

**CALL PROCESSOR  
DESCRIPTION OF SYSTEM OPERATION  
NO. 101 ELECTRONIC SWITCHING SYSTEM**

CONTENTS	PAGE	CONTENTS	PAGE
1. GENERAL . . . . .	2	12. THE PROGRAM SHIFT REGISTER CIRCUIT . . . . .	35
2. ORGANIZATION OF THE STORES . . . . .	2	A. General . . . . .	35
A. Call Store Sectors . . . . .	2	B. The Clock Switch Circuit . . . . .	35
B. Time Slots . . . . .	2	C. Gating the Shift Command Addresses . . . . .	39
C. Progress Marks . . . . .	2	D. The SHR (Shift Right) Command . . . . .	39
D. The Program Store . . . . .	2	E. The SHL (Shift Left) Command . . . . .	39
3. FUNDAMENTAL CIRCUIT FEATURES OF THE CALL PROCESSOR . . . . .	5	F. The SH0 (Shift to Zero) Command . . . . .	39
A. General . . . . .	5	G. The LSH (Long Shift) Command . . . . .	39
B. Registers . . . . .	5	H. The AD1-CO (Add One to the CO) and the INV (Invert) Commands . . . . .	39
C. Connections to the Input-Output Area . . . . .	5	13. MAINTENANCE COMMAND LOGIC . . . . .	39
D. Command Translators . . . . .	5	A. The EXC (Execute) Command . . . . .	39
E. Conditional Transfer Command Logic . . . . .	5	B. Operating the Maintenance Center Keys . . . . .	43
F. Advance Command . . . . .	5	C. The All Zeros Detector . . . . .	43
4. THE PROCESSING OF CALLS . . . . .	9	D. Parity and Transfer Errors . . . . .	43
5. DETAILED BLOCK DIAGRAM OF THE CALL PROCESSOR . . . . .	9	E. The WRT EB1 Command . . . . .	44
6. THE CLOCK CIRCUIT . . . . .	15	14. PC AND PCI LEADS . . . . .	44
7. READY COMMAND LOGIC AND TIMING . . . . .	19	A. General . . . . .	44
8. THE ADV (ADVANCE) COMMAND . . . . .	19	B. PC Leads . . . . .	44
9. THE STT (START) COMMAND . . . . .	25	C. PCI Leads . . . . .	44
10. THE LDT (LOAD DATA) COMMAND . . . . .	29	15. THE LINE INFORMATION STORE CONTROL CIRCUIT . . . . .	44
11. THE LSD (LOAD SENDER) COMMAND . . . . .	29	A. General . . . . .	44
		B. The Red L Commands . . . . .	49
		16. THE IDENTIFIED OUTWARD DIALING INTERFACE CIRCUIT . . . . .	49
		17. REFERENCES . . . . .	49

This material is for use of Bell System employees only, and for Bell System purposes only, and its distribution is in no sense a publication. Neither the material nor any portion thereof is to be reproduced in any form without the written permission of the American Telephone and Telegraph Company.

**1. GENERAL**

**1.01** The call processor can be thought of as a device for solving the problems presented to it by the system conditions and by the input-output area of the control unit. The call processor consists of two semipermanent stores, a temporary store, some registers, counters, buffers, and gating paths, all of which work together with several blocks of wired logic. A program has been placed in one of the semipermanent stores. The program is designed to work with the call processing equipment to solve the problems which can be presented to it.

**2. ORGANIZATION OF THE STORES (Fig. 1)**

**A. Call Store Sectors**

**2.01** It may be seen from Fig. 1 that the greater part of the call store has been divided into sectors. There is one sector for each switch unit and there may be as many as 32 switch units or sectors; the sectors are numbered from 0 to 31.

**B. Time Slots**

**2.02** Each sector has been divided into time slots. The time slot areas in the call store are intended to represent the actual time slots in the switch units. The number of time slots in each sector will be determined by traffic considerations. There may be as many as 50 of them in any one sector. The lowest numbered time slot is No. 2. It corresponds to time slot No. 1 on the even bus in the switch unit. Call store time slot No. 3 corresponds to No. 1 on the odd bus in the switch unit. No. 4 in the call store is equivalent to No. 2 on the even switch unit bus, etc, up to a maximum of 50 time slots which are numbered from 2 to 51 in the call store.

**C. Progress Marks**

**2.03** Each time slot consists of four 16-bit words. Because the 4-word time slot is used as temporary storage to keep a running record of a call in process, certain types of information have been somewhat loosely assigned to each time slot word. For example, a progress mark is written twice into word one of the "typical" time slot in Fig. 1. The progress mark indicates the status of a telephone call; for example, progress mark zero indicates that there is no call in the time slot under consideration.

**2.04** Progress mark 1 means that the "A" party has gone off-hook but that he has not yet received dial tone. Progress mark 103 means that an attendant is talking to one party. There are about 50 progress marks.

**D. The Program Store**

**2.05** The call processor makes use of a stored program to direct the problem solving job. The program consists of a sequence of commands. The commands are instructions to the control unit circuitry to perform certain functions; for example, "read the call store at a given address" or "gate the contents of one register into another register." (See CD-1H049 for a complete description of each command.) The entire program for the No. 101 ESS has been placed in a semipermanent twister memory called the program store.

**2.06** As may be seen in Fig. 1 the program is divided into three major categories: reference data, call processing, and maintenance. The call processing program has been further divided into routines which have been designed to solve specific problems such as those presented by the progress marks and other system conditions. The routines are, of course, made up of commands.

**2.07** Each 22-bit word in the program store contains either one 20-bit (J) or two 10-bit (K) commands. In either case, bit 1 of the 22-bit word is the "transfer allowed" bit, and bit 22 establishes even parity for the entire word.

**2.08** The J and K commands are divided into a 5-bit operation code and an address. The operation code is intended to identify the commands, while the address provides space for useful numbers concerning the command. In the case of a transfer command the address is an actual address or location in the program store. For some of the J commands, a 16-bit address is required. In this case, the transfer allowed bit will be used as the sixteenth bit.

**2.09** Since there are about 150 commands, the 5-bit operation code is not sufficient to identify them all. However, many of the commands do not require an address; for example, "add one to the call store address register" or

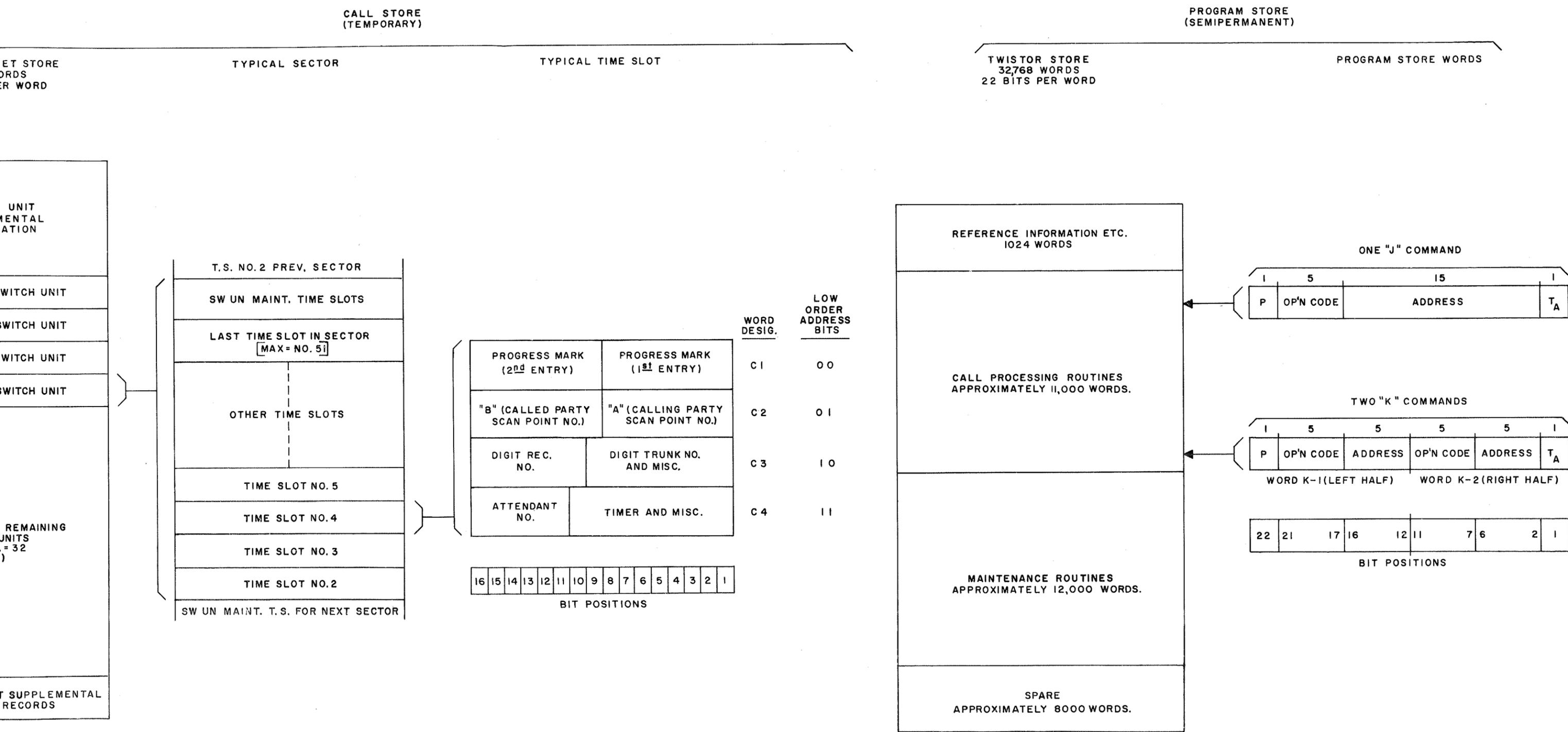


Fig. 1 - Basic Arrangement of Word Information in Call and Program Stores

CALL STORE  
(TEMPORARY)

PROGRAM STORE  
(SEMIPERMANENT)

FERRITE SHEET STORE  
4096 WORDS  
16 BITS PER WORD

TYPICAL SECTOR

TYPICAL TIME SLOT

TWISTOR STORE  
32,768 WORDS  
22 BITS PER WORD

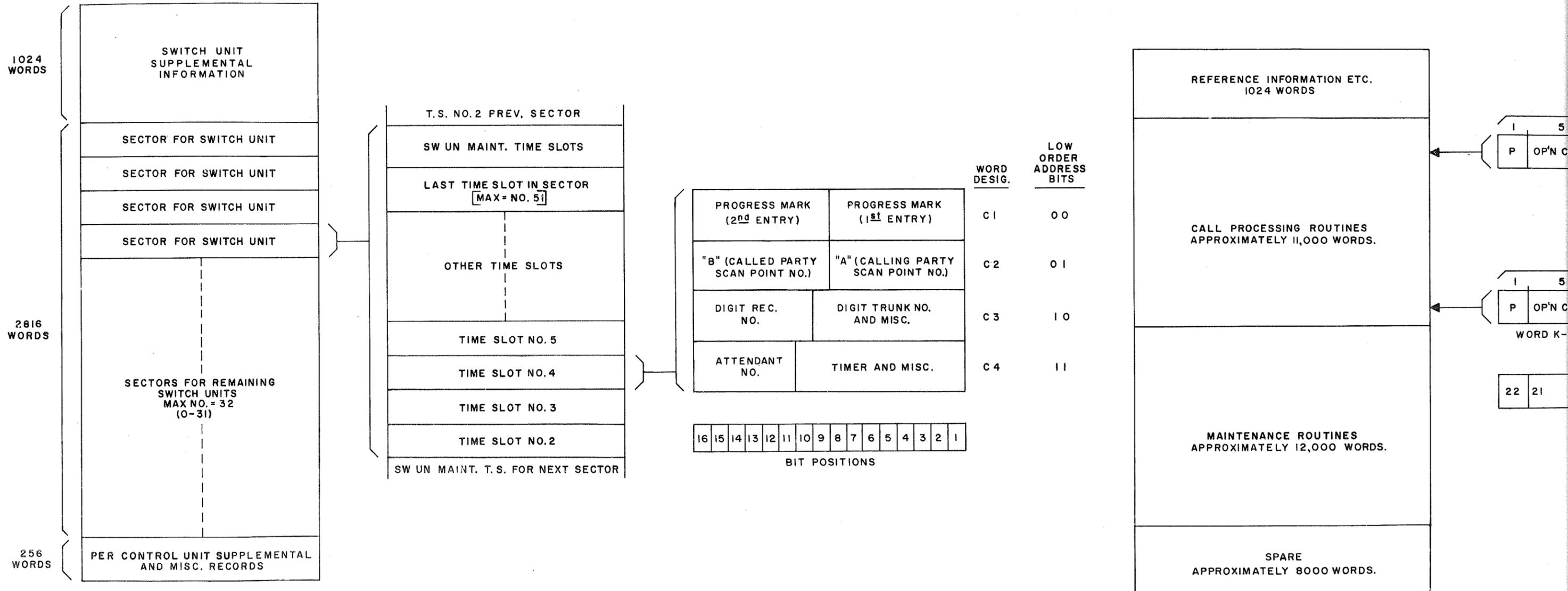


Fig. 1

"clear the input buffer." For commands of this type, the address bits are used with the operation code to help identify the command.

### 3. FUNDAMENTAL CIRCUIT FEATURES OF THE CALL PROCESSOR (Fig. 2)

#### A. General

**3.01** Fig. 2 shows how the call and program stores are connected to some of the more important registers, gating paths, and logic circuits.

#### B. Registers

**3.02** It will be noted that an address register has been connected to each store. The address registers may also be used as counters so that the words in the store can be addressed sequentially by adding a one to their contents. In the case of the call store address register (CA), provision has also been made to address every fourth word in sequence by adding ones to the ten highest order bits.

**3.03** An output register has been provided to receive the words which have been read out of the store. The output side of the call store output register (CO) connects through gating paths to many other registers and buffers, some of which are shown in Fig. 2. The input side of the CO also connects to many other registers. In addition, the CO can be used as a counter and as a high speed shift register.

**3.04** Between the program store and its output register is an anticipation register which works with the ready command logic in providing a smooth flow of commands into the remainder of the call processor.

#### C. Connections to the Input-Output Area

**3.05** As can be seen on the left-hand side in Fig. 2, address information from the switch unit buffer (SU) and the CO is made available to various parts of the input-output area. In addition, problems in the form of data messages to the call processor are gated from the input-output area into the incoming message buffer (MB) during the execution of the STT (start) command. Through the use of the comparator, the contents of the MB can be compared

with those of the CO. Also the contents of the MB can be gated into the CO by means of the RED MB command.

**3.06** The solutions to data problems are accumulated in the CO, the program store output register, the add shift counter, and the time slot counter, for gating into the input-output area by the LDT (load data) command. The gates for the LDT command are in the input-output area and are not shown in Fig. 2.

**3.07** Digits are gated into the CO by means of the RED DDR (read digits completed register) command.

**3.08** Six commands are allowed to interrupt the scan of the digit and data store. They are STT, LDT, RDT (request data trunk), LSD (load sender), RED DS (read data store), and WRT DS (write data store). The message transfer control logic is used to control the six interruption commands.

#### D. Command Translators

**3.09** As words are gated into the program store output register (PSOR), the high order bits are made available to the command translators. The translation of a command will activate a lead which will cause the command to be executed. After each command has been completed, another will be placed in the PSOR for translation and execution until an entire routine has been done.

#### E. Conditional Transfer Command Logic

**3.10** Thirteen of the commands are conditional transfers. That is, the execution of the command will result in a transfer to another program routine at some new address or location in the program store if a given system condition has been satisfied (for example, if certain bits in the MB match certain other bits in the CO). However, if the condition is not satisfied, the next command in sequence is executed rather than the one at the new or transfer address. All branches in the program coincide with conditional transfer commands.

#### F. Advance Command

**3.11** The appropriate call processing program routines are found by the advance command (ADV) with the assistance of the progress

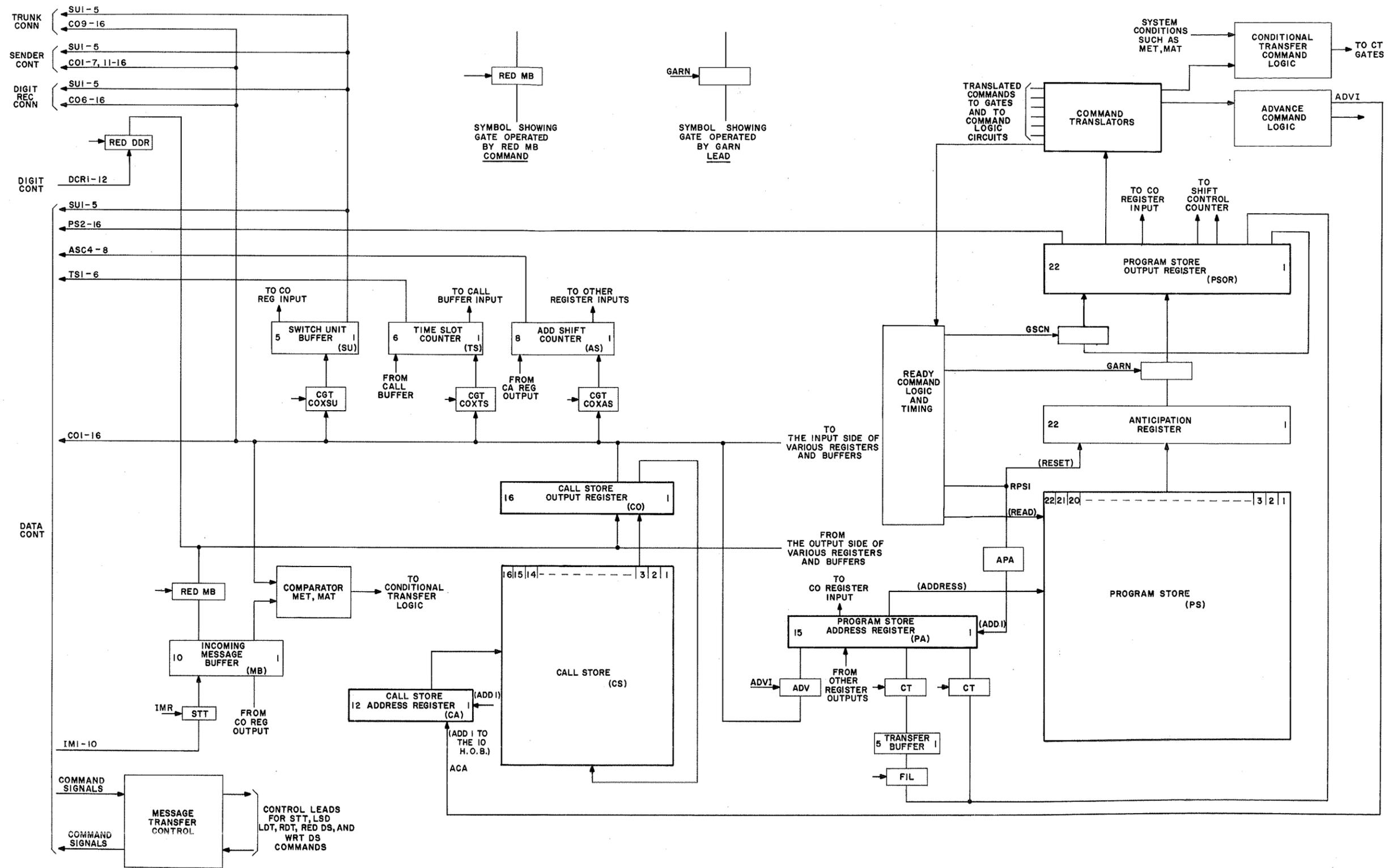


Fig. 2 - Call Processor, Fundamental Circuit Features

mark. If a progress mark is read out of the call store as a result of an advance command, it will be gated from the CO to the program store address register (PA). The progress mark itself may indicate that there is a problem to be solved; for example, progress mark No. 1 means dial tone is required. The progress mark, therefore, is used to address the area in the store where the routines for solving the problem at hand may be found. It may be assumed that the routine will solve the problem. The last command in the routine is ADV. Ordinarily this will cause the call store address register to be advanced to the progress mark in the next time slot. In other words, the advance command causes the call and program stores to interact with each other. That is, when the ADV1 lead is energized, it will gate the progress mark from the CO into the PA. Some time after the program store has been read at the progress mark location, a new advance command will energize the ACA lead, advancing the CA to the next time slot progress mark. After the call store has been read, the new progress mark is ready to be gated into the PA.

