entry at the ADDSW table for the previous switch. This condition, a cut approaching one switch when its address is still stored in the ADDSW table for the previous switch, is possible if a wheel detector has failed at the previous switch. Assuming there is such an entry, then decision point 218 determines if the previous switch is out of correspondence. If it is not, function 219 determines if the cut is overdue. That is, has the stall timer at the previous switch timed out without recognizing the presence of the cut at the previous switch. If it is overdue, then decision point 220 determines if the switch is lined in accordance with the DESTB. That is, is it laying so as to properly route the cut. If it is, then the routine assumes that the cut which has been detected is the one whose entry was found at the previous switch. From this point on, the LOST routine simulates the necessary functions that would have been performed by SWOCC at the previous switch to cause an entry to be made at the ADDSW table for the switch at which it has just been detected as LOST. Therefore, function 221 clears the axle counter associated with this switch and obtains the number of axles in this cut from CUTST. Function 222 flags this condition as a total wheel detector failure and directs the routine to A. At this point, decision point 223 determines if the number of axles is equal to 0. In going through this example we have assumed that the register AXCNT has just been loaded with a number of axles in the identified cut and therefore the number of axles would not be equal to zero. Function 224 sets the previous switch change bit so as to simulate a wheel detector actuation. Function 225 decrements the count in AXCNT. Function 226 then sets LOST on and the routine returns.

At this point function 224 has set the change bit for the previous switch simulating wheel detector actuation and by reason of the decrementing performed by function 225, the count in the axle counter is one less than the number of axles in the cut. The simulated switch change bit causes routine SWWD (FIG. 5A) to be performed for the previous switch. To quickly run through the operations that will be performed, function 72 determines if the ADDSW table is not clear and function 73 adds one to the temporary axle count, which previously had been 0 and now will contain 1 count. Decision point 74 will determine that this is the first axle and cause the switch protection bit to be set by function 76 and then returned. At that point, the routine will determine that LOST is on, having been set by function 226 and decision point 210, after determin-

ing that LOST is on will cause decision point 227 to determine that LOST is not completed. Decision point 223 will check the number of axles and it will not be 0, therefore, function 224 will again set the previous switch change bit and function 225 will again decrement the count in axles. At this point when the number of axles remaining in axle count will be two less than the number of axles in the cut and the previous portion of SWWD will again be entered due to this simulated wheel detector actuation. At this point, decision point 77 will determine that this is the second axle and cause routine SWOCC (FIG. 5B) to be performed for the previous switch. Briefly, this will ensure that DEST and function 113 will be performed to enter the cut address in the cut into the ADDSW table for the switch at which the LOST cut was identified.

The completion of SWOCC will cause the routine to return to SWWD which will again determine that LOST is on and run through decision points 210, 227, and 223-226. This will again simulate a further wheel detector actuation at the previous switch and also decrement the count in axle counts. This looping process will continue until decision point 223 determines that the number of axles is equal to 0. Decision point 228 will determine that this is not ARC and function 229 will set the switch change bit for the present switch and then function 230 will set LOST completed. This will again cause SWWD to be performed. However, in this performance, decision point 72 will determine that the ADDSW table is not clear since an entry had been made by function 113 in SWOCC referred to above.

The functions described, between the recognition of a LOST cut at a switch, in response to a wheel detector actuation, to the end of the above preceding discussion, will take place prior to the time the second axle of the cut causes a second wheel detector actuation at the switch. Upon this second wheel detector actuation, the routine SWWD will again be performed as we have explained above, for normal operation.

Essentially LOST has simulated the functions that should have taken place at the previous switch in order to make the ADDSW table entry at the primary switch so that the operation could proceed in a normal fashion. Once this simulation is completed, the normal operation takes over and LOST is no longer referred to.

However, this operation presupposed that the previous switch was not out of correspondence, the cut was overdue and the switch was lined right, see the discussion with respect to decision points 218-220.

If, in the example discussed above, decision point 217 indicated that there was no entry for a cut at a previous switch, then function 231 clears the LOST request and function 232 directs the routine to point C to take the next LOST call in sequence. Since there is no entry at a preceding switch, there is nothing the program can do to obtain the identify of the particular cut and therefore it is allowed to proceed through the yard.

Similarly, if decision point 219 indicates that the cut which has been identified at decision point 217 is not overdue, then the LOST cut is not the identified cut and functions 231 and 232 are also executed. In the same fashion, if decision point 220 indicates that the previous switch is not lined properly, the tentative cut identification made at decision point 217 is not the LOST cut, since with this switch improperly lined, the identified cut would not reach the switch at which the LOST cut was identified. As a result, functions 231 and



232 are executed and the next sequential LOST call is interrogated.