#### 4. THE PROCESSING OF CALLS (Fig. 3)

**4.01** Fig. 3 shows the sequence of events which occurs while processing calls. Follow the line which weaves into and out of the numbered boxes representing call store time slots. The time slots shown for sector No. 30 all have "progress mark zero" written into the first word. (See typical time slot in Fig. 1.) The line leaves each time slot and enters a program area called basic call processing. This represents gating the progress mark into the PA and reading the program store at the progress mark address. At the program store address for progress mark zero, the single command ADV is found, rather than a sequence of commands. As may be seen in Fig. 3, the ADV command ordinarily directs the call store to be read in the next time slot. If the progress mark is again zero, another ADV command will result, and so on.

**4.02** If an ADV command is read out of the program store and the time slot number is 2 (last time slot), then the PA will be jam set to the first command in a program sequence called end of sector. Among many other commands encountered in this routine is CGT

COXSU (clear the switch unit buffer and then gate the contents of the CO into it). This updates the switch unit number to that of the next sector to be scanned.

**4.03** After the end of sector routine has been completed, a transfer command will cause part of the beginning of sector routine to be performed. The initial command is STT. As discussed in 3.08, the STT command interrupts the scan of the DDS. It then tries to discover if a new data message is available for the sector to be scanned (in this case No. 31). As we shall see when we study the STT command, if there is a new data message, it is immediately gated into the MB at this time. After doing some switch unit maintenance and completing the beginning of sector routine, the first time slot of the next sector is addressed (time slot No. 23 for switch unit No. 31). If a progress mark other than zero is found, it is used to address the appropriate routine in the program store. After the problem represented by the progress mark and the system conditions has been solved, an ADV command will cause the progress mark in the next time slot to be addressed. When the last time slot in the last sector has been addressed, the ADV command will again cause the end of sector routine to be jam set into the PA. However, because this routine will discover that the last sector for the machine has been examined, the transfer out of end of sector will be to end of scan routines rather than to beginning of sector. As can be seen in the figure, the end of scan consists mainly of maintenance routines. One of the last commands is FIN (end of scan) which informs the input-output area that a call store scan has been completed. The scan of the first sector is preceded by the beginning of sector and switch unit maintenance routines.

#### 5. DETAILED BLOCK DIAGRAM OF THE CALL PROCESSOR (Fig. 4)

**5.01** Fig. 4 shows in block form all of the details of the call processor. The main functions for all of the commands are also indicated. If call processing is to be well understood, it will be necessary to make frequent reference to this figure, memorize the structure of all commands and gating paths, or use other reference

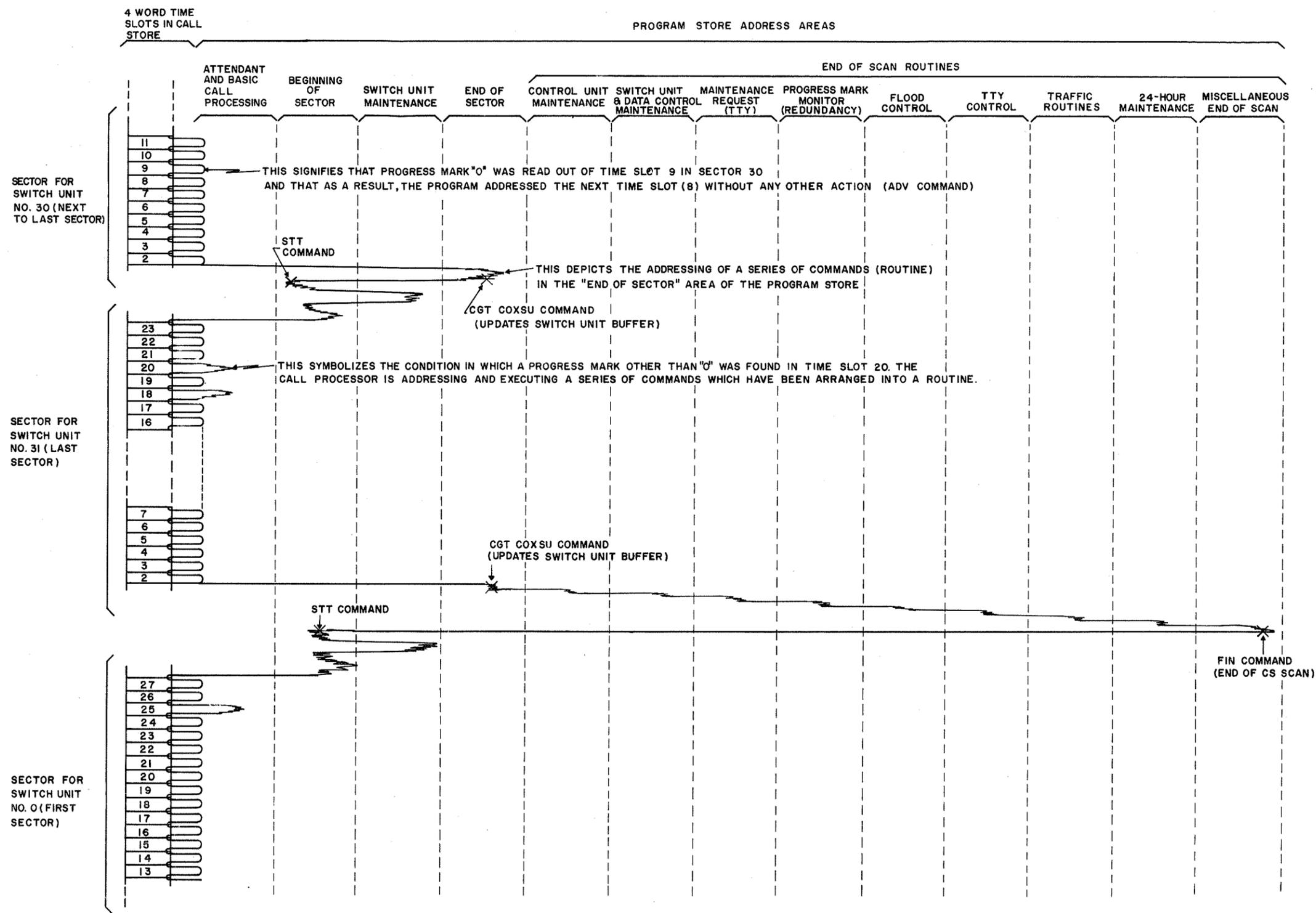


Fig. 3 - Visualization of Events While Processing Calls

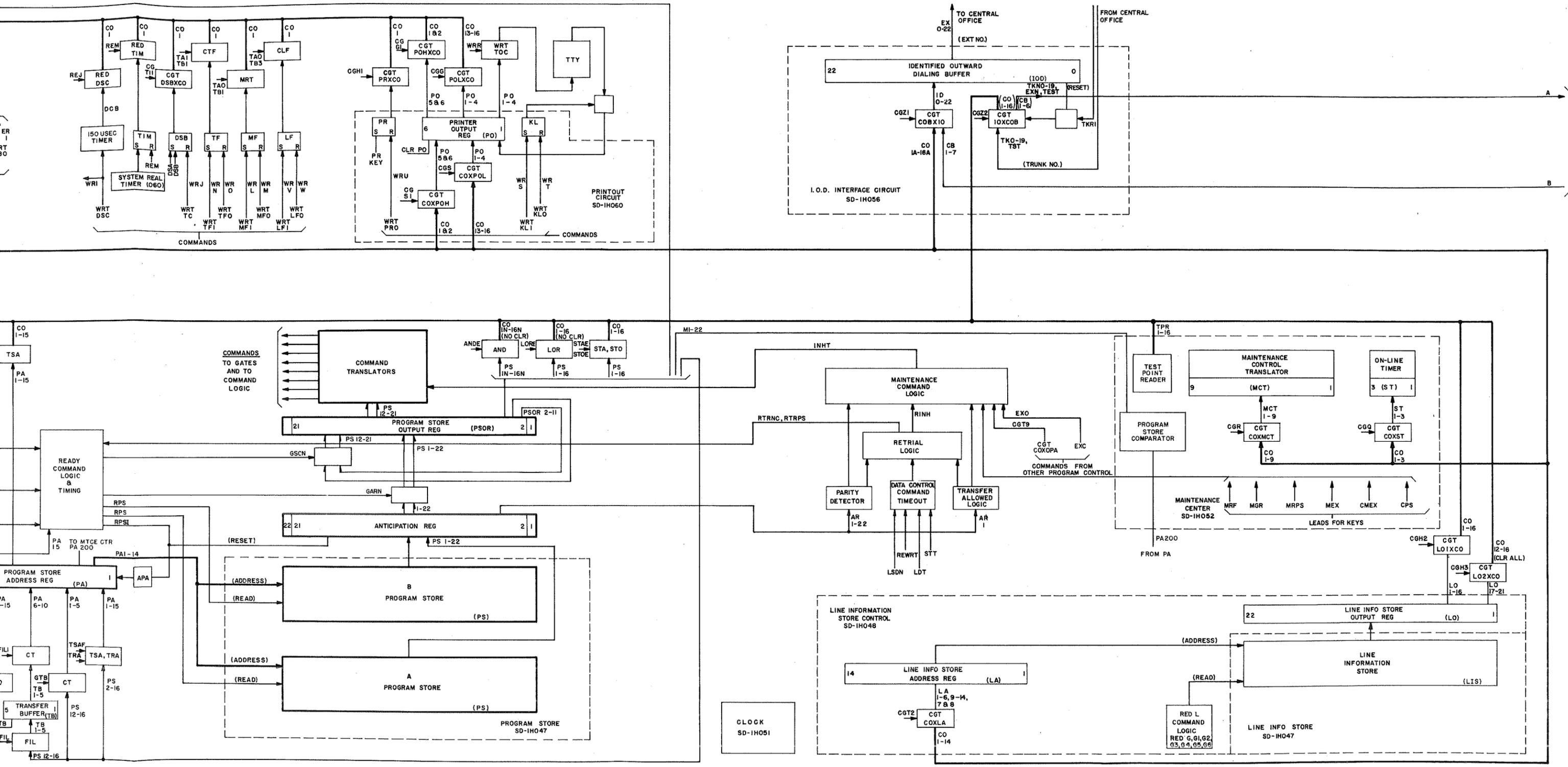
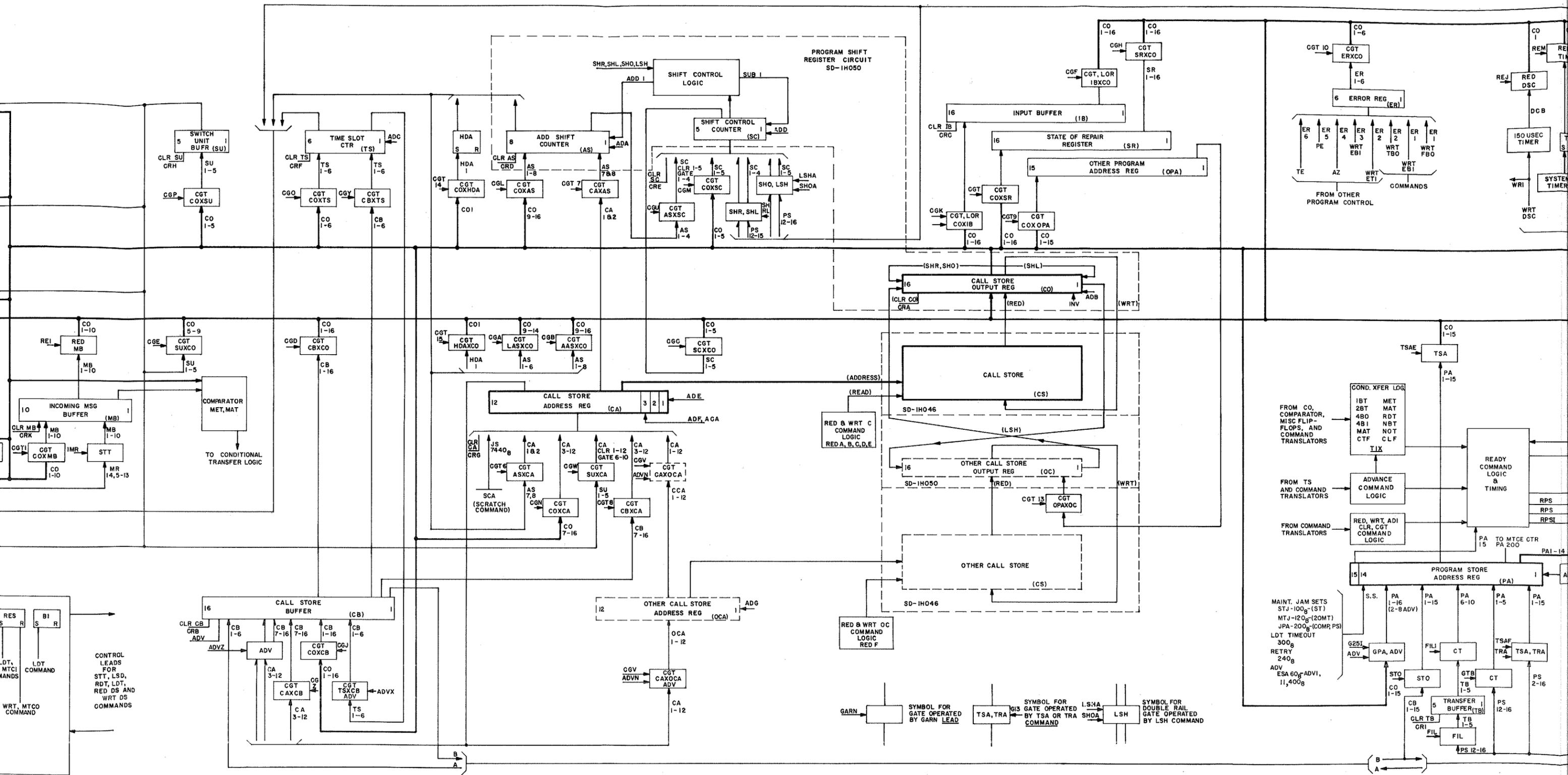


Fig. 4 - Detailed Block Diagram of Call Processor



GARN → SYMBOL FOR GATE OPERATED BY GARN LEAD  
 TSA, TRA → SYMBOL FOR GATE OPERATED BY TSA OR TRA COMMAND  
 LSHA → SYMBOL FOR DOUBLE RAIL GATE OPERATED BY LSH COMMAND  
 LSH → SYMBOL FOR DOUBLE RAIL GATE OPERATED BY LSH COMMAND

MAINT. JAM SETS  
 STJ-100<sub>g</sub> (ST)  
 MTJ-120<sub>g</sub> (20MT)  
 JPA-200<sub>g</sub> (COMP, PS)  
 LDT TIMEOUT  
 300<sub>g</sub>  
 RETRY  
 240<sub>g</sub>  
 ADV  
 ESA 60<sub>g</sub> ADV1,  
 11,400<sub>g</sub>

CONTROL LEADS FOR  
 STT, LSD, RDT, LDT,  
 RED DS AND WRT DS  
 COMMANDS

FROM CO, COMPARATOR,  
 MISC FLIP-FLOPS, AND  
 COMMAND TRANSLATORS  
 FROM TS AND COMMAND  
 TRANSLATORS  
 FROM COMMAND  
 TRANSLATORS



material such as CD-1H049. A short description of each buffer and counter found in Fig. 4 follows.

**5.02** The switch unit buffer (SU) is used to record the number of the switch unit or sector being scanned.

**5.03** The time slot counter (TS) records the number of the time slot under investigation during the sector scan. Because the higher numbered time slots are scanned first, the time slot counter has been wired to count backwards.

**5.04** The add shift counter (AS) is used as a general purpose counter and to record the number of bit positions shifted during the shift to 0 command.

**5.05** The shift control counter (SCC) stores the number of bit positions to be shifted. As each shift of one position takes place, the counter is advanced by one. When the counter reaches all zeros, the shifting stops. For this reason the shift control counter has been wired to count backwards.

**5.06** The input buffer (IB) is used to store information regarding the incoming data message which was previously gated into the MB. It is also used as a general purpose buffer.

**5.07** The state of repair register (SR) is used to record the condition of the control unit.

**5.08** The other program address register (OPA) has been shown on this diagram to illustrate the gating path into it from the online CO.

**5.09** The error register (ER) is used to store maintenance information for the maintenance program.

**5.10** The print-out register (PO) is used to store bits for the teletype machine.

**5.11** The identified outward dialing buffer (IOD) is used as an interface when gating an extension number to the central office for identified outward dialing.

**5.12** The incoming message buffer (MB) is used to store new data messages and as a general purpose storage location for words to be compared with the contents of the CO.

**5.13** The call store buffer (CB) is used as a general purpose buffer especially in connection with temporary storage of the contents of the CO.

**5.14** The transfer buffer (TB) is used with conditional transfer commands to extend their 5-bit addresses by an additional five bits. The transfer buffer is usually filled prior to the conditional transfer command by a FIL command.

**5.15** The line information store (LIS) together with its logic circuits and its address (LA) and output (LO) registers is used to store special service constants for individual switch units and lines, for example, compressed dialed numbers.

## 6. THE CLOCK CIRCUIT (Fig. 5)

**6.01** The system clock furnishes 4-phase timing pulses to the control unit. The 4-phase pulses are used in the call processor, and the 8-phase pulses are used in the input-output area. A single clock is used for timing both on- and off-line equipment. Reliability is dependent on the redundancy which has been built into the single clock circuit.

**6.02** The outputs of two identical 1-megacycle oscillators are combined on the input side of three shaping networks. The shaping networks produce 1-megacycle square waves. The square-wave frequency is divided by two in the three binary cells. The 1- $\mu$ sec pulses from the binary cells are used to generate the 4- and 8-phase signals in three modified gray code counters. The 4- and 8-phase signals are combined in 2-out-of-3 comparator circuits. The comparator produces an output which is similar to any two of its three inputs which agree with each other. A synchronizing circuit also compares the  $\phi 7$  outputs of the gray code counters in another 2-out-of-3 comparator and keeps all three clock chain circuits in step with one another through feedback connections to the input side of the binary cells and the counters. Failure of a 1-megacycle oscillator or of any output from the three clock chains will activate an alarm lead so that a lamp will light at the maintenance center.

ALARM LEADS TO MTCE, CTR

01, A02  
S1, OS2

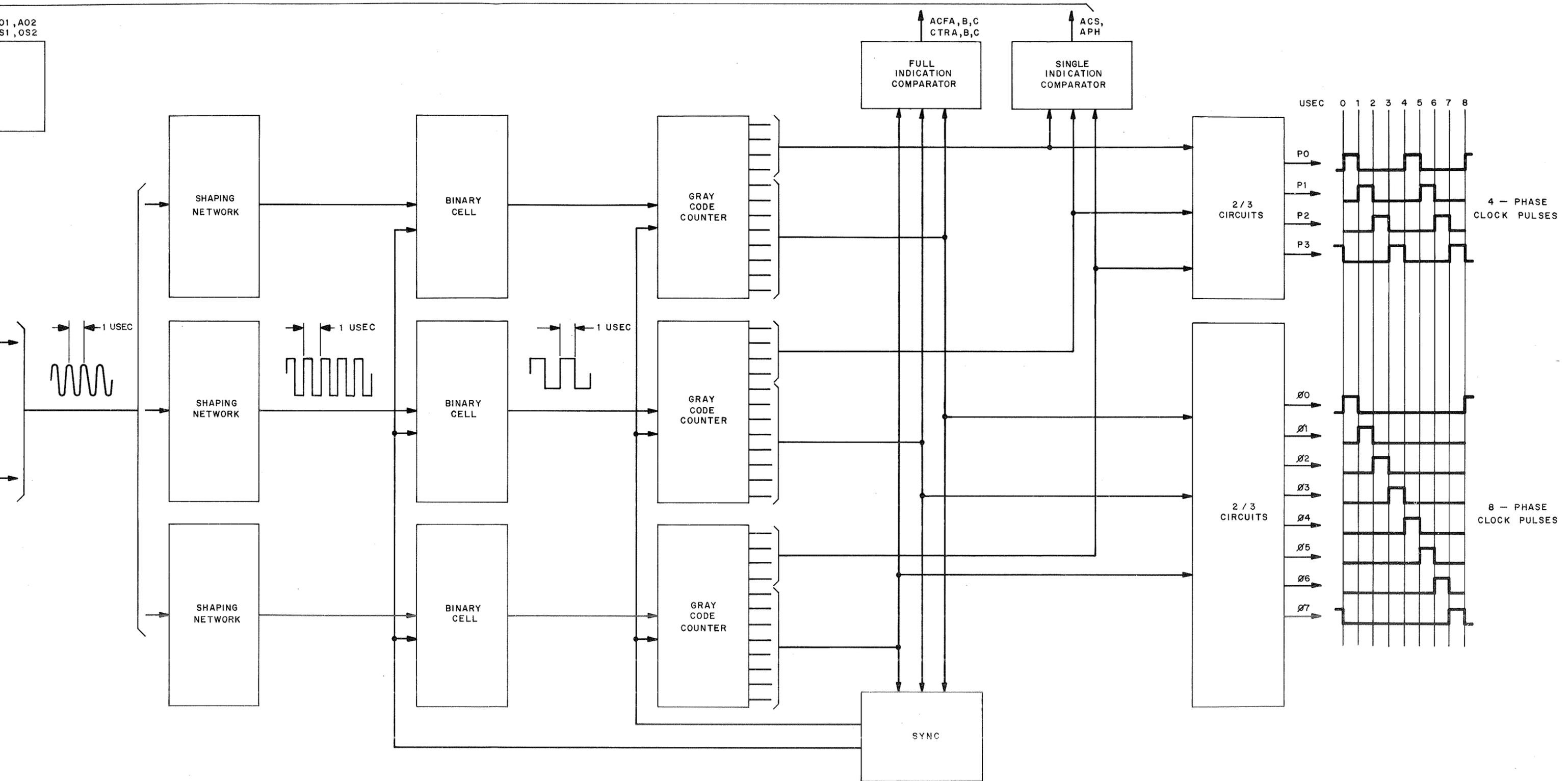
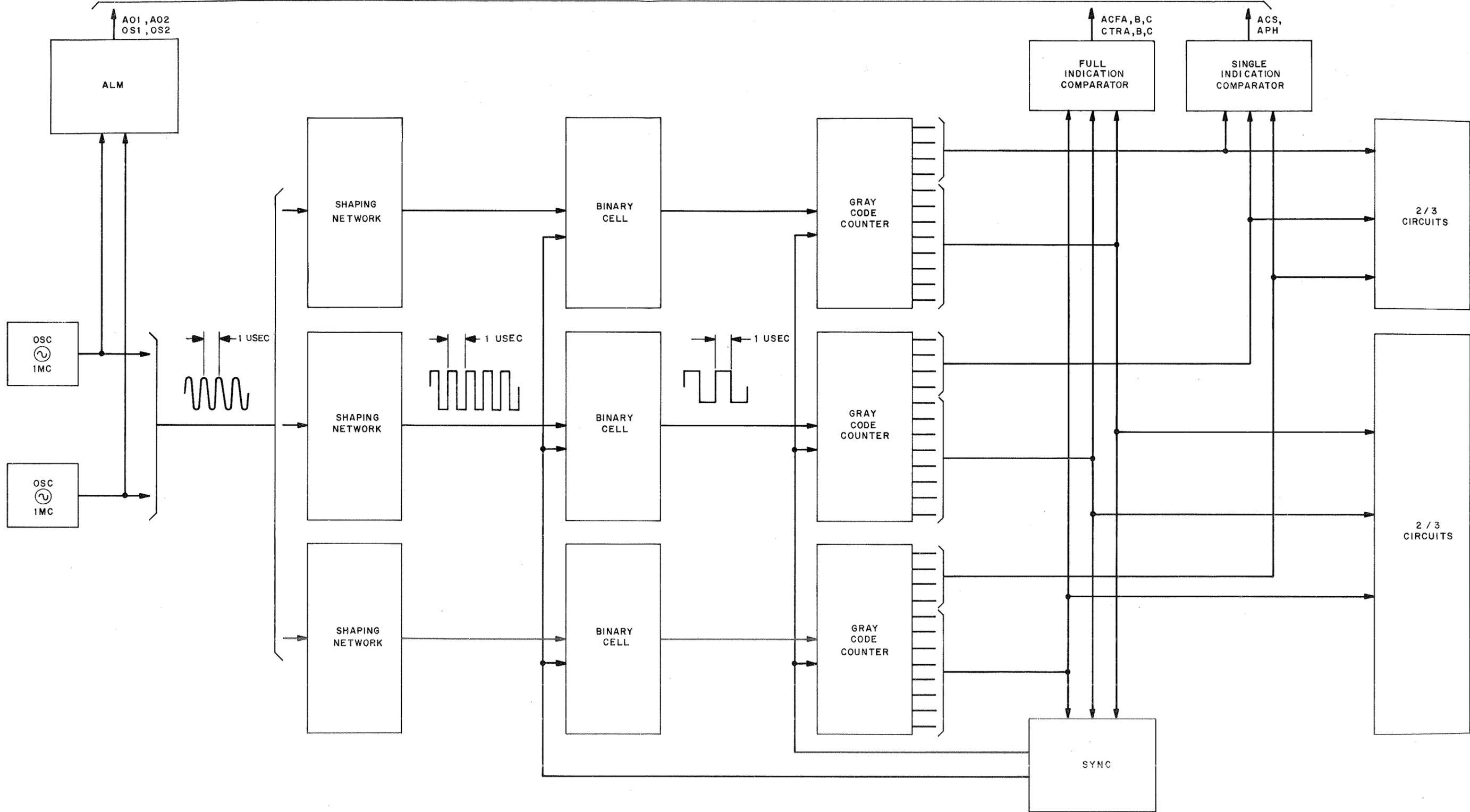


Fig. 5 - Clock Circuit

ALARM LEADS TO MTCE , CTR



## 7. READY COMMAND LOGIC AND TIMING (Fig. 6)

**7.01** The ready command logic insures a smooth flow of commands from the program store into the call processor. Timing is accomplished through the use of pulses from the 4-phase clock and by means of a special command timing circuit. The six outputs from the command timing circuits are shown in Fig. 6.

**7.02** In the ready command logic circuit the commands are divided into eight categories depending on whether they are J- or K-type and on how much time is required to complete the command. There are seven J commands: LDT, TRA, STO, STA, AND, LOR, and TSA. Note in Fig. 6 that the J commands together with the transfer type ADV and conditional transfer (CT) commands are not wired to set the RSC (ready second command) flip-flop. It should also be noted that the FSC (first or second command) flip-flop controls only the CT, AR, EX, and LS commands, all of which are K-type.

**7.03** If we assume that two 8- $\mu$ sec K commands are in the anticipation register and that the ready next command (RNC) flip-flop has been set, then at time PO the two commands will be gated into the PSOR. The command translators are disabled by the gate anticipation register (GAR) pulse at this time. At the beginning of time P1 the lead from the translator is activated and it sets the LS flip-flop and also initiates the command timing binary counters T1 and T2. TCA and TCB pulses are furnished to the LS gates. Because the first or second command (FSC) flip-flop was set by the GAR pulse, the RPS (read program store) flip-flop will be set by TCA through gate LSR and the appropriate program store module will be read into the anticipation register at the next PO. The condition of bit 15 in the PA will determine which program store module is to be read.

**7.04** The RSC flip-flop will be set by TCB through gate LSS. At the next PO, the bits in the right half of the PSOR will be double-rail gated to the left half. At P1 the command translator will no longer be inhibited by the GSC pulse, and the second command will set the LS flip-flop and initiate the command timer. (Please note that the GSC pulse has previously reset both the LS and the FSC flip-flops). Also,

the APA lead goes positive at P1, 5  $\mu$ sec after the anticipation register was cleared by RPSI. APA advances the PA by one so that the next command in the routine will be addressed. At the second P2 the RNC flip-flop is set by TCB through the gate LSN. GARN gates the next word into the PSOR. The two 8- $\mu$ sec commands are actually executed during the 7- $\mu$ sec period that their respective translator leads were activated, in other words, while the LS flip-flop was set. The gates for the other seven types of ready command logic operate in a manner similar to that for the LS gates.

## 8. The ADV (ADVANCE) COMMAND (Fig. 7)

**8.01** Fig. 7 shows the circuit detail for the ADV (advance command). If we assume that the advance command is in the left-hand side of the anticipation register and the RNC flip-flop has been set, then at time PO the lead GARN will go to ground and cause the contents of the anticipation register to be gated into the PSOR. Translation of the command will take place beginning at time P1. The bits in the time slot counter will now determine which leads will be activated.

**8.02** If the time slot number is even (0 in bit position 1), then lead CCB will go positive and clear the call buffer (CB).

**8.03** If the time slot number is odd, then the leads ADVX and ADVZ are activated. This will cause the contents of the time slot counter to be gated into bit positions 1 to 6 of the CB, and bits 3 to 12 of the CA will be gated into bit positions 7 to 16 of the CB. This information may be used if a conference call is initiated.

**8.04** Translation of the command will cause lead ESAR to go positive with TCE timing (see Fig. 6). Positive voltage on ESAR will clear the PA at time P2.

**8.05** If the time slot number is equal to or less than 2, the TS2 flip-flop will be set and the ESA lead will jam set the PA to 000,000,000,110,000 (60 octal). This will cause the end of sector routine to be performed as illustrated in Fig. 3.

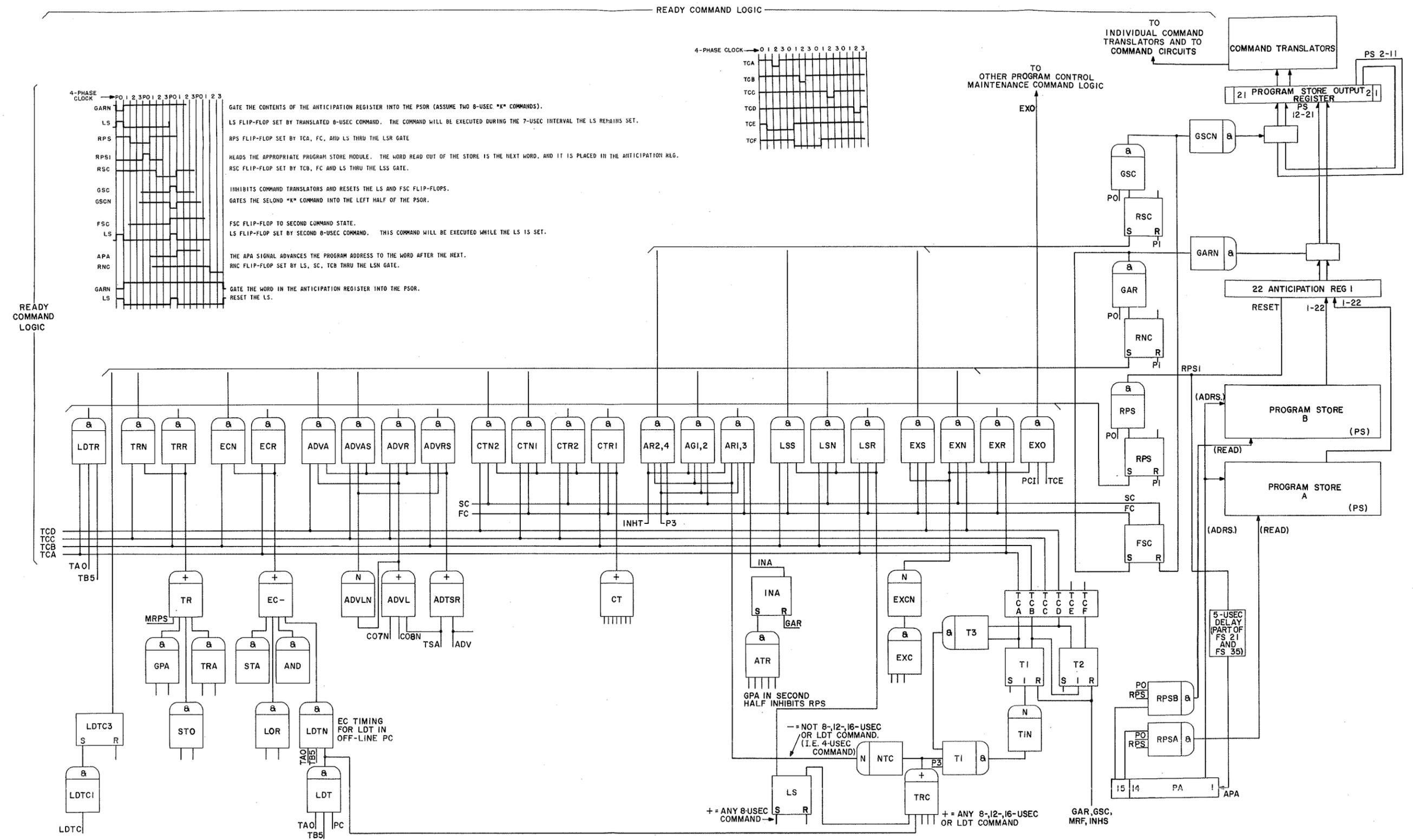
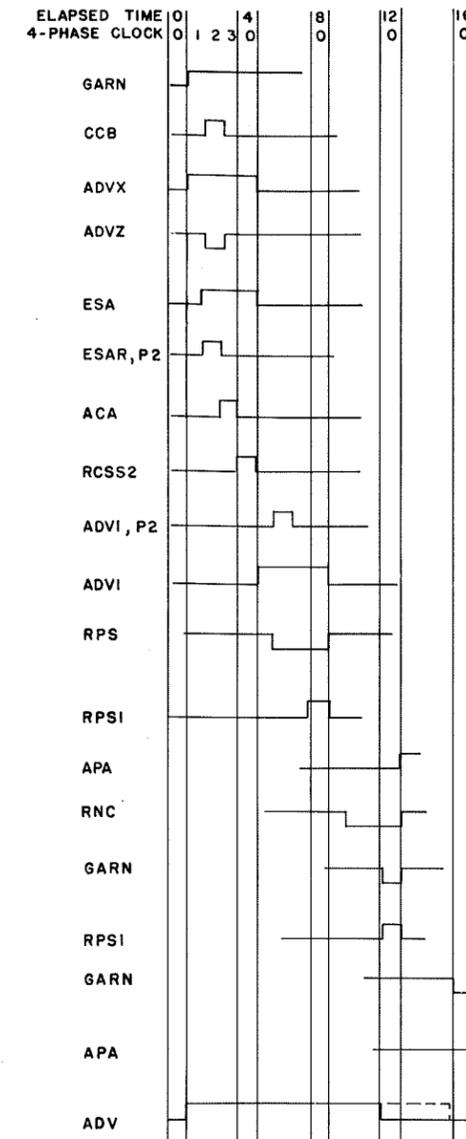
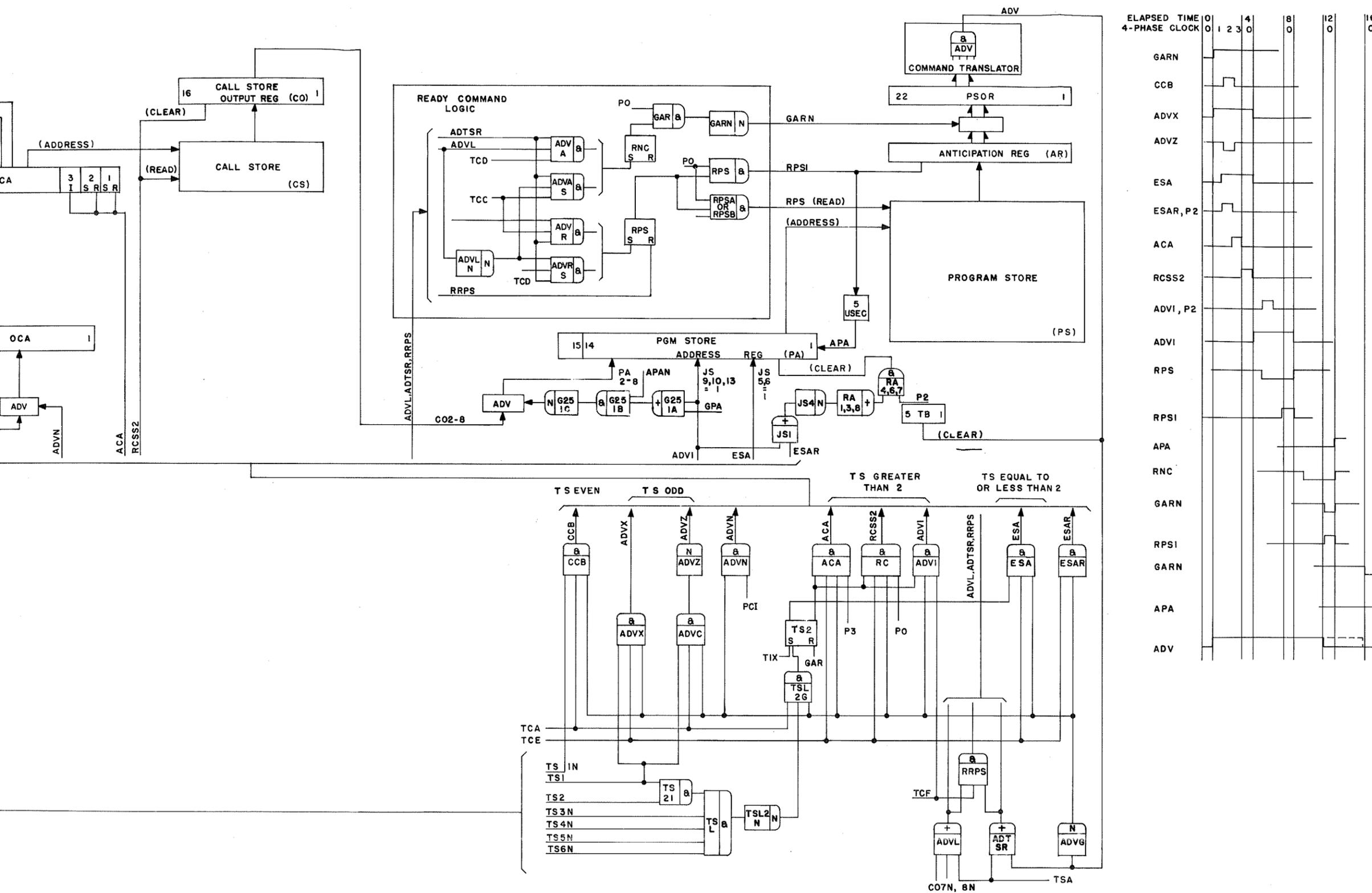


Fig. 6 - Ready Command Logic and Timing



GATE THE CONTENTS OF THE ANTICIPATION REGISTER TO THE PSOR (ADV IN LEFT HALF)

IF THE TIME SLOT NO IS EVEN, CLEAR THE CB.