There is one further case that the routine must handle, that is, the case where the LOST cut is identified, by its entry at the preceding switch, but that switch was out of correspondence. As a result, the switch at which the cut was identified may not be in the cut's route to its intended destination track. Therefore, function 233 stores the cut's destination, obtained from DESTB. Decision point 234 determines if that destination is still achievable and if it is, then the routine skips to B. If the destination is not achievable, then function 235 obtains another destination track for the cut from table NEXTB. Subsequently the program will also skip to point B.

At this point in the routine, the routine DEST (FIG. 5D) is performed and subsequently, function 236 enters the cut address in the ADDSW table for the next switch in the route. The routing POSSW positions this switch if such action is safe and CALARV calculates the expected arrival time of this cut at that switch. Decision point 237 determines if the prior switch was before a retarder. If it is, function 237 causes data to be transferred to the automatic retarder control program. In either event, function 239 is then performed to reset the SWOC bit indicator for the previous switch and to set the switch okay bit for that switch. Since the out of correspondence condition has been handled by LOST, there is no need to further retain this indication. And then the program loops around to decision point 228 and then sets the switch change bit (229) for the switch at which the cut was identified and then sets LOST completed, by function 230.

Although we have not completed the discussion relating to the routines SWWD and SWOCC, we have discussed the normal operation of these routines, where there is no malfunction to be reported and accounted for. Before returning to, and completing, the discussion of, SWWD and SWOCC, we will now discuss a routine which handles switch unoccupancy indications and is entitled "SUN." Unlike many of the other routines that have been discussed before, SUN is not called in response to particular changes in field conditions. Instead, SUN is executed at periodic intervals to handle all switch unoccupancy indications that have been received from the field subsequent to the last execution of SUN. Although SUN does not operate in response to field changes, its delay period is ordinarily less than one second and indeed may be on the order to tenths of seconds. In comparison to the speed of movement of a cut through a classification yard, we can assume that SUN responds almost instantaneously to switch unoccupancy indications.

When a presence detector, which had been signaling the presence of a cut, changes to a no presence indication, the routine SUN responds to that as a cut having left the switch. In addition, SUN also checks the cornering prevention timers which are set by function 159 (FIG. 5G) to determine whether any of them have expired. Of course, when a cornering prevention timer has expired, then it is safe to throw the switch associated with that timer as cornering has been avoided.

FIG. 6A and FIG. 6B, taken together, show the SUN routine. The first function, 240, sets a pass counter to be equal to NOTKS (number of track switches in the yard). Decision point 241 determines for the last switch whether or not the cornering prevention timer is run-

ning and if it is, function 242 decrements the timer, and decision point 243 determines whether or not the timer has expired. If the timer has expired, then function 244 resets the switch protective bit thus allowing this switch to be thrown, cornering prevention having been assured. If the timer did not expire, then function 244 would not be performed and the routine would skip to function 245 to decrement the pass number. In a like fashion, if the CP timer was not running, then functions 242, 243 and 244 would be unnecessary and function 245 would decrement the pass number.

Decision point 246 determines whether or not each of the CP timers has been checked and if it has, function 245 clears the pass number to set up to perform a different function. However, if decision point 246 determines that we are not done, there are other CP timers which have not been checked and therefore the program loops back to decision point 241. This loop, between decision points 241 and 246, is completed, once for each timer and function 244 makes the appropriate entry when a timer is detected as having been expired.

The functions 248-255 identify any status changes which may have occurred since the last performance of SUN. Functions 265 and 266 control the number of passes through the loop including functions 248-255 so as to ensure that the entire yard is checked for status changes. The remaining functions of SUN, functions 267-310, service the status changes as required.

Before discussing functions 248-255 in detail, a word about the organization of the data referred to therein would be helpful in understanding the operation. For each switch in the yard, there are three bits which have to be checked to determine status changes. These are the stall protect bits, the presence detector status, and the cornering prevention protection bits. The cornering prevention protection bits generation has been discussed with respect to CP and the clearing of these bits has been discussed with reference to functions 241-244, above. The generation of the stall protect bits and their meaning will become clear when FIG. 7, relating to CTC, is discussed. Briefly, the routine CALARV (FIG. 5E) computes an expected time of arrival for a cut at the next switch in its route when it passes a preceding switch. The corner prevention logic is based upon the assumption that a cut will proceed at or near its theoretical velocity. However, if a cut does not arrive at the next switch in its route within the allotted time, this fact may indicate that the cut is stalled somewhere on the track between the two switches. Thus, this condition may require additional cornering prevention action to be taken. The stall protect bits are related to the condition of the various stall timers, that is, whether or not any of them have timed out. The presence detectors placement in the classification yard has been discussed with reference to FIG. 1. Each presence detector may provide a signal indicating the presence of a cut or alternatively, may provide a signal indicating the absence of a cut. The change from indications of the presence of a cut to an indication of the absence of a cut is the significant presence detector occurrence which must be detected by SUN and responded to. This is an indication that the cut has left the area of the switch and for one thing, the ADDSW table entry for that cut at the switch can be removed by the routine REMOV (FIG. 6C).