IF THE TS IS EQUAL TO OR LESS THAN 2, JAMSET THE PA TO END OF SECTOR (60<sub>g</sub>).

RESET THE PA REGARDLESS OF TIME SLOT NO.

IF THE TIME SLOT NO IS GREATER THAN 2, CLEAR BITS 1 & 2 AND ADD 1 TO THE 10 HIGH ORDER BITS OF THE CA (ADDRESS NEXT PROGRESS MARK).

CLEAR THE CO AND READ THE CALL STORE.

SUBTRACT 1 FROM THE TIME SLOT COUNTER AND CLEAR THE PA.

GATE 2-8 OF THE CO (PROGRESS MARK) TO 2-8 OF THE PA AND JAMSET BITS 9-14 TO THE PROGRESS MARK AREA (11400<sub>g</sub>).

READY COMMAND LOGIC WAVE FORMS IF CO 7+8 = 0.

READY COMMAND LOGIC WAVE FORMS IF CO 7 OR 8 = 1.

CLEAR THE TRANSFER BUFFER.

Fig. 7 - Advance Command



**8.06** If the time slot number is greater than 2, the ACA, RCSS2, and ADVI leads will go positive. Voltage on the ACA lead will cause CA bits 1 and 2 to be set to 0 and a 1 to be added to the ten highest order bits of the CA. Thus, the ACA lead addresses the progress mark in the next time slot. Then the RCSS2 lead goes positive and reads the new progress mark into the CO. The ADVI voltage is timed by TCF and is therefore active during the second 4  $\mu$ sec of the command (see Fig. 6). At clock phase P2, ADVI subtracts 1 from the time slot counter contents. It also clears the PA and then jam sets the PA to 001,001,100,000,000 (11400 octal). At the same time, bits 2 through 8 of the CO (the progress mark) are gated to bit positions 2 through 8 of the PA. Thus, the appropriate progress mark routine is addressed.

**8.07** The only part of the ADV command which remains to be done is to read the program store at the progress mark address. However, since the quadrant bits (7 and 8) of the program store address require about 2  $\mu$ sec to settle before the store can be read, and since the PA was cleared at time P2, the store can only be read immediately if bits 7 and 8 of the PA are still 0. If a 1 has been gated into bit position 7 or 8, an additional 4- $\mu$ sec delay is provided by TCD rather than TCC timing in the ready command logic.

## 9. THE STT (START) COMMAND (Fig. 8)

**9.01** The STT (start) command is an interruption command which is used once during each beginning of sector routine to investigate the condition of the outgoing data trunk and to gate any new incoming data messages into the MB.

**9.02** Assuming that the STT command is in the left half of the anticipation register and that the RNC flip-flop has been set, the GARN lead will go to ground at time PO. The STT command will be translated beginning at time P1. It will set the STT flip-flop at time P2 if the scan interruption counter in the data control has not reached 15. If the scan interruption counter has reached 15, the entire call processor will wait for the interruption counter to be reset for the next DDS scan.

**9.03** After the STT flip-flop has been set, the CST5 flip-flop will be set by the next  $\phi 3$  of the 8-phase clock. This will occur either 1  $\mu$ sec or 5  $\mu$ sec after the STT flip-flop has been set, depending on the synchronization of the 8-phase clock with the STT command. The command start (CST) gates and flip-flops in the call processor will generate one 5- $\mu$ sec ground pulse for each interruption command. Battery on the STTR lead is used to reset the MB, the bad parity (BP) flip-flop, the start code (STC) flip-flop, the inhibit RDT (INRDT) flip-flop, and the outgoing trunk reserved (RES) flip-flop and to set the outgoing trunk busy idle (BI) flip-flop. The CST signal in the data control is used with the 8-phase clock to gate the contents of the address counter to the temporary register, clear the address counter, and gate the switch unit (sector) number into the address counter while setting AC bits 2, 8, and 9 equal to 1.

**9.04** This process addresses word 3 in the appropriate switch unit area of the data and digit store. When word 3 has been read, it is inspected for all zeros. All zeros will make the busy-idle information (IDL) lead go to ground, which will reset the outgoing data trunk busy idle flip-flop. During the scan of the sector, the reset busy idle flip-flop will serve to record that the outgoing data trunk is idle.

**9.05** The address counter and address register will now automatically address word 4 which is used to store new data messages from the switch unit. When word 4 has been read out of the digit and data store, bit 4 is examined by the gate IMR1. If bit 4 is a 1, then a data message is complete in the register and the incoming message ready (IMR) lead goes to ground. This sets the start code (STC) flip-flop and if bit 1 of the memory register equals 0, the bad parity flip-flop is set. In Fig. 4 it may be seen that the state of these two flip-flops is gated into the CO by the RED STC command. Therefore, it will be possible at any time while scanning the sector to find out whether a new data message is present and whether its parity is good.

**9.06** Ground on the IMR lead will also gate the memory register bits 14 through 5 into MB bit positions 1 through 10 respectively. This places the scan point number into the nine highest order bits of the MB and the on-off-hook

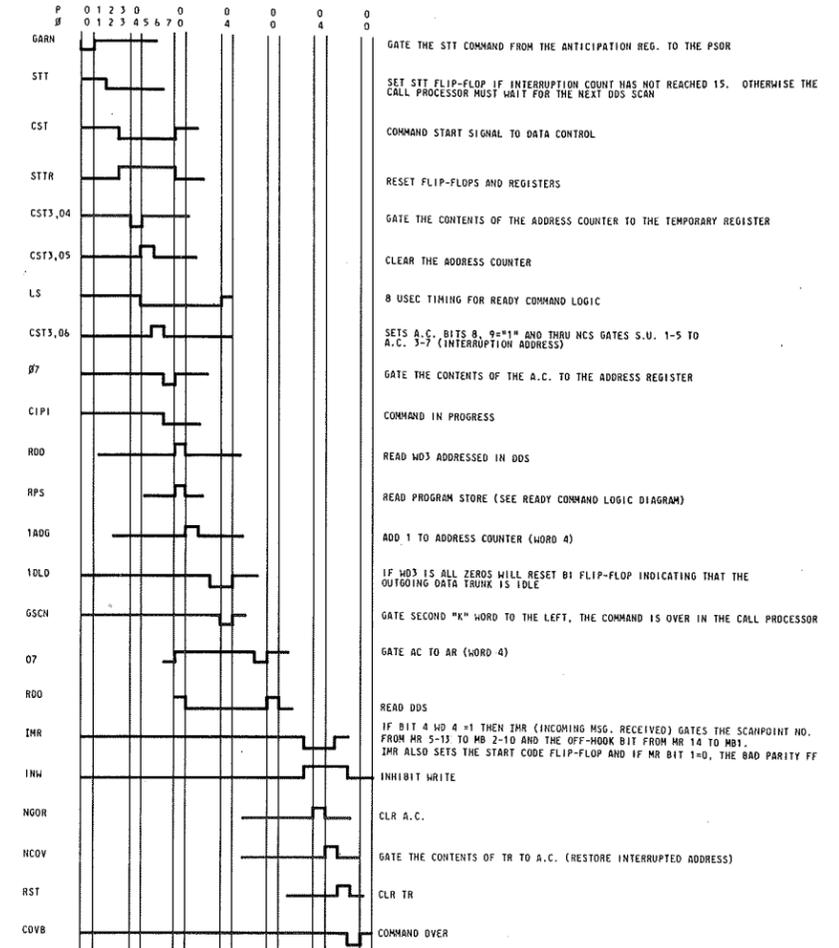
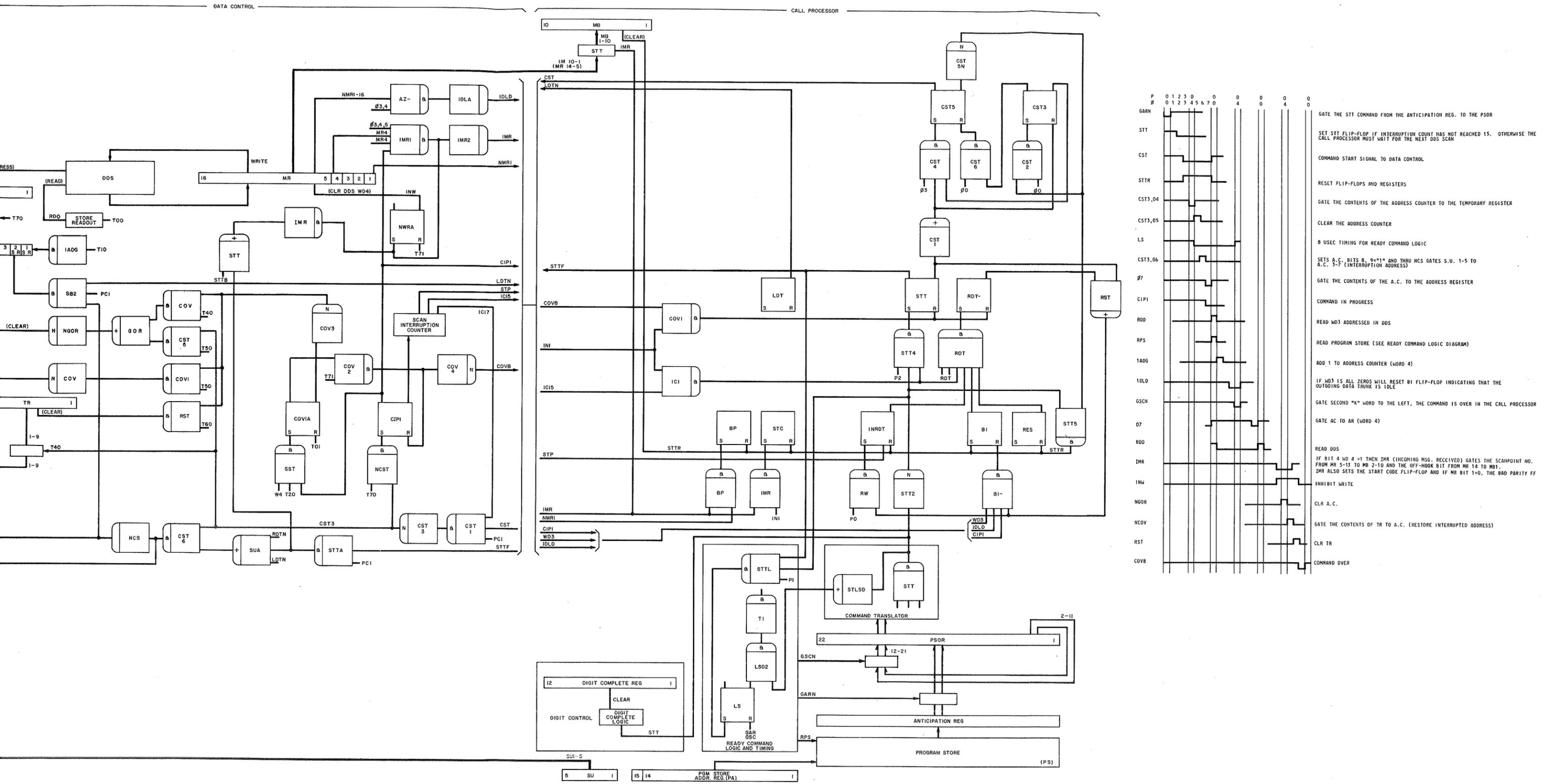
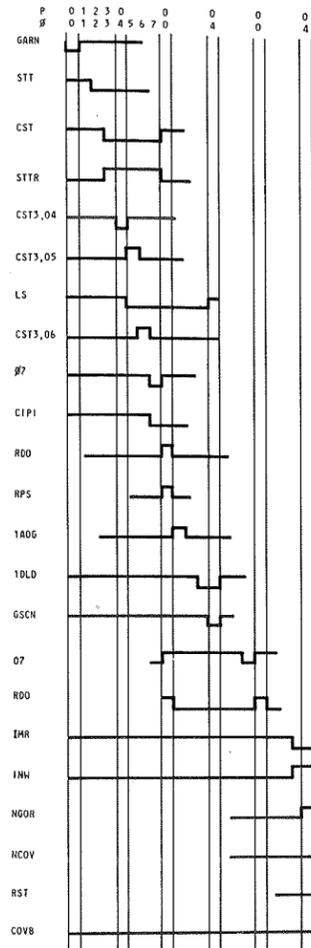
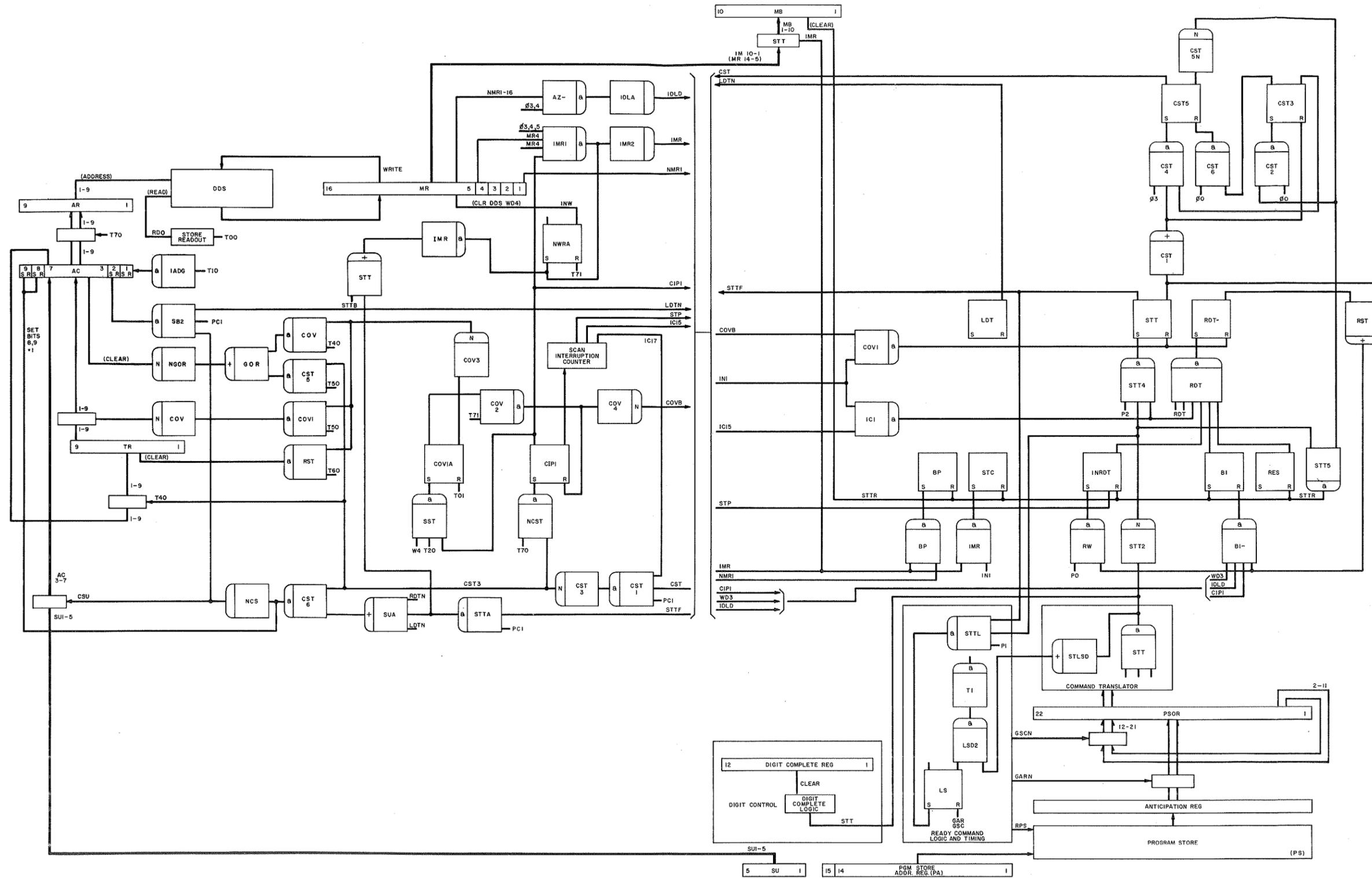


Fig. 8 - STT Command



bit into bit position 1. Word 4 of the DDS is cleared at this time through the inhibit write mode of the memory register.

**9.07** The COV1A flip-flop is set while word 4 is being addressed. As a result, the proper steps are taken to resume the digit and data store scan at the address stored in the temporary register. Ground on the COVB lead will reset the STT flip-flop; however, it should be noted that as far as the ready command logic is concerned, the command was over some time ago, and another command is already in progress.

**9.08** It may be noted that the STT command activates a lead in the digit control circuit. The digit complete logic will clear the digit completed register during the STT command if a RED DDR (read digit completed register) command was executed during the scan of the previous sector.

#### 10. THE LDT (LOAD DATA) COMMAND (Fig. 9)

**10.01** LDT is another of the six commands which will interrupt the scan of the digit and data store. The LDT1 flip-flop will be set at the first  $\phi 0$  after the command has been translated if the scan interruption count has not reached 15. If the interruption count has reached 15, all call processing halts until the scan interruption counter is reset. The CST signal is generated by  $\phi 3$  after the LDT1 flip-flop has been set. CST3 together with the 8-phase clock signals will gate the contents of the address counter into the temporary register, clear the address counter, and gate the contents of the switch unit buffer into it. Bits 8 and 9 of the address counter will be set equal to one. The above process addresses word one of the appropriate switch unit area of the digit and data store.

**10.02** The command in progress (CIP1) flip-flop will be set at the same time that the contents of the address counter are being gated into the address register. The appropriate bits are then gated in three word-times into the word 1, 2, and 3 areas of the digit and data store. During the third word-time, a signal is provided on the load data trunk completion (LDTC) lead to set the LDTC3, the busy-idle (BI), reserve (RES), and the ready next command (RNC) flip-flops; that is, the outgoing data trunk is

made busy and the next word will be gated out of the anticipation register at time PO.

**10.03** The COV1A flip-flop is also set during word 3. As a result, the address in the temporary register is restored to the address counter and the LDT1 flip-flop is reset.

#### 11. THE LSD (LOAD SENDER) COMMAND (Fig. 10)

**11.01** The LSD (load sender command) is used to interrupt the scan of the DDS so that the appropriate digit area may be loaded with compressed dialed numbers. However, before the LSD command is executed, the CO, the line information store output register (LO), the CB, and the add shift counter (AS) must be loaded with the compressed dialed digits which are normally stored in the line information store. In addition, the sender control register must be loaded with the digit receiver number and the central office trunk number through the WRT S command.

**11.02** The program which follows the command LSD is as follows:

WRT SCD — Gate the contents of the CO into the MR and then write into the DDS.

CGT LO1XCO — Clear the CO gate into it the 16 lower order bits of the LO.

WRT SCD — Gate the contents of the CO into the MR and then write into the DDS.

CGT CBXCO — Clear and gate contents of CB to CO.

WRT SCD — Gate the contents of the CO into the MR and then write into the DDS.

CGT AASXCO — Clear and gate all 8 bits of the AS to bits 9-16 of the CO.

WRT SCD — Gate the contents of the CO into the MR and then write into the DDS.

**11.03** In Fig. 10 it has been assumed that the 8-phase clock will set the LSD3 flip-flop 8  $\mu$ sec after the LSD command was gated into the PSOR. It has also been assumed that the scan interruption counter has not reached nine.

**11.04** After the LSD3 has been set, the LS flip-flop will be set at T-1 for timing the ready command logic. The CST signal will be

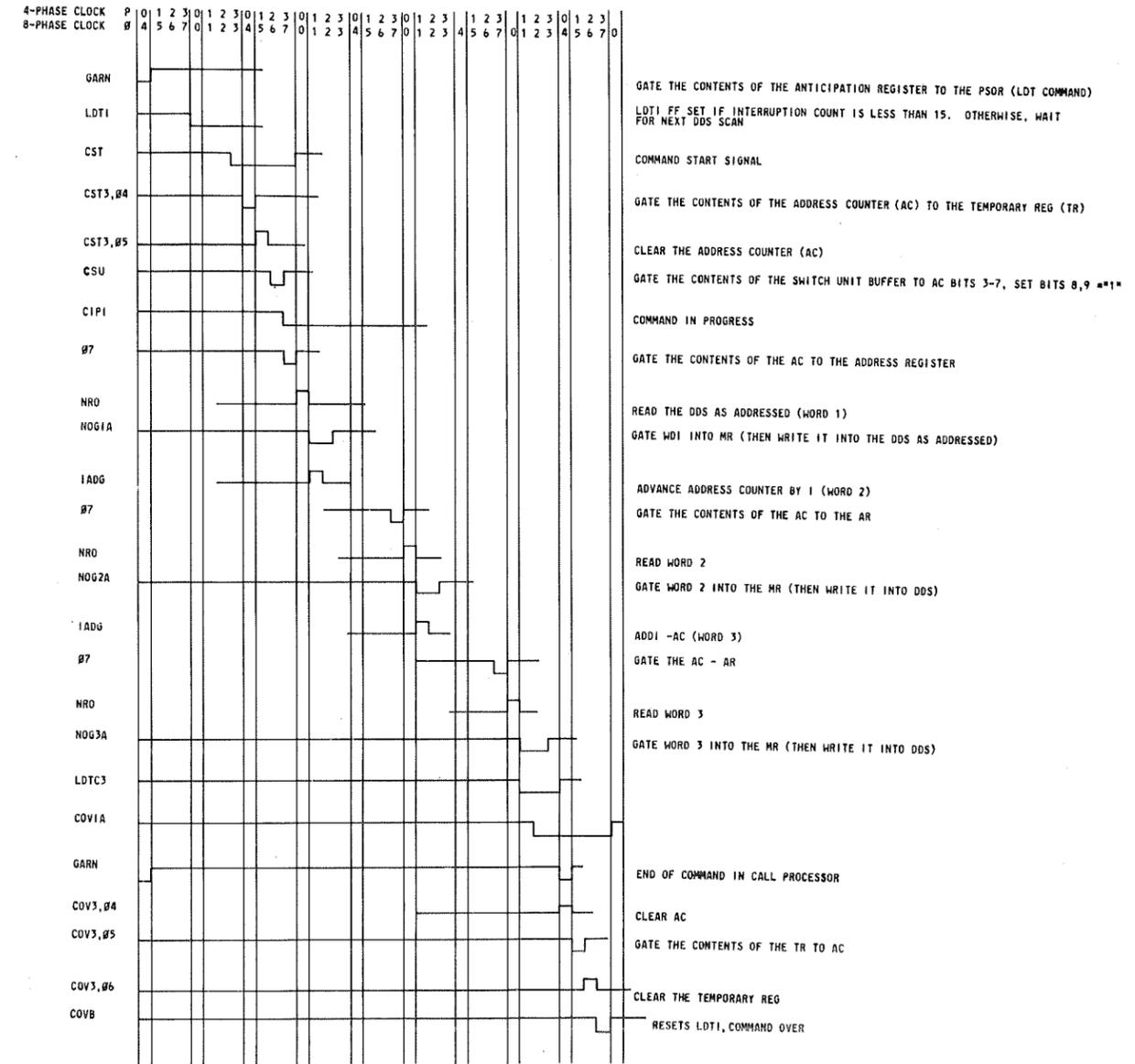
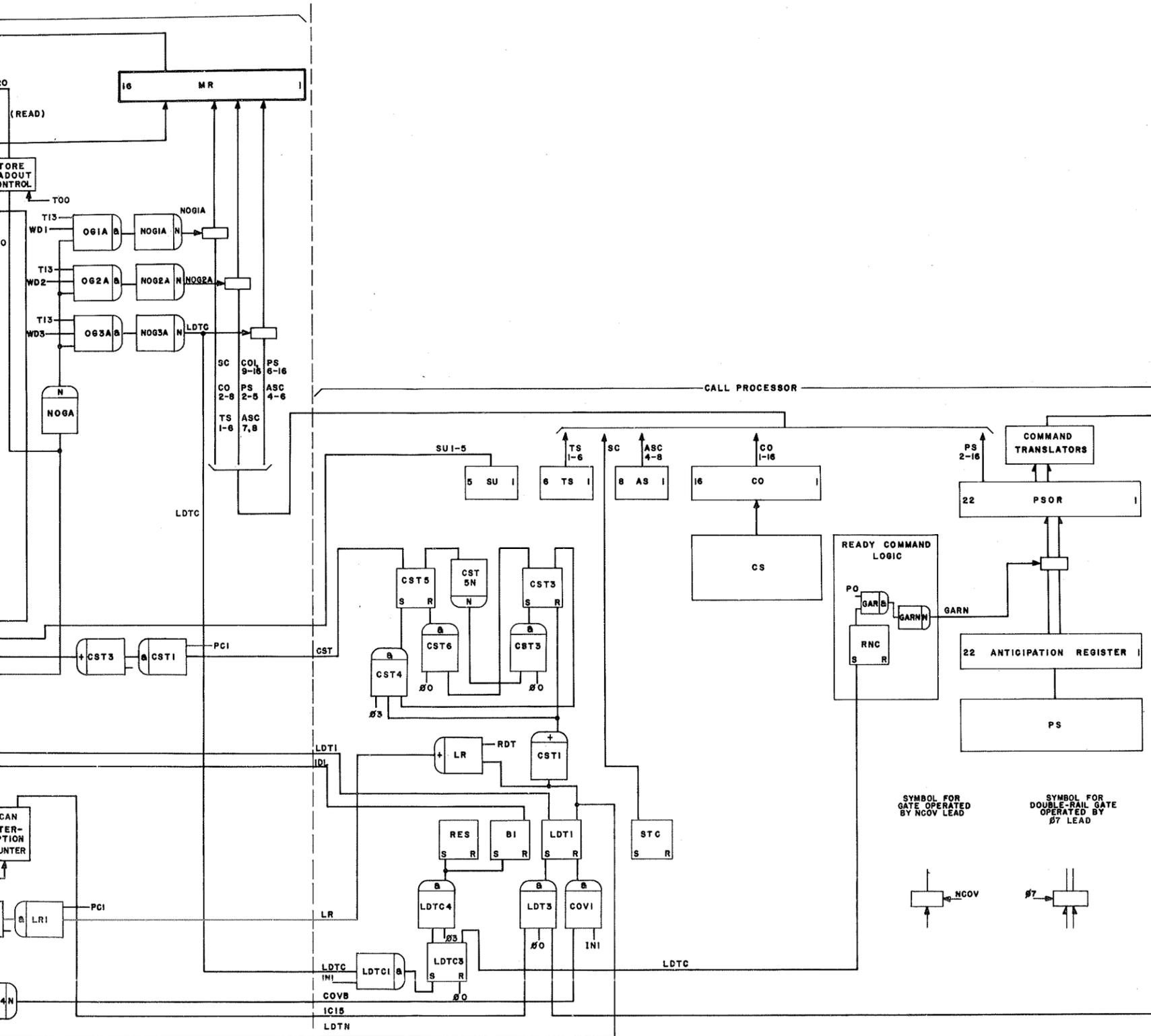
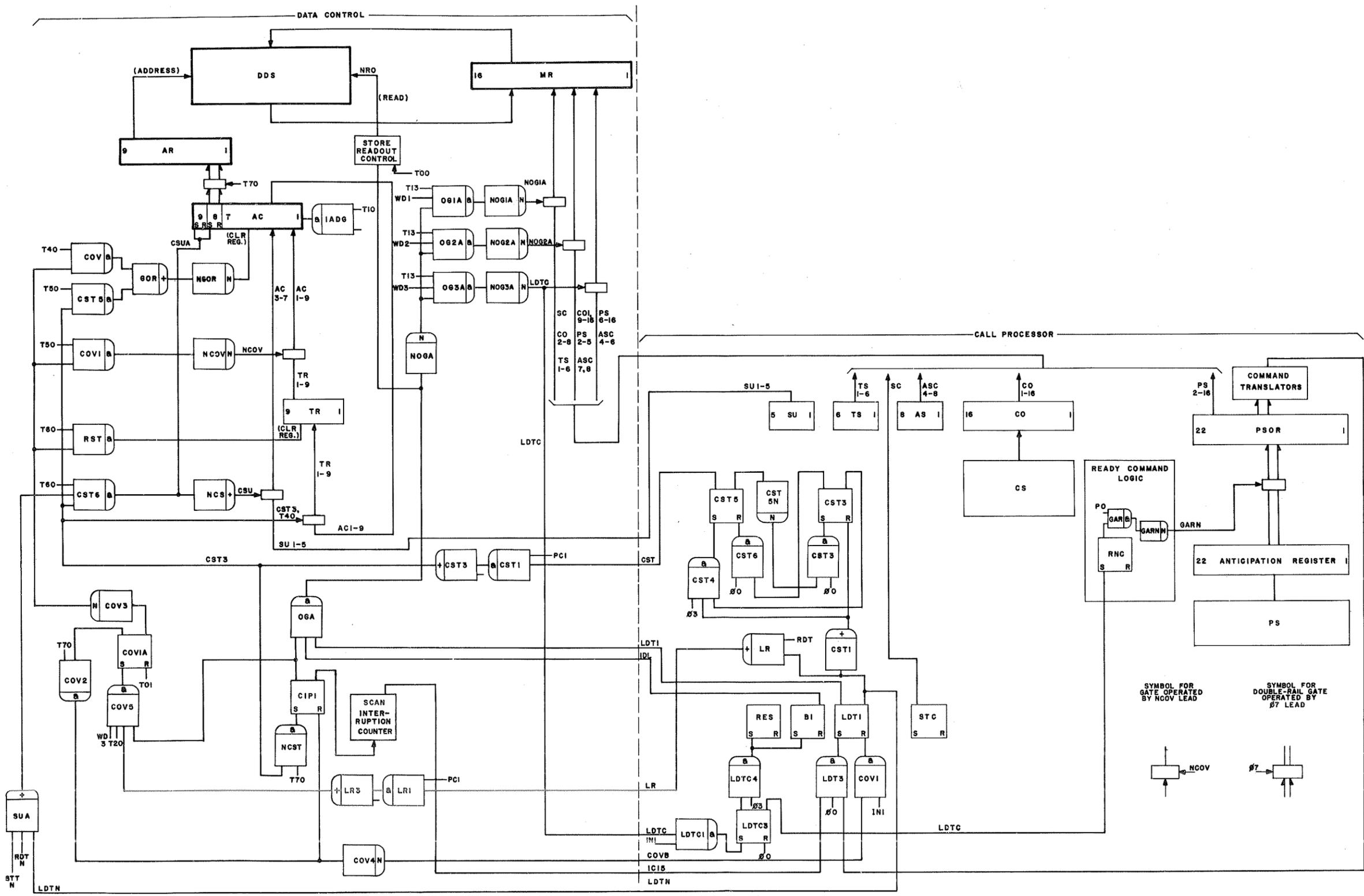
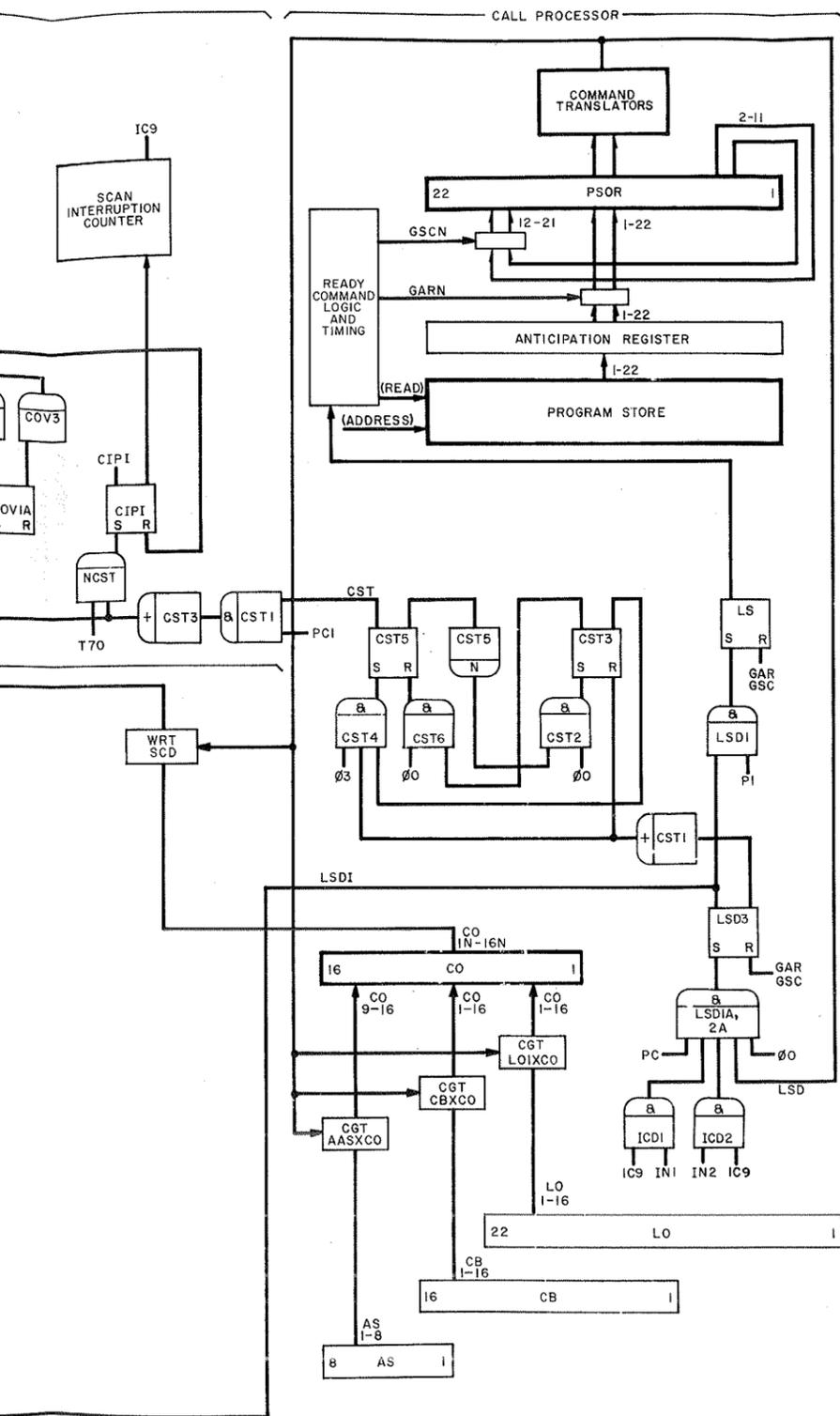


Fig. 9 - LDT Command



	4-PHASE CLOCK				8-PHASE CLOCK				
	P	0	1	2	3	0	1	2	3
GARN	0	1	2	3	0	1	2	3	0
LOT1	4	5	6	7	0	1	2	3	4
CST	0	1	2	3	0	1	2	3	0
CST3, $\beta 4$	4	5	6	7	0	1	2	3	4
CST3, $\beta 5$	0	1	2	3	0	1	2	3	0
CSU	4	5	6	7	0	1	2	3	4
CIPI	0	1	2	3	0	1	2	3	0
$\beta 7$	4	5	6	7	0	1	2	3	4
NRO	0	1	2	3	0	1	2	3	0
NOG1A	4	5	6	7	0	1	2	3	4
IADG	0	1	2	3	0	1	2	3	0
$\beta 7$	4	5	6	7	0	1	2	3	4
NRO	0	1	2	3	0	1	2	3	0
NOG2A	4	5	6	7	0	1	2	3	4
IADG	0	1	2	3	0	1	2	3	0
$\beta 7$	4	5	6	7	0	1	2	3	4
NRO	0	1	2	3	0	1	2	3	0
NOG3A	4	5	6	7	0	1	2	3	4
LDT3	0	1	2	3	0	1	2	3	0
COV1A	4	5	6	7	0	1	2	3	4
GARN	0	1	2	3	0	1	2	3	0
COV3, $\beta 4$	4	5	6	7	0	1	2	3	4
COV3, $\beta 5$	0	1	2	3	0	1	2	3	0
COV3, $\beta 6$	4	5	6	7	0	1	2	3	4
COV8	0	1	2	3	0	1	2	3	0