Since there are only three significant bits associated with each switch in the yard, the routine, instead of making one pass through its logic for each switch, combines the information with respect to a number of switches in one word and examines that word during a pass. For instance, if there are 16 switches in the yard and an 8-bit word may be examined by the logic, than only two passes would be necessary to check all of the switches in the yard. Knowledge of the pass number and the location of the particular bit in the word that was set will then enable the logic to determine the particular switch to which that bit referred which must be serviced. With that in mind, we will now examine the operation of functions 248-255.

Function 248 obtains the stall protect bits for a number of switches. Function 249 OR's this information with the presence detector bits for the same switches. As a result of this logical operation, any switch which had its stall protect bit set or which had its presence detector bit set would result in the corresponding bit being set in the result. The resulting data is then OR'ed with the CP protect bits for the corresponding switches by function 250. The overall result is that for any switch which had its stall protect bit set, its presence detector bit set, or its CP protect bit set, the resulting data would have the corresponding bit set. The result is then stored, by function 251, at a temporary location SWOCC temp word. Function 252 stores the pass number to aid in identification of the significance of any bit in the word. Function 253 then complements the word which had previously been stored by function 251. Function 254 AND's this complemented version with the previous word which had been stored in SWOCC prior to initiating SUN. Decision point 255 then determines whether or not any of the resulting bits are set. Any set bit would indicate a status change which required servicing.

The first function in the servicing routine, CMZ, counts the significant O's to determine the first set bit. The first item of business, function 267, is to reset this bit. Since it now will be serviced, it is reset so that it will only be serviced once. Function 268 determines, from the position of the bit which is set and the pass number, the switch number to which it refers. Decision point 269 determines whether the ADDSW table is clear and if it is not, function 270 determines if the cut has arrived. It will be recalled that in SWWD (FIG. 5A) on the next to the last axle count, function 81 sets cut arrived. If either the ADDSW table is clear or the cut has not yet arrived, then function 275 reports this as a position detector bobble, a false indication, and returns to SUN 7 to service further set bits. However, if the ADDSW table is not clear and the cut has arrived, then function 271 obtains the value in the temporary AXCNT (indicating the number of axles which have been counted). Decision point 273 determines whether or not a wheel detector has failed by comparing the value is TEMPAXCNT with the total number of axles in the cut found in CUTST. If a wheel detector has failed, function 274 flags this and reports the failure. In any event, decision point 276 determines whether or not the switch is out of correspondence. If it is, this terminates the servicing for the bit and the routine loops back to SUN 7 to service further indication changes. Since the switch was out of correspondence, the system has no way of knowing where the cut was directed and therefore the ADDSW table entry is not removed so

that when the cut is located at the next switch it reaches, LOST, by reference to the ADDSW table for this switch, will be able to determine the identity of the cut.

However, if the switch is in correspondence, then REMOV is performed to remove the ADDSW table entry for this switch. This routine is shown in FIG. 6C. The first function, 311, determines if this is the last switch in the route and if it is function 312 indicates that information must be transferred as in effect the cut is leaving the system. If this is not the last switch in the route, then decision point 313 determines if a retarder is next and if it is, RETOT is performed to transfer information to the retarder operating system. Whether or not a retarder is next function 314 is performed to remove the ADDSW entry and function 315 repacks the ADDSW table to insert the next sequential entry in the effective location. Decision point 316 then determines if the repacked ADDSW table is clear. If it is, function 317 can release that block since it is no longer necessary. However, if the table is not clear, then POSSW is performed to position this switch in accordance with the destination of the cut which is identified by the ADDSW entry. Subsequently, function 318 sets up a new register for AXLSW to count the number of axles traversing the switch. That completes REMOV and the routine then returns to SUN to perform function 277 to reset the position detector failure bit. Since the position detector has operated properly, the failure bit is reset. Function 288 clears the temporary axle count inasmuch as this is no longer needed.