DDS  
WORD

WORD 5

WORD 6

WORD 7

WORD 8

WORD 1

WORD 3

WORD 4

ACTIVE  
COMMAND  
OR LEAD  
DESIGNATION

GARN

RPSI

LSD3

LS

CST

CST3, #4

NGOR

LSD, #6

#7

GSCN

WRT SCD

RNC

GARN

CGT LOIXCO

RPS, PSC

RPSI

GSCN

WRT SCD

RNC

GARN

CGT CBXCO

RPSI

GSCN

WRT SCD

GARN

CGT AASXCO

RPSI

GSCN

WRT SCD

IADG

#7

ASK

LSD1,2

#7

IADG

#PR1

#7

LDTR

COMP, LDTR

COV

B - PHASE CLOCK #

4 - PHASE CLOCK P

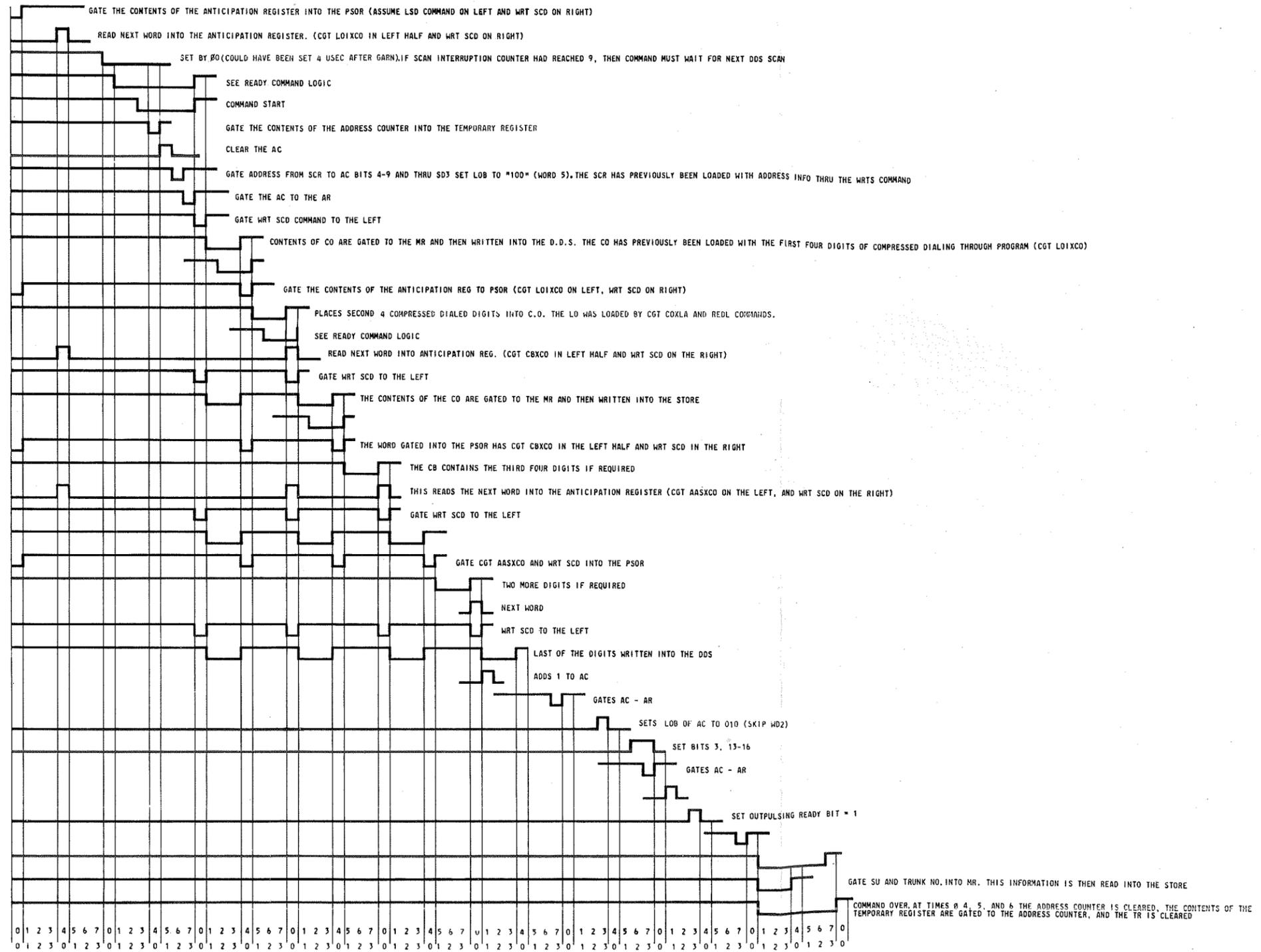
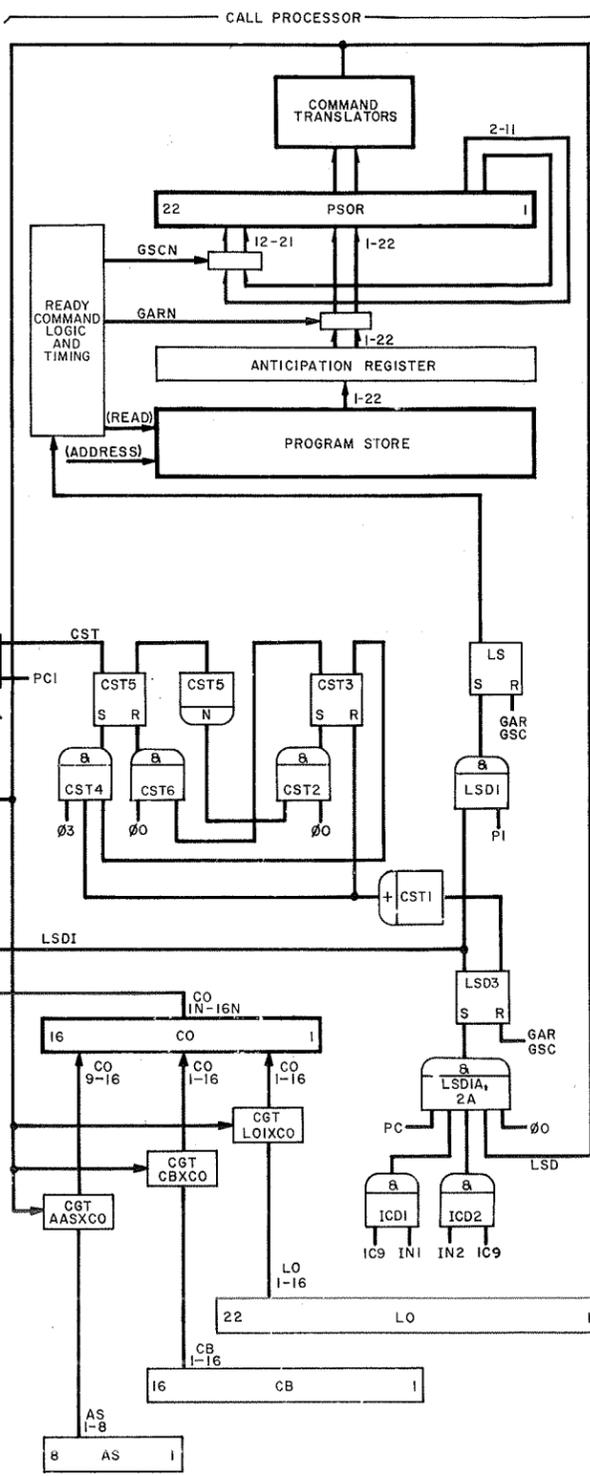
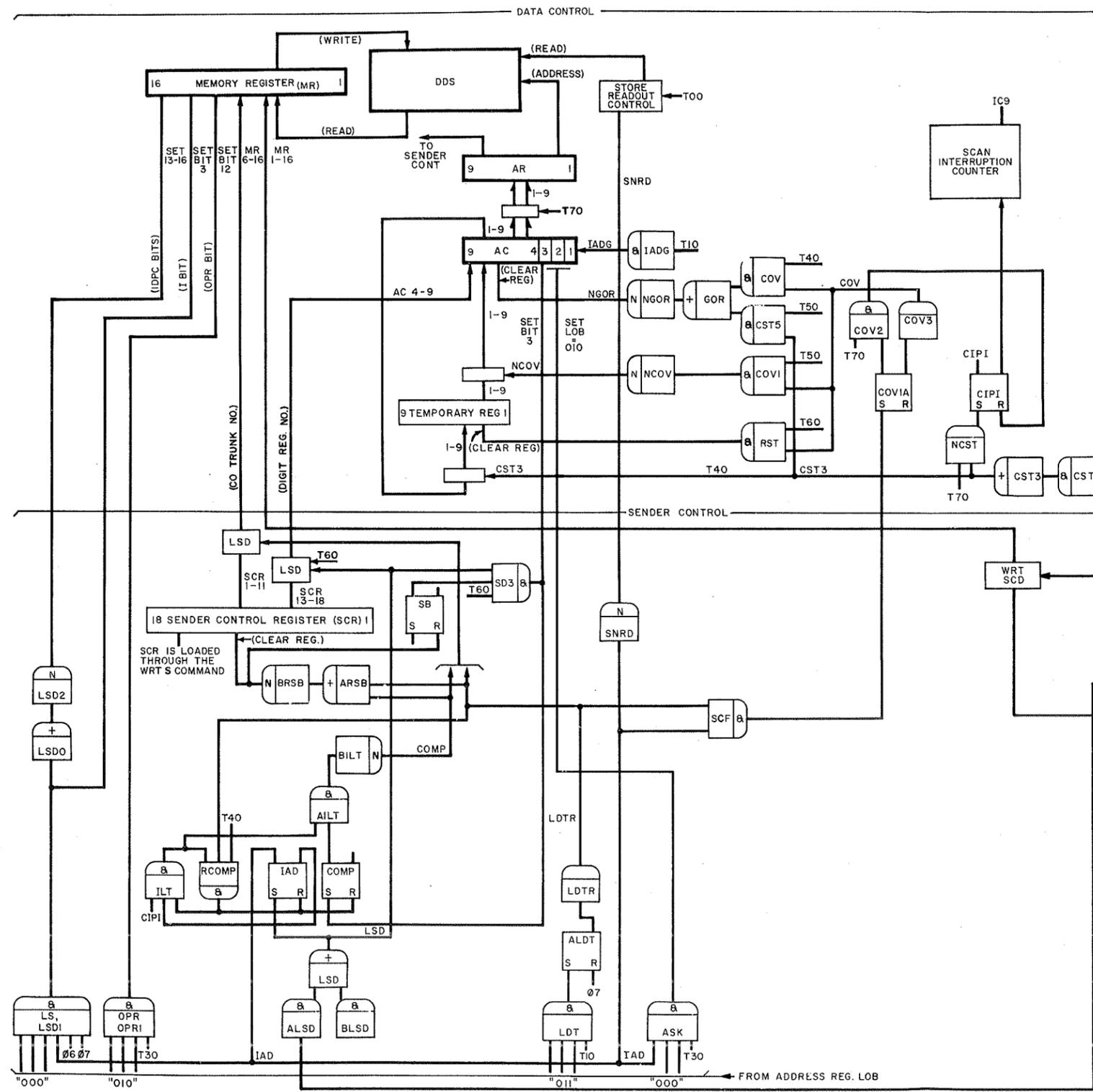
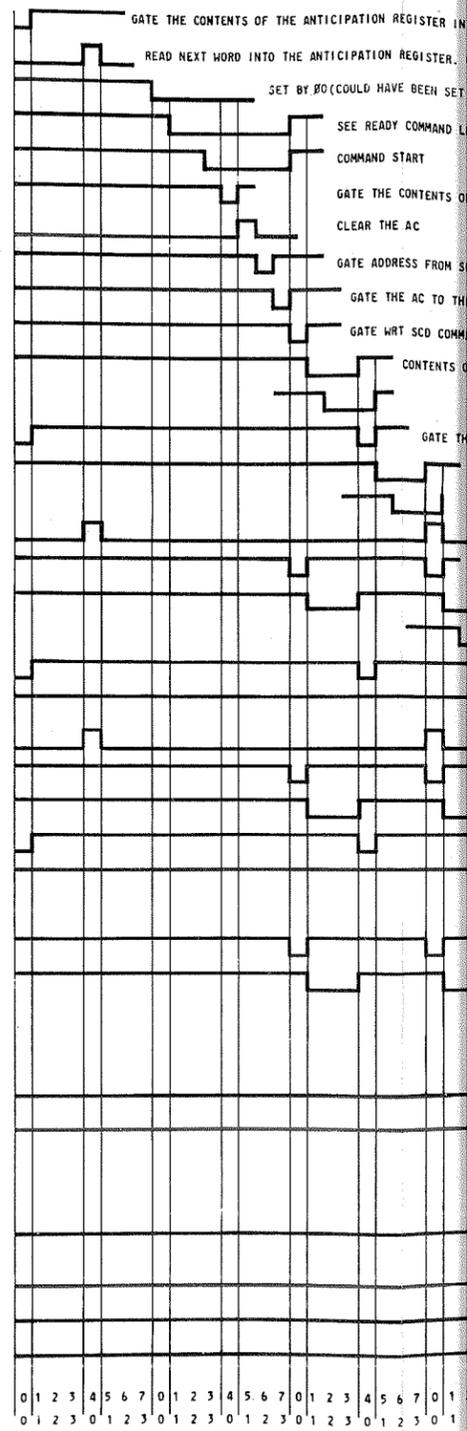


Fig. 10 - LSD Command



DDS WORD	ACTIVE COMMAND OR LEAD DESIGNATION
WORD 1	GARN
WORD 2	RPSI
WORD 3	LSD3
WORD 4	LS
WORD 5	CST
WORD 6	CST3, 04
WORD 7	NGOR
WORD 8	LSD, 06
WORD 9	07
WORD 10	GSCN
WORD 11	WRT SCD
WORD 12	RNC
WORD 13	GARN
WORD 14	CGT LOIXCO
WORD 15	RPS, RSC
WORD 16	RPSI
WORD 17	GSCN
WORD 18	WRT SCD
WORD 19	RNC
WORD 20	GARN
WORD 21	CGT CBXCO
WORD 22	RPSI
WORD 23	GSCN
WORD 24	WRT SCD
WORD 25	GARN
WORD 26	CGT AASXCO
WORD 27	RPSI
WORD 28	GSCN
WORD 29	WRT SCD
WORD 30	IADG
WORD 31	07
WORD 32	ASK
WORD 33	LSD1, 2
WORD 34	07
WORD 35	IADG
WORD 36	0PR1
WORD 37	07
WORD 38	LDTR
WORD 39	COMP, LDTR
WORD 40	COV



generated and the COMP and inhibit advance (IAD) flip-flops in the sender control circuit will be set.

**11.05** The CST3 signal will gate the contents of the address counter to the temporary register, clear the address counter, and set the CIP1 flip-flop. The LSD lead in the sender control will gate the digit receiver number from the sender control register into the address counter shortly after the address counter has been cleared. Bit 3 of the address counter is set equal to one at this time, and after the address has been gated into the address register, word 5 of the digit receiver area will be read. The LSD command is now over in the call processing area; however, the normal scan of the digit and data store will not be resumed until after several additional commands have been executed.

**11.06** The first of these is WRT SCD. This writes the first four compressed dialed digits into word 5 of the appropriate digit receiver area.

**11.07** The next command gates the contents of the LO into the CO. While this command is being executed, the next word in the digit receiver area of the DDS (word 6) is being addressed. At the command WRT SCD, the second four compressed dialed digits are written into the DDS.

**11.08** The remaining digits, if any, are then written into words 7 and 8 through a similar process, and then word one is automatically addressed and bits 3, 13, 14, 15, and 16 are set equal to one.

**11.09** The address skip word 2 (ASK) gate is wired so that word 2 is not addressed.

**11.10** When word 3 is read, bit 12 (the outpulsing ready bit) is set through the outpulsing ready (OPR) and (OPR1) gates. The LDT gate allows the central office trunk number to be gated into word 4 and, in addition, sets the COV1A flip-flop through the set command-over function (SCF) gate. Ground on the COV lead replaces the contents of the temporary register into the address counter. The normal scan of the DDS is then resumed.

## 12. THE PROGRAM SHIFT REGISTER CIRCUIT (Fig. 11)

### A. General

**12.01** Fig. 11 shows the main circuit features of the program shift register. The six commands which are of interest in this circuit are SHR (shift right), SHL (shift left), LSH (long shift), SH0 (shift to 0), INV (invert), and AD1-CO (add one to the CO).

**12.02** The call store output register (CO) is comprised of 16 binary cells with auxiliary steering networks which can be used to shift digits in either direction or between the on- and off-line shift registers.

### B. The Clock Switch Circuit

**12.03** The clock switch which has been shown in some detail in Fig. 11 utilizes low level logic. As we shall see, for any of the shift commands either the SHR or SHL flip-flop will be set. Thus for any shift command, one of the two input diodes for the gate D will be forward biased. In this event, the third diode will not conduct since both sides of it are at ground potential. The transistor will not conduct and its collector will be at a positive potential. However, with no shift command in progress both input diodes are nonconducting and the third or "shifter" diode is forward biased. This places a positive voltage on the base of the transistor, which makes the collector go to ground potential. Since all of the collectors in the clock switch are connected together, the 3-megacycle clock pulses CIN and CINO will not be transmitted to the clock signal switch amplifier (CSA) gate unless one of the shift commands is taking place.

**12.04** It should be noted that the polarities for low level logic (LLL) gates and flip-flops are the inverse of the polarities encountered in TRL logic. That is, to obtain a ground output the two AND inputs had to be positive battery. If either of the two OR inputs goes to ground, the output will be positive.