Function 289 then finds the first entry in DESTB corresponding to the first switch in the route for this cut. Functions 290-293 will remove the ADDSW table entries for this cut from all preceding switches in the route if that has not already been accomplished by some prior operation of SUN.

Function 290 obtains the switch number corresponding to the first entry in DESTB. Decision point 291 determines if the ADDSW table is clear and if it is, decision point 293 is operated. However, assuming it is not, then decision point 292 determines if this is the same cut whose address had just been removed by REMOV. If it is not, then decision point 293 is also performed. However, if it is, then decision point 295 determines if cut arrived has been set for this switch. If it is not, function 296 flags this as a wheel detector failure. Since the cut has proceeded beyond this switch, the failure to set cut arrived is due to the failure of the wheel detectors and this is flagged.

Alternatively, if cut arrived is set, then decision point 297 determines if the presence of the entry is due to a stall.

If the entry is not due to a stall, then decision point 298 determines if the presence of the entry is due to cornering protection and if it is not, then the presence of this entry is flagged as due to a presence detector failure by function 299. In any event, the ADDSW table entry is removed and then function 300 clears the temporary axle count for that switch and function 301 resets the position detector failure bit and then decision point 293 is performed. This checks whether or not the switch which is being examined is the primary switch. If it is not, function 294 increments DESTB to look at the next switch in the route and similar functions are performed for each switch from the first switch in the route up to the primary switch. The primary switch is



the switch whose identity is determined by function 268. When the primary switch is reached then one switch has been serviced and the routine loops back to SUN 7 to determine if any further bits had been set. For each bit which is set, the same functions, that is, functions 267-301, are performed in turn.

When decision point 255 determines that there are no longer any bits set, then function 265 increments the pass number and decision point 266 determines whether or not the functions have been completed. Assuming, in a typical example, that two passes would be required through functions 248-255 to completely check an entire classification yard, on the second pass, decision point 266 would determine that we are not done and loop back and repeat the functions for the remaining switches in the yard. When decision point 266 determines that each switch has been checked, then the program skips to point A (FIG. 6B).

The function performed by this routine is to service the presence detector failure indications for reporting purposes. Function 302 obtains the presence detector failure bits which had been set in various routines discussed above. For instance, function 299 (FIG. 6A) after identifying a presence detector failure, flags it as such. This is the data to which function 302 has reference. However, not all presence detector failure bits that are set actually indicate presence detector failures. Therefore, logical operations of functions 303 and 304 are performed. Function 303 performs an exclusive OR operation between the presence detector failure bits and the presence detector status. The result of that operation is then logically AND'ed with the presence detector status. The resulting bits, if set, indicate presence detector failure which must be serviced. Decision point 305 determines the existence of any such failures. If there are none, then decision point 309 is performed to determine if the routine is completed. If it is, it is turned off. If it is not, then function 310 is performed to look at the next group of presence detector failure bits, if any exist.

If there were presence detector failure bits that will set as determined at decision point 305, the next function to be performed would be to count the most significant 0's to determine which presence detector failure caused the bit to set. Function 306 would reset the bit and function 307 would report the presence detector failure. Decision point 308 then would check to see if any more are set and if there were, the same functions would again be performed, that is, the counting of most significant 0's, and functions 306 and 307. That concludes the operation of SUN.

The only remaining routine to be discussed is CTC which performs a number of functions. In essence CTC handles the timers and determines the expiration of the various timers and sets the necessary software flags to indicate the expiration.

The first function in CTC, 311, is to check the destination track timers. It will be recalled that in SWOCC (FIG. 5B) if decision point 94 indicated that the switch was the last switch in the route, functions 95 and 96 computed a time to clearance and set up to prevent fouling. In essence this initiates the running of a timer for that destination track. It is those timers to which function 311 has reference.

Decision point 312 determines if the first timer is running and if it is, function 313 decrements the timer. Decision point 314 determines if the timer has expired

and if it has function 315 flags this occurrence as  $\overline{\text{TWD}}$  failure. The significance of the expiration of the track timer is that the wheel detectors have not failed. Since the track timer is started upon the actuation of a wheel detector, its expiration is evidence that the wheel detectors are operating properly. Subsequently, function 316 resets CTC, the clearance track circuit, as ineligible for a foul, meaning that CTC is eligible for a foul since the timer has expired.

If the timer has not expired or at the completion of function 316, decision point 317 determines whether or not each of the track timers has been checked. If it has not, then the routine loops back to decision point 312 to check other timers. One passage is made through this loop for each timer until they have all been checked.

At this point function 318 obtains the CTC occupancy indications, that is, the presence detector current status. Function 319 AND's these indications with  $\overline{\text{TWDF}}$ . The result is stored, by function 320, in  $\overline{\text{TWDF}}$ . The remaining functions related to the track timers are performed subsequently and decisions are made as to whether or not a wheel detector failure or a fouled track is present. However, functions 321-328 related to the switch stall timers and these will now be discussed.

CALARV (FIG. 5E) computed a time quantity for the expected arrival of a cut at the next switch in the route. This time value was stored, by function 209, in a stall timer. There is one stall timer for each switch in the yard.

Function 321 directs the routine to the first stall timer and decision point 322 determines if it is running. If it is not, function 327 looks at the next stall timer and decision point 328 determines if we are completed. That is, if there is another stall timer being referred to. If there is, the program loops back and decision point 322 determines if that timer is running. For each timer which is running, function 323 decrements it and decision point 324 determines if it has expired. If it has not expired, then the routine jumps to function 327 to look at the next timer. However, if the timer has expired, this means that absent a wheel detector failure, the cut is overdue and function 325 flags the cut as overdue. Function 326 prevents a cornering condition by preventing the preceding switch, that is, the switch from which the cut was traveling, from being thrown. The cornering prevention logic is not based upon the act actually passing the clearance point, but is based upon a computed time for the cut to reach the clearance point. If the cut becomes stalled, this may indicate that it was not traveling at its expected theoretical velocity and therefore the cornering prevention logic, although accurate, based on the assumption that the cut was traveling at its theoretical velocity, may not be related to reality. As a result, function 326 assures cornering prevention by disabling the preceding switch from being thrown.

Once each of the stall timers has been checked and appropriate action taken, the routine proceeds to decision point 329 to determine if there are any foul tracks or wheel detector failures that must be reported. Decision point 332 determines if the clearance track circuit is unoccupied. Of course, this is merely a check of the presence detector current status. If it is unoccupied, then function 333 resets any foul indication for this track. Since it is unoccupied, it can hardly be fouled



and then the routine loops back to decision point 329 to service the remaining destination tracks. If the CTC is occupied, then decision point 334 determines if the CTC is ineligible for foul. For instance, if a track timer is running, CTC would be ineligible for a foul inasmuch as the expected time for the cut to clear the clearance track circuit is not yet expired. If CTC is ineligible for a foul, then that concludes the servicing for this destination track and the routine loops back and selects the next. For any destination track where the CTC is occupied and CTC is not ineligible, then decision point 335 determines if the track is fouled, that is, whether or not a foul has been reported for that track. If no foul has been reported for the track then function 336 reports this as a foul track. Whether or not function 336 is performed, the next operation is decision point 337 which determines if  $\overline{\text{TWDF}}$  is set. If it is, that indicates that the wheel detectors have operated and therefore no further servicing is required. At this point the program loops back and picks another destination track to service.

However, if  $\overline{\text{TWDF}}$  is not set then decision point 338 determines if there is an ADDSW entry. If there is not, then no further operation is required to service this destination track. However, if there is an ADDSW entry, decision point 339 determines if the switch is lined correctly. If it is not, again, no further servicing is required. However, if the switch is lined right, decision point 340 determines if the cut is overdue. Of course, this has reference to the stall timer having timed out. If the cut is not overdue, then this also concludes the servicing for this track. However, if the cut is overdue, then function 341 turns off the position calls for the preceding switch and function 342 flags this as a  $\overline{\text{TWDF}}$ , a total wheel detector failure. Subsequently, REMOV removes the ADDSW entry.

After the functions 329-342 are performed for each of the destination tracks in the yard, then the next pass at decision point 329 will indicate that we are done and function 330 will delay for a predetermined period of time. Subsequently, function 331 initiates the CTC routine once again.

Now that all the other routines have been discussed we can return to the undiscussed portions of SWWD (FIG. 5A) and SWOCC (FIG. 5B).

We had terminated our discussion of SWWD at the point where decision points 80 and 81 indicated that the count in AXCNT was negative. That is, the wheel detectors have recorded more axles than there are in the cut. On the first overcount, decision point 81 would return the routine to the initiation of SWWD. On the second or subsequent overcounts, decision point 82 would determine if there was another entry in ADDSW. If there is not, the result would be the same, no further processing in SWWD.

However, if there is another entry in ADDSW then REMOV (FIG. 6C) is called. This causes the initial ADDSW entry to be removed. Subsequently, decision point 350 determines if the corner protection timer is on. If it is, function 531 flags this as a misroute due to cornering prevention and then calls SWOCC. If the cornering protection timer is not on, function 352 flags this as a misroute to the catch-up. That is, a following cut has caught up to a preceding cut and has followed it through the switch. Subsequently SWOCC is also called.