**12.05** To continue with the operation of the clock switch circuit, it is clear that the condition of the LSH2 flip-flop will determine whether the CIN or CINO clock pulses will be sent into the CSA gate. If the LSH2 flip-flop is set, the CINO pulses from the other program shift register will be used, and if it is reset the

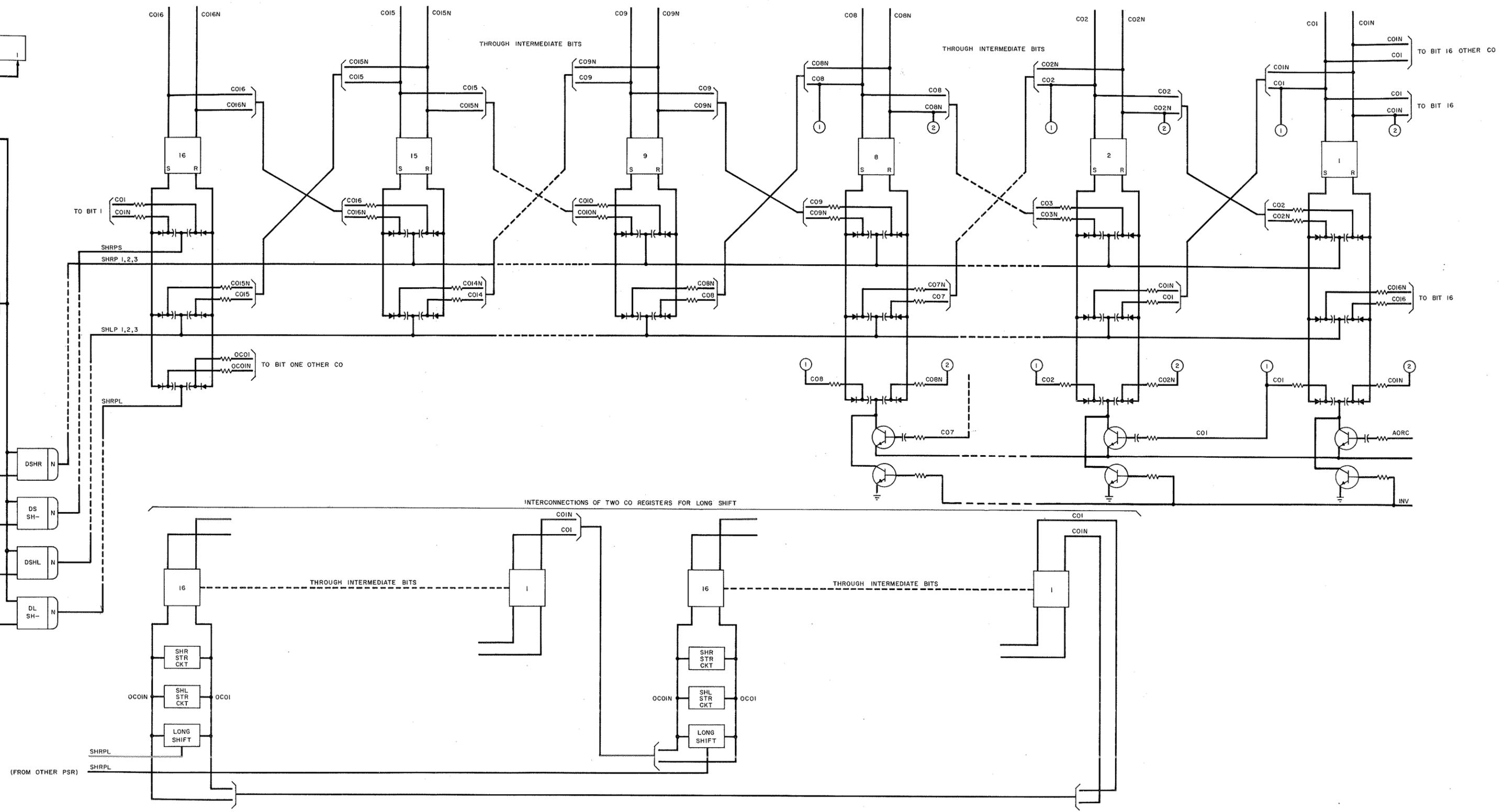


Fig. 11 - Program Shift Register



CIN pulses from its own program shift register will be used.

### C. Gating the Shift Command Addresses

**12.06** If one of the four shift commands is translated, either 4 or 5 bits will be gated from the address portion of the command into the shift control counter. The gating is double-rail and will take place during time P1.

**12.07** The address portion of the command may be all zeros. The command "shift right zero places" may not sound reasonable. Nevertheless, it is a valid command and it means shift right the number of places already stored in the shift control counter (SC). The PSAZ and PSBZ gates will inhibit the gating of an all-zero address into the SC. At the first P2 of the command (TCA timing), the appropriate flip-flops are set to enable the desired shifting mode.

### D. The SHR (Shift Right) Command

**12.08** For the command shift right the SHR and the shift right short (SHRS) flip-flops will be set. This turns on their associated drivers so that clock pulses through the CSA gate will kick the upper steering networks in Fig. 11. Shifting continues at the 3-megacycle clock rate until the SC has counted backwards down to zero. At this time a signal at the output side of G7 resets the SHR and SHRS flip-flops.

### E. The SHL (Shift Left) Command

**12.09** The SHL command operates in a similar way, but it uses another set of steering networks to accomplish the shift towards the higher order bits. Note that for all shift commands the bits shifted off one end of the register go back into the other end.

### F. The SH0 (Shift to Zero) Command

**12.10** The SH0 command shifts the bits in the CO to the right until a zero is found in bit position one, or until the shift control counter reaches all zeros. When the command shift to zero has been translated, the address bits are gated into the SC at time P1, and the SHR, SHRS, and the SH0 flip-flops are set by the TCA pulse. The shift will now take place until stopped by a signal from either gate G7 or G8. The number of places shifted will be recorded in the add shift counter (ASC) through the RCTL gates ASCC, 1ASC, and SASC.

### G. The LSH (Long Shift) Command

**12.11** When the LSH command has been translated, the address bits are gated to the SC, and the LSH1, SHR, and shift right long (SHRL) flip-flops are set. In addition, the LSH2 flip-flop in the other program shift register is set. This in turn will set the SHR and SHRL flip-flops in the other program shift register. Thus, the two 16-bit COs are connected as one 32-bit shift register, and the command LSH 16 will cause the contents of CO No. 1 to be transferred to CO No. 2 and those of CO No. 2 to be transferred to CO No. 1. Because of the state of the two LSH2 flip-flops, the control unit from which the command was issued will furnish a 3-megacycle clock signal to both COs during the LSH command.

### H. The AD1-CO (Add One to the CO) and the INV (Invert) Commands

**12.12** The eight low order bits of the CO can be used as a binary counter or as an inverter. An inverter will change all of the ones to zeros and all of the zeros to ones. The leads from the translators for the two commands AD1-CO and INV are shown on the right-hand side in Fig. 11 (leads AORC and INV). A third steering network is used on the eight low order bits of the CO to perform these two functions.

## 13. MAINTENANCE COMMAND LOGIC (Fig. 12)

### A. The EXC (Execute) Command

**13.01** The execute command (EXC) instructs the off-line call processor to perform the command which is present on the input of its command translator. The off-line translator is normally inhibited by the set inhibit (INH) flip-flop; therefore, the main function of the execute command is to reset the INH flip-flop long enough so that one command will be performed.

**13.02** Referring to the timing diagram in Fig. 12, it may be seen that an execute command in the on-line program control will set the EXC1 and EXC2 flip-flops in the off-line call processor. This causes battery to appear on the input side of the INHR gate which inhibits the INHS1 gate and enables the INHR1. At time P0 the INH flip-flop is reset causing the INHT lead to go to ground. This enables the command translators.

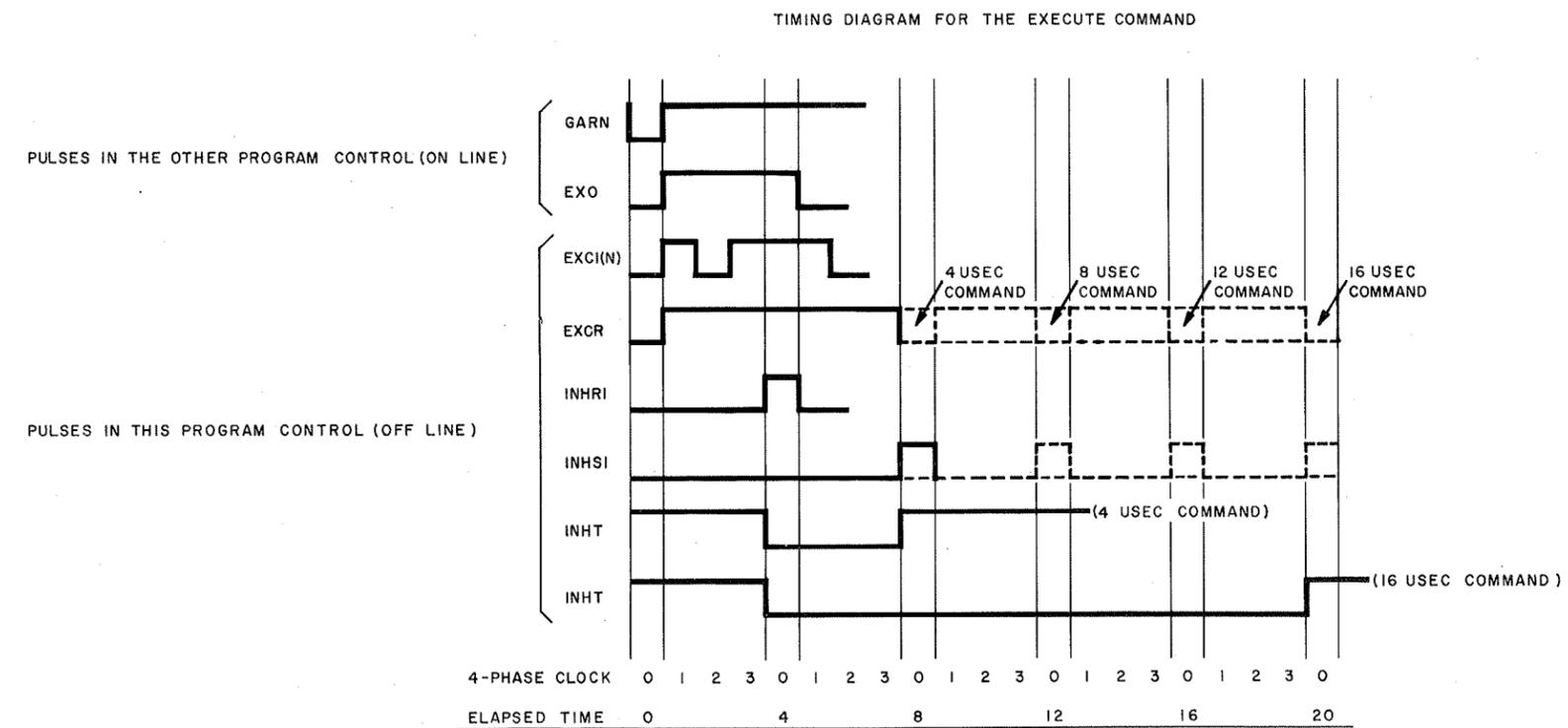
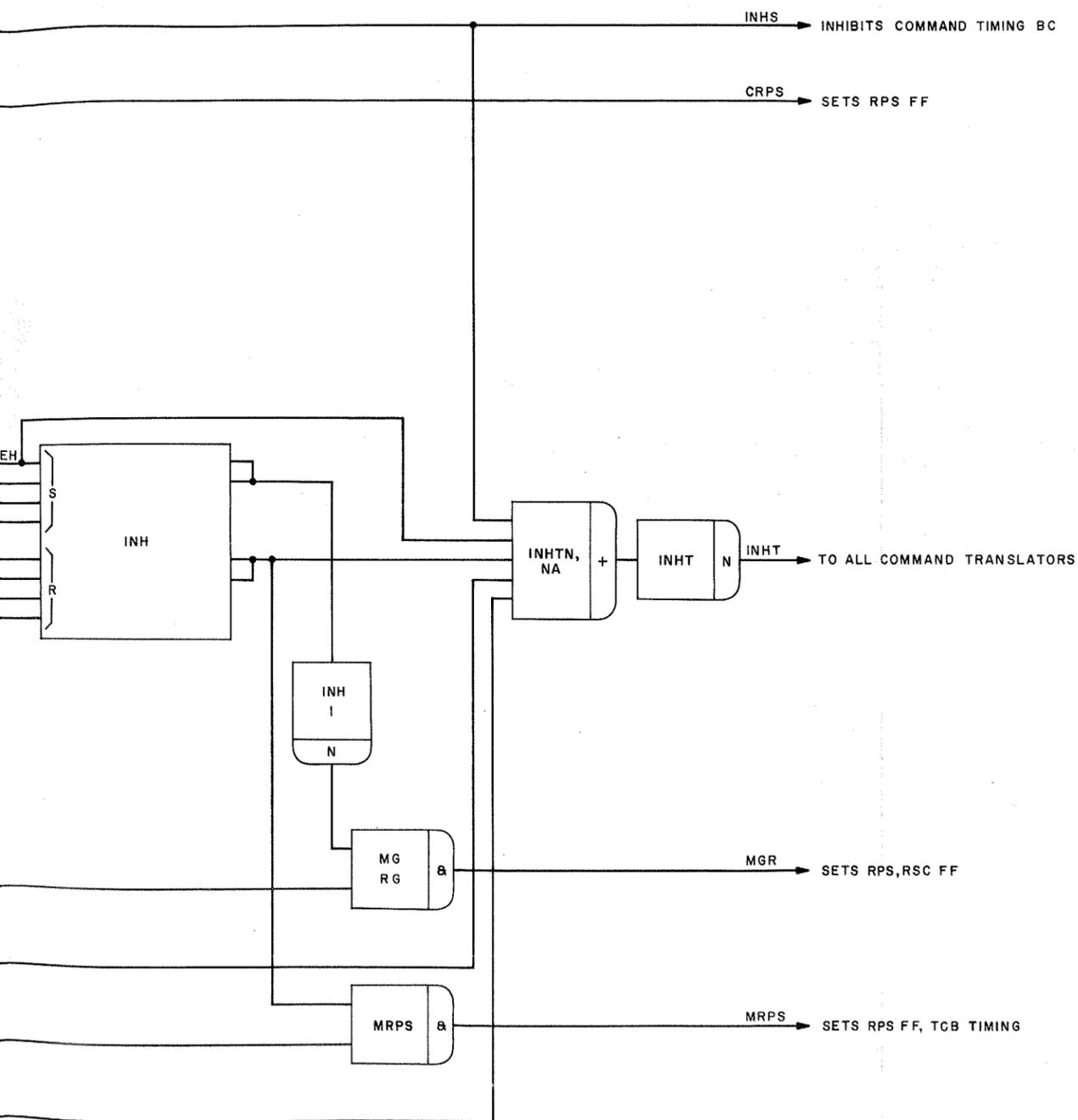
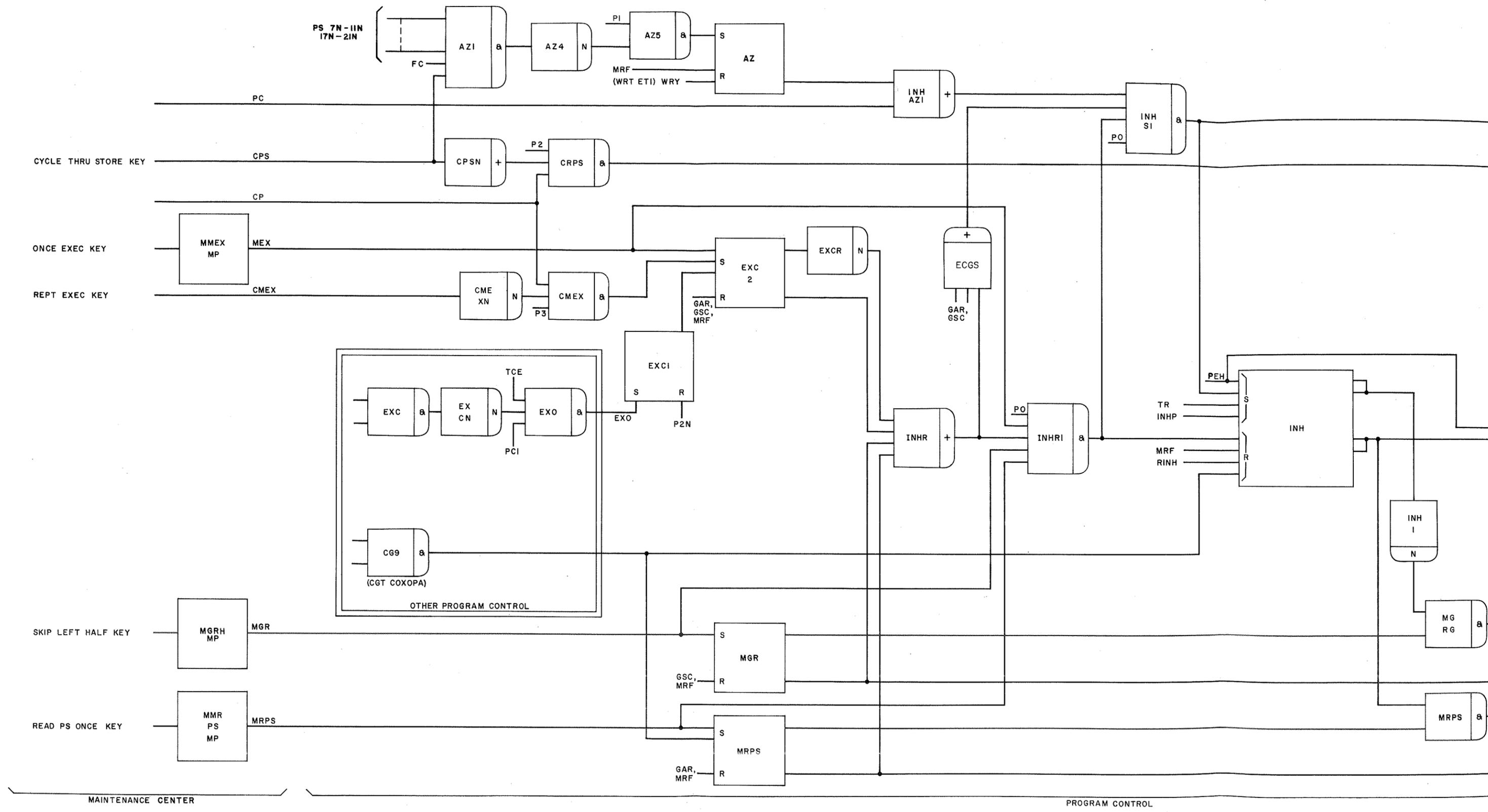


Fig. 12 - Maintenance Command Logic



MAINTENANCE CENTER

PROGRAM CONTROL

**13.03** Once translated, the command will be executed. In 4, 8, 12, or 16  $\mu$ sec, a GAR or GSC signal from the ready command logic circuit will indicate that the command has been completed. This resets the EXC2 flip-flop and enables the INHS1 gate through the ECGS gate. The INH flip-flop is then set so that the next command in the off-line call processor will not be translated.

#### B. Operating the Maintenance Center Keys

**13.04** The execute command can also be accomplished manually from the maintenance center. First, the remove system access to call processing (REM SYS ACC TO CP) key must be operated. Then the once execute (ONCE EXEC) key is operated. The output from the MMEK monopulser then sets the EXC2 flip-flop, and the command on the input side of the command translator will be performed just as though the execute command had come from the on-line call processor.

**13.05** If the repetitively execute (REPT EXC) key is operated, the CMEX gate is enabled, and the 4- $\mu$ sec pulses which occur every 16  $\mu$ sec on the CP lead, will set the EXC2 flip-flop every fourth P3. Thus, an entire routine can be executed from the maintenance center.

**13.06** If the read program store once (READ PROG STORE ONCE) key is operated, the RPS flip-flop will be set provided that the INH flip-flop has been reset. This causes the word addressed by the program store address register to be read out of the program store (see Fig. 6); however, this word will not be translated unless the ONCE EXEC key is thrown.

**13.07** The SKIP LEFT HALF key has been provided to manually set the RSC flip-flop in the off-line call processor. This will gate the K command which is on the right, over to the left, so that it can be translated through the use of the ONCE EXEC key.

**13.08** Operation of the CYCLE THROUGH STORE key enables the CRPS gate so that the RPS flip-flop will be set every 16  $\mu$ sec. The entire contents of the off-line program store can be read by holding this key down for about 1/2 second.