In our prior discussion of SWOCC, we had omitted any discussion with respect to functions 355-366. At this point, after discussing the operation of SWWD in connection with misroutes due to cornering protection and catch-up, we can now discuss these functions in SWOCC.

If decision point 92 indicates that this is a misroute, that is, the cut is traversing the switch in the wrong direction, function 355 obtains from NEXTB the next best destination track. DESEN (FIG. 5C) is called. Subsequent to the running of DESEN, decision point 356 determines if a misroute had already been identified by SWOCC. Decision point 357 determines if a misroute due to catch-up, decision point 358 determines a misroute due to cornering protection and decision point 359 determines a misroute due to presence detector or loop failure. If none of these are the causes of the failure, then decision point 360 determines if the switch is in manual operation. If it is, then function 362 flags this as a switch in manual operation. However, if the switch is not in manual operation, then function 361 flags this as a switch failure. In any event, the succeeding decision point, 93, determines if continue is set in DESTB. If it is, function 363 looks at the next DESTB entry and decision point 364 determines if do not throw is set. If it is, decision point 364 determines if this is a misroute.

If do not throw is not set, then decision point 366 determines if this is a misroute. Based upon the outcome of decision points 365 or 366, depending upon which is performed, the routine can return to the initial portion of SWOCC to perform the initial functions for this new switch or the routine can proceed to decision point 94 to determine if this is the last switch in the route.

## OPERATION

From the detailed description of the various routines making up the program, the manner in which the different routines interface with one another, and the manner in which the entire program interfaces with and responds to and controls the classification yard, the operation of the system should be apparent. However, in order to recapitulate we will briefly examine the manner in which the system operates.

Prior to the actual start of humping operations, it is necessary for the computer to be loaded with the program, as disclosed in FIGS. 3 through 7 as well as the various tables characteristic of the yard which are referred to in the specification. In particular, DESTB, NEXTB, as well as the tables defining the distance between switches and the grade between switches. At this point, the consist, listing the cars in the train in sequence, can be received to form the basis for CUTST.

As a cut is uncoupled, CARDET counts the axles for each car and transfers pertinent cut parameters to CUTST. The address of the cut information located in CUTST is entered into the ADDSW table for the first switch for FORW and POSSW positions this switch if such action is safe. As the second car is detected, the preceding is repeated with the following exceptions. If the cut is detected as a multi-car cut, then no entry is made in ADDSW but instead the total of the various parameters are added into the appropriate location in CUTST. If this is not a multi-car cut, then an additional entry is made into the ADDSW table for the first switch. The second entry of a cut address into the ADDSW table for the first switch will have no effect on



the positioning of the switch so long as the first entry remains. of course, when the first cut clears the first switch, as noted by SUN, REMOV will see that the cut's address is removed from the ADDSW table, then allowing the first switch to be positioned in accordance with the destination track for the second cut.

This same operation is repeated for each identified cut as it is uncoupled from the train and its presence is noted by wheel detectors WD1a and WD1b and the cut light detector CL.

The following operation will be generalized for the case of any particular cut approaching any particular switch in the yard. The first wheel detector actuation will cause the ADDSW entry to be found. If no entry is found, then LOST is entered to handle the situation in accordance with the available information.

On the first wheel detector actuation, the software switch protection will be set prohibiting the throwing of this switch until the cut clears the switch.

Upon the second wheel detector actuation, the loop failure detector flag will be set as an aid to identifying presence detector failures. The correspondence of the switch is determined, by SWOCC, and if a misroute is found, it is flagged. Furthermore, a misroute will result in a reference to NEXTB to determine a new destination for the cut. Once the destination of the cut has been determined, either from NEXTB or from CUTST, if no misroute is flagged, then the destination is checked for blocked or fouled tracks by DEST. If a blocked or fouled track is found, then the cut is re-routed in accordance with the conditions.

At the same time, an entry is made into the ADDSW table for the next switch in the route for this cut and this switch is positioned if possible, that is, if it is not occupied. Furthermore, CALARV is called in order to calculate the front arrival time of the cut at the next switch in the route and a stall timer is set and started to detect stalled cuts if the cut does not arrive at the switch within the allotted time.

Upon the next to the last wheel detector actuation, the cornering prevention routine is called and a cornering prevention timer is set and started in order to provide cornering protection for this cut. At this time, the indicator cut arrived is set indicating that the cut has arrived at the switch.

Upon detecting two overcounts, the routine SWWD also handles misroutes due to catch-ups or cornering protection.

As the presence detectors become unoccupied due to the cuts clearing the switch, the cut address is removed from the ADDSW table assuming that the cut arrived indicator has been set. This operation will also indicate wheel detector failures, if any. Furthermore, all previous ADDSW tables for all previous switches in the route are checked and the entries removed if any are still there. Furthermore, the presence of entries at these switches indicates position detector or wheel detector failures, and such failures are flagged as appropriate.

The CTC routine runs upon its own time basis servicing their various times which had been started by other routines. In particular, this serves to detect fouled tracks and will also provide detection of wheel detector failures. If such wheel detector failures are discovered in the course of the CTC routine, the routine REMOV will be entered to remove these superfluous cut address enters from the appropriate ADDSW tables.

As the various cuts clear the clearance track circuit subsequent to the last switch in the route, the necessary arrangements are made to have the information respecting the cut transferred out of the switching control system since it is no longer necessary or appropriate to retain this information.

From the foregoing it should be apparent that the logic contained in the routines disclosed in FIGS. 3-7 is entirely divorced from the characteristics of the yard to which the system is to be applied. Thus, the logic is entirely general and with the appropriate DESTB, NEXTB, and associated tables defining the yard, the system can be applied to any classification yard.

We claim:

1. Apparatus for controlling the switches in a railway classification yard to route each cut of cars rolling down the crest to its proper destination including car sensing devices at the crest and at each switch, said apparatus comprising;

computer means responsive to said sensing devices and which has stored therein:

a. a sequential list of cars as they appear at the crest and an intended destination track for each of said cars,

b. the characteristics of the yard including a table which identifies, for each destination track, the switches included in the route from the crest to said destination track and the proper position for each of said switches to route a cut from said crest to said destination track,

c. said computer means further providing,

1. means for determining the number of cars in each cut,

2. means responsive, to each cut at said crest, to said table and said list to determine the position of the first switch in said classification yard,

3. means responsive to said last named means to correctly position said first switch to route said cut to its destination track unless the proper positioning of said switch would result in an unsafe condition,

4. means responsive, to the presence of a cut at any subsequent switch, to said table and said list to determine the position of the next succeeding switch in the yard for the cut to reach its required destination track,

5. means responsive to said last named means to correctly position said next succeeding switch unless the correct positioning of said next succeeding switch would result in an unsafe condition,

6. means to compare the position of each switch as each cut is adjacent thereto with the required position for said cut to reach its intended destination track,

7. means responsive to said last named means when indicating a lack of comparison and to said table to select a different destination track for said cut, and

8. means responsive to said last named means to modify said list for said cut by substituting said different destination track for said intended destination track.

2. In the apparatus of claim 1 including apparatus to prevent a switch movement until a cut has cleared a switch, said apparatus comprising:



bistable means to prohibit the movement of said switch, when in one condition, and allowing movement of said switch, when in said other condition, first means responsive to the actuation, by said cut, of said sensing device at a switch to set said bistable means to its one condition, timing means, second means responsive to said sensing device for setting said timing means to a predetermined value sufficient to allow said cut to clear said switch, and third means, responsive to said timing means having timed out said predetermined value of time to set said bistable means to its other condition.

3. The apparatus of claim 2 including apparatus to prevent cornering, said apparatus comprising:

- means responsive to the multiple actuation of said sensing device at a switch by the multiple axles of said cut to compute the velocity of said cut,
- time computing means responsive to said last named means to compute and store the expected time for said cut to reach the clearance point of said switch,
- identifying means responsive to the velocity computation for said cut to identify and locate the next immediately following cut destined for said switch,
- means responsive to said last named means to compute the estimated velocity for said cut at said switch,
- said time computing means being further responsive to said last named means to compute and store the expected time for said last named cut to reach the clearance point of said switch,
- comparison means, responsive to said time computing means last recited operation, to compute the time difference between said first and second time computed values,
- and means to disable said second means and set said timing means to said difference time if said difference time exceeds the predetermined value of time.

4. The apparatus of claim 3 including further apparatus to provide cornering prevention, said apparatus comprising,

- means responsive to the multiple actuation of said sensing device at a switch by the multiple axles of a cut to compute the velocity of said cut at said switch,
- time and velocity estimating means, responsive to said last named means and further responsive to the grade and distance between said switch and the next switch in the route for said cut to compute the estimated velocity of said cut at said next switch and the expected time of travel for said cut from said switch to said next switch,
- storage means, responsive to said last named means to store said estimated velocity of said cut at said next switch,
- timing means,
- means responsive to said time and velocity estimating means to set said timer to said estimated time,
- monitoring means to indicate when said timing means times out said estimated time,
- means responsive to said monitoring means to set said bistable means to its one position when said monitoring means indicates said timing means has timed out.