#### C. The All Zeros Detector

**13.09** The AZ1 gate will detect all zeros in the operation code bit positions for both K commands. Ordinarily this condition indicates trouble. However, the operation code for the command 1BT is all zeros. If this command appears on the right and is then gated to the left, there will be all zeros in both operation code bit positions. For this reason the FC output of the FSC flip-flop is wired to inhibit the AZ1 gate after any command has been gated to the right (see Fig. 6). The set all zero (AZ) flip-flop in an on-line call processor will in turn set the INH flip-flop, and the command translators will no longer operate. This causes an on-line time-out in the maintenance center which will cause the on- and off-line call processors to be exchanged.

**13.10** The more usual way to switch the on- and off-line call processors is to place positive battery on the PC lead in the on-line call processor and ground on the PC lead in the off-line call processor. This is done by the maintenance center. As may be seen in the figure, battery on the PC lead has the same effect as a set AZ flip-flop.

**13.11** Whenever the on- and off-line call processors are exchanged, the MRF lead is energized so that the INH flip-flop in the newly on-line call processor will be reset. However, the PC lead in the off-line call processors will become positive. The positive PC lead will, of course, set the off-line INH flip-flop.

#### D. Parity and Transfer Errors

**13.12** We have seen that the INH flip-flop in an off-line system can be set by a GAR or GSC signal after it has been reset by the EXC command. We have also seen that the INH flip-flop in an on-line system can be set by an all-zeros operation code in both K word positions of the program store output register and by a positive PC lead. In addition, the INH flip-flop can be set by battery on the PEH or TR leads indicating either a parity error or a transfer error. Although the bad word will be gated into the PSOR by the ready command logic, it will not be translated. A retrieval logic circuit will read the program store again; and if a parity or transfer error is found a second time, an on-line time-out in the maintenance center will exchange

the on- and off-line call processors. If no error is found on retrial, the RINH lead is energized to reset the INH flip-flop. In this case the on- and off-line call processors will not switch.

#### E. The WRT EB1 Command

**13.13** The INHP lead becomes positive if two WRT EB1 (set the error register bit equal to one) commands result from a maintenance routine with no intermediate WRT FB0 (reset the error register bit) command. The positive INHP lead will set the INH flip-flop. The RINH lead cannot reset the INH flip-flop during a WRT EB1 command, so there will be an on-line time-out as a result of the second WRT EB1.

### 14. PC AND THE PCI LEADS (Fig. 13)

#### A. General

**14.01** The PC leads in the two call processors are connected to the PC1 and PC2 flip-flops in the maintenance center. The two flip-flops and their associated relays have been represented symbolically in Fig. 13 by a PC- flip-flop. When the PC- flip-flop is set, call processor No. 1 is on-line, and when it is reset call processor No. 2 is on-line. The operation of the INH flip-flop in the maintenance command logic circuit by the PC lead is also indicated in this figure (see Fig. 12 for the details which have been shown as dotted lines). The INHT lead will be ground to enable the on-line command translators, and it will be positive battery to inhibit the off-line translators. The state of the PC lead, and as a result the INH flip-flop, determines which call processor is on-line.

#### B. PC Leads

**14.02** The PC lead is also wired to certain selected command translators. Fig. 13 shows it connected to the LSD translator. In addition, the PC lead connects to the STT, LDT, RDT, RED DS, and WRT DS command translators. All of these commands cause information to be gated between the input-output area and the call processor. Positive battery on the PC lead will prevent the performance of any of these commands in an off-line system as a result of the EXC command. This is desirable since the EXC command is the means for doing maintenance routines in the off-line call processor. If

the off-line call processor should be in trouble, it ought not to be allowed to affect the input-output area.

#### C. PCI Leads

**14.03** The command translators for the following group of commands are enabled or inhibited by the PCI leads:

EXC — Execute	WRT OC — Write into other call store.
ADV — Advance	RED OC — Read word in other call store into other call store register.
CGT CAXOCA — Gate call store address register to other call store address register.	AD1-OCA — Add 1 to other call store address register.
CGT OPAXOC — Gate other program store address register to other CO.	LSH — Long shift register.
CGT COXOPA — Gate call store output register to other program store address register.	

These are the commands which will allow conditions in one call processor to affect the other call processor. As may be seen on the examples shown in Fig. 13 (CGT CAXOCA, EXC) the PCI lead inhibits the performance of the off-line command translators to which it connects, just as the PC lead does. In addition, the PCI lead also inhibits its command translators when the REM SYS ACC TO CP key is operated in the maintenance center.

### 15. THE LINE INFORMATION STORE CONTROL CIRCUIT (Fig. 14)

#### A. General

**15.01** The line information store (LIS) is used as a repository for special service information for individual switch units or lines; for example, the numbers to be compressed dialed are placed in the LIS. As may be seen in Fig. 14, access to the store is *from the CO* and the word read out of the store is gated *into the CO*. The command CGT COXLA enables a gating path into the address register. The word and plane

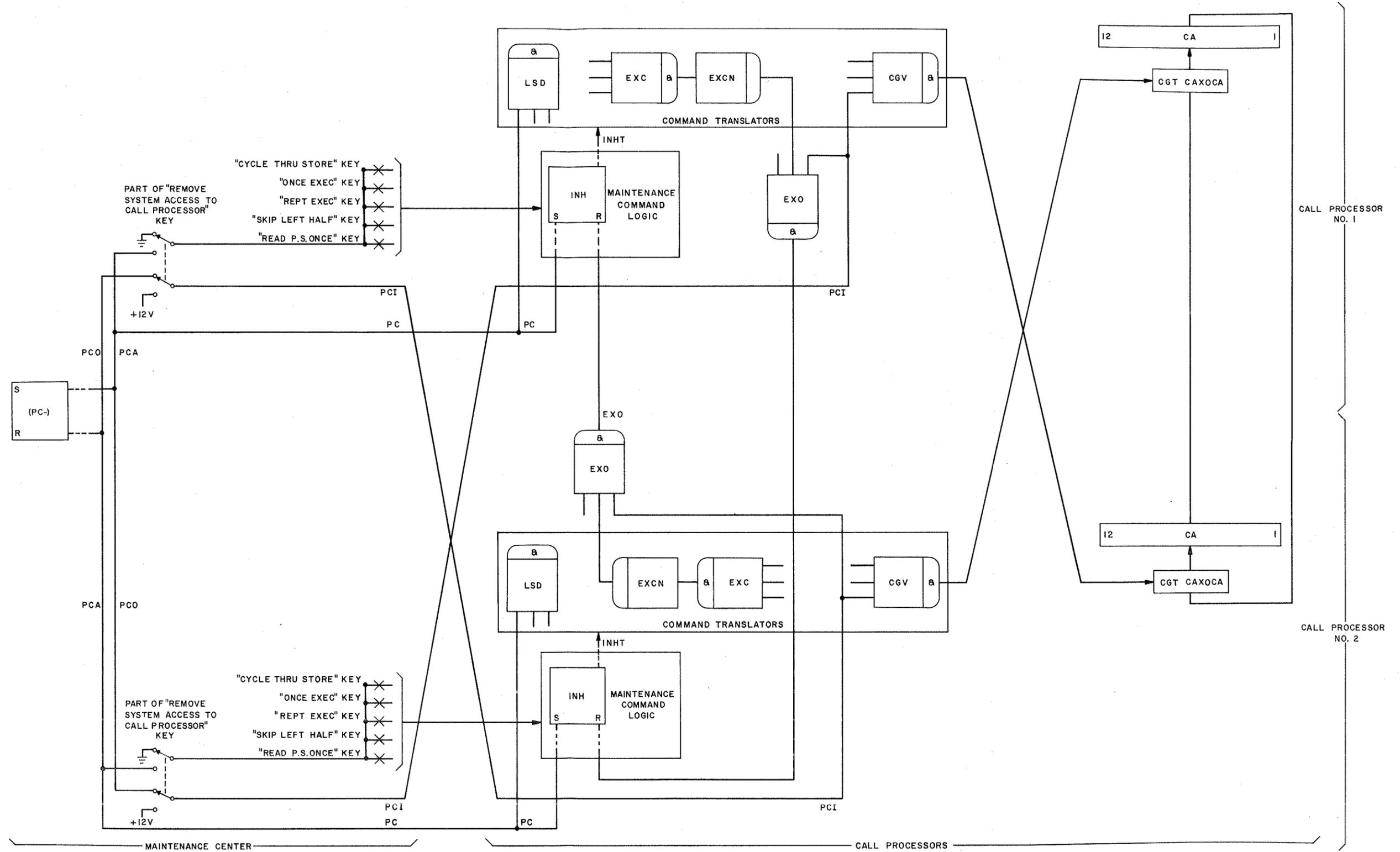


Fig. 13 - PC and PCI Lead Controls

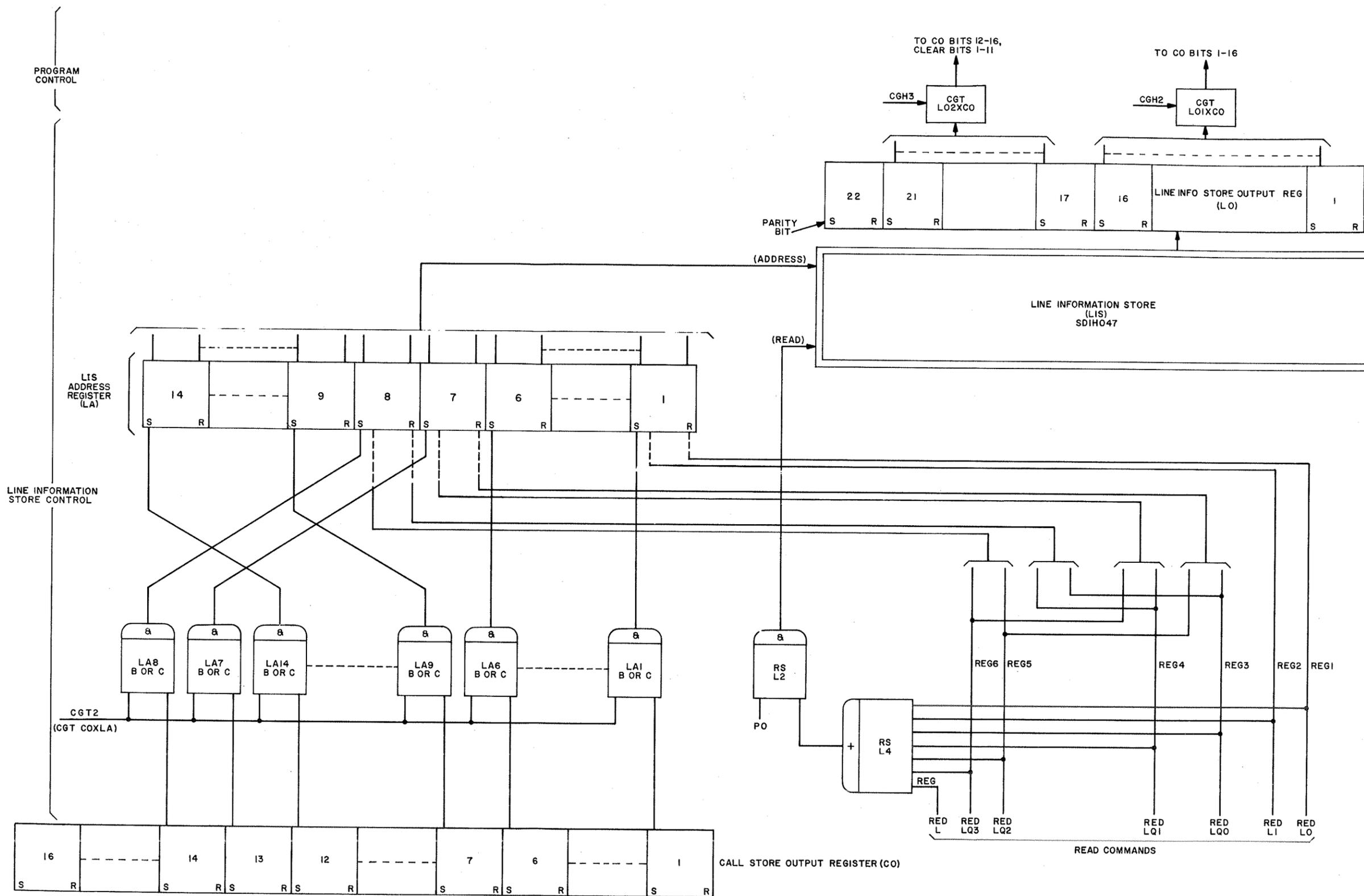


Fig. 14 - Line Information Store Control

bits are gated from CO bit positions 1 through 12 into the LIS address register. The quadrant bits are gated from CO bit positions 13 and 14. After the store has been read, the two commands, gate LO bits 1-16 to CO (CGT LO1XCO) and gate LO bits 17-21 of LO to bits 12-16 of the CO (CGT LO2XCO), are used to gate the 21-bit word into the CO.

#### B. The Red L Commands

**15.02** There are seven commands which are used to read the LIS.

- (1) RED L0 means to read the LIS after modifying the address so that bit one equals 0.
- (2) RED L1 means to read the LIS after modifying the address so that bit one is equal to 1.
- (3) RED LQ0 means to read the LIS after modifying the quadrant bits to equal 00.
- (4) RED LQ1 means to read the LIS after modifying the quadrant bits to equal 01.
- (5) RED LQ2 means to read the LIS after modifying the quadrant bits to equal 10.
- (6) RED LQ3 means to read the LIS after modifying the quadrant bits to equal 11.
- (7) RED L means to read the LIS without modifying the address.

#### 16. THE IDENTIFIED OUTWARD DIALING INTERFACE CIRCUIT (Fig. 15)

**16.01** When the central office with which a No. 101 ESS is associated is equipped with automatic message accounting (AMA) equipment, charges for toll calls originating at a given switch unit will be made against the trunk number (equivalent to the customer "switchboard" number) or to the particular extension involved, depending on whether certain options are provided. Service permitting a particular extension to be identified, when making outgoing toll calls, is termed "automatic identi-

fied outward dialing" (AIOD or IOD). This service requires the provision of an IOD interface circuit in the No. 101 ESS control unit and a cooperating AIOD translator in the AMA equipment of the central office. Identifying numbers are also required in the line information store for the extensions given this service.

**16.02** When a call requiring extension number identification is dialed, the AIOD translator in the central office will furnish the call processor with a number which identifies the central office trunk used by the subscriber. The 4-digit trunk number will appear on the leads TK0 through TK19 in the 2-out-of-5 code. The TKRI lead will be energized at this time to enable a gating path to the on-line program control circuit.

**16.03** Program is then used to find which extension is connected to the central office trunk. The 4-digit directory number of the extension is gated in 2-out-of-5 form to the IOD buffer. A 3-bit office index code is also gated to the IOD buffer at this time. All of this information is furnished to the AIOD circuit over the leads EX0 through EX22. The EXN **OR** gate is used to signify to the program whether an extension number is present in the IOD buffer.

#### 17. REFERENCES

**17.01** The following SD drawings and associated circuit descriptions (CDs) give additional information. These are not attached.

Call Store	SD-1H046
Program and Line Information Stores	SD-1H047
Line Information Store Control	SD-1H048
Program Control	SD-1H049
Program Shift Register	SD-1H050
System Clock	SD-1H051
Maintenance Center Circuit	SD-1H052
Identified Outward Dialing Interface	SD-1H056
Printout Circuit	SD-1H060

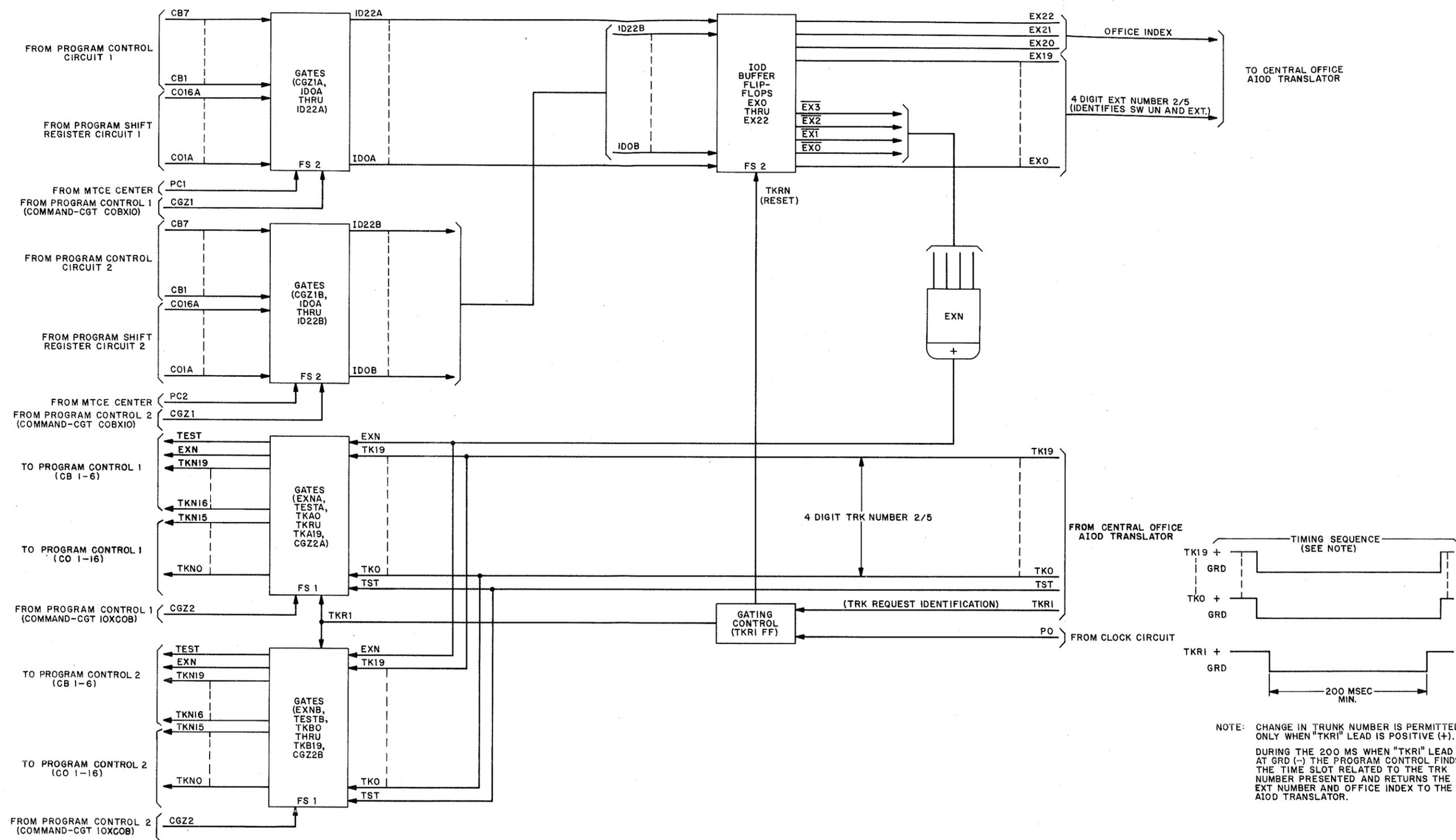


Fig. 15 - Identified Outward Dialing Interface