5. Apparatus for controlling the switches in a railway classification yard in order to provide cornering protec-

tion for each cut of cars rolling from the crest to its proper destination track at each of the several switches in the route of the car wherein said classification yard includes car sensing devices at the crest and at each switch, said apparatus comprising;

- first means responsive to said sensing devices and which has stored therein data which identifies, for each destination track, the switches included in the route from the crest to said destination track and the proper position for each of said switches to route a cut from said crest to said destination track, said first means further including second means for computing when a cut is sensed at a switch, the estimated time required for said cut to travel from said switch to the clearance point associated with said switch,
- means for identifying and locating a second cut immediately following said cut and destined for said switch,
- means for computing the estimated velocity of said second cut at said switch,
- computing means to compute the estimated time required for said second cut to travel from said switch to the clearance point associated with said switch,
- and means, if said last named time is less than said first named time, to disable said switch from being thrown for a time at least equal to the difference between said times.

6. In the apparatus of claim 5 wherein said first means further includes,

- a plurality of registers, each of said registers associated with one of the switches in the yard and each said register contains the designation of the next approaching cut,
- storage means containing data with respect to each cut, said data including for each cut, its intended destination track, said data being correlated with the designation stored in each register,
- said identifying and locating means, including means to interrogate in sequence, the register associated with the immediately preceding switch in the route of said cut to identify and locate said immediately succeeding cut.

7. The apparatus of claim 6 including additional apparatus to limit the operation of said identifying and locating means, said additional apparatus comprising,

- counting means responsive to the number of registers interrogated by said means to interrogate,
- register means settable to a predetermined number, and comparing means to limit the operation of said identifying and locating means to the predetermined number of interrogation stored in said register means.

8. Apparatus for controlling the switches in a railway classification yard to route each cut of cars rolling down the crest to its proper destination including car sensing devices at the crest and at each switch, said apparatus comprising;

- computer means responsive to said sensing devices and which has stored therein:
  - a. a sequential list of cars as they appear at the crest and an intended destination track for each of said cars,
  - b. the characteristics of the yard including a table which identifies, for each destination track, the switches included in the route from the crest to said



41

destination track and the proper position for each of said switches to route a cut from said crest to said destination track,

c. said computer means further providing

1. means, responsive to operationn of one of said car sensing devices at one of said switches, to compare said switch position with said table to determine if said switch is correctly positioned,
2. means, responsive to said last named means when a lack of comparison has been determined, to select a different destination track, and,
3. means, responsive to said last named means, to modify said list by substituting said different destination track for said intended destination track.

9. The apparatus of claim 8 in which said means to select a different destination track includes a second table identifying, for each switch and destination track said different destination track.

10. Apparatus for controlling the switches in a railway classification yard to route each cut of cars rolling down the crest to its proper destination including car sensing devices at the crest and at each switch, said apparatus comprising,

computer means including a digital computer, a plurality of tables stored therein, said tables uniquely defining characteristics of said yard, a sequential list of cars as they appear at the crest and an intended destination track for each of said cars stored in said computer and a program distinct

42

from said tables also stored in said computer, said computer means responsive to said car sensing devices said list, said tables and said program to control each of said switches to route each of said cars to its destination track.

11. The apparatus of claim 10 in which said program and said digital computer include means to determine a misroute of a cut, means responsive thereto to refer to one of said tables to determine another destination track for said cut and further means responsive to said means to determine for routing said cut to said another destination track.

12. The apparatus of claim 10 in which said program and said digital computer include means to prevent cornering.

13. The apparatus of claim 10 in which said program and said digital computer include means to detect failure of said sensing devices.

14. The apparatus of claim 10 in which said program and said digital computer include means to detect blocked tracks.

15. The apparatus of claim 10 in which said program and said digital computer include means to detect fouled tracks.

16. The apparatus of claim 10 in which said program and said digital computer include means to detect stalled cars.

\* \* \* \* \*

30

35

40

45

50

55

60

65



**UNITED STATES PATENT OFFICE**  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 3,865,042

DATED : February 11, 1975

INVENTOR(S) : John J. DiPaola et al

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

On two locations on the first page of the above referenced patent correct "DePaola" to -- DiPaola --.

In column 2, line 2 change "poptentially" to -- potentially --.

Signed and sealed this 22nd day of April 1975.

(SEAL)  
Attest:

RUTH C. MASON  
Attesting Officer

C. MARSHALL DANN  
Commissioner of Patents  
and Trademarks