3A CENTRAL CONTROL (3A CC)
THEORY OF OPERATION
COMMON SYSTEMS

CONTENTS PAGE

Figures

Printed in U.S.A.

CONTENTS	PAGE

CONTENTS PAGE

CONTENTS PAGE

CONTENTS	PAGE

1. GENERAL

INTRODUCTION

1.01 This section describes the theory and operational characteristics of the 3A central control (3A CC) which is a part of the 3A Processor used in a variety of control and processing systems.

1.02 Whenever this section is reissued, the reason(s) for reissue will be listed in this paragraph.

1.03 This section includes:

(a) Functional discussions of signal, control, and power circuits

(b) Functional relationships between 3A CC control panel and system status panel (SSP).

1.04 The 3A CC (Fig. 1 and 2) obtains and translates instructions (commands) from main store (MAS) into system actions. These actions include not only those operations completed entirely within the 3A CC, but also interactions between the 3A CC and other units which it accesses. The 3A CC receives inputs from data terminals, data processors, monitors, and teletypewriters and performs predetermined operations necessary for message formatting, data collection, data transfers, system maintenance, diagnostics, and system recovery.

PURPOSE OF 3A CC

1.05 The 3A CC is used to obtain a series of instructions as commands from MAS and translate these into actions. These actions may be within the 3A CC itself and/or its associated units. The 3A CC receives input information from lines, trunks, scanners, controllers, etc. It performs logical operations on the data and transmits output data to control switches, data terminals, controllers, etc. All of these actions are directed toward providing data transfers, diagnosing system troubles, and responding to status conditions affecting continued system operation.

CONFIGURATION

1.06 Data and instructions are received from main memory via a store bus network which allows each 3A CC access to the stores. Figure 3 shows a typical duplex 3A CC configuration.

1.07 Reliability of operation is achieved through duplication of all major system components. The store bus is a bidirectional link with read data utilizing the same paths as store data. Also, most of the peripheral units are actually duplexed, with one of each unit permanently connected to one of the 3A CCs. In this configuration, one 3A CC has active control of the system (on-line) while the other is in a standby (off-line) mode. Although off-line operation is asynchronous to the on-line process, it is not completely independent of the active system. One example is the on-line 3A CC keeping both on-line and off-line memory updated so that the off-line 3A CC can assume control, if necessary.

2. 3A CC INTERFACES

2.01 Figure 4 is a high-level block diagram depicting typical 3A CC interfaces. Diagram orientation is around unit zero of the duplicated 3A CCs. Main store bus labels correspond to the 3A CC providing direct control. The My main store bus is the data path between a 3A CC 0 and its associated main store. The Other main store bus is used for communication with the Other 3A CC memory. This bus is used by the on-line 3A CC for off-line memory update and double-store read functions. When viewing bus and interfacing circuits structure from the other 3A CC (1) position, all bus and channel designations remain as shown.

2.02 Occasionally in 3A documentation, a 3A CC and its associated main store(s) will be identified as the control unit (CU). A still larger label reference encountered includes all controlling circuitry as system control (SYC). Figure 5 is a block diagram of the Transaction Network which illustrates a typical interface of the 3A CC with an application.

3. THEORY OF OPERATION—SIGNAL AND CONTROL CIRCUITS

INTRODUCTION

3.01 The 3A CC is a highly compact device, incorporating techniques not previously found in BTL-designed central control units. A large number of general and special purpose registers together with their control circuitry is included in the design. This number of registers allows data manipulation with fewer memory accesses, thereby enhancing 3A CC processing efficiency. A microprogram control technique is used in the 3A CC instead of instruction function translation and synchronized execution control.

3.02 Generally, the 3A CC obtains a stored program instruction from main store (MAS) and translates it into action. The translation consists of executing a set of microinstruction steps located in microstore to obtain the action required. Essentially, each instruction stored in MAS is a pointer to a starting address in microstore of a sequence of microinstructions to perform the action. Each microinstruction contains a portion of the action and address of the next microinstruction of the sequence. This microinstruction process continues until the instruction action is completed. When the sequence is completed, microcontrol interrogates MAS to determine whether the next stored program instruction is available to execute. Microcontrol loops until the MAS instruction fetch cycle is complete. When the next instruction is available, a new microinstruction sequence is initiated. By breaking instructions down in this manner, considerable flexibility is incorporated into the 3A CC. This technique also allows regularity in data handling sections of the machine. Special functions such as interrupts and initialization are coded into microinstructions which employ normal process paths, but are initiated by status conditions rather than instructions. The execution of an instruction may be changed without redesign of the machine by changing the microinstruction store circuit packs to packs in which the new microinstructions are encoded.

3.03 The 3A CC is a self-checking machine. It contains a large number of check circuits which indicate correct operations of each instruction. An example of this capability is the bit slicing technique used in most of the registers. In such an arrangement, consecutive bits of a register are located on different circuit packs. This reduces the possibility of compensating bit failures and undetected bit failures since it would be necessary for two circuit packs to fail simultaneously.

3.04 Supporting the 3A CC maintainability is a maintenance channel circuit which provides a dedicated path between the duplicated 3A CCs. This channel is separate from other input-output (I/O) circuits and includes special direct access to 3A CC control logic. By utilizing this link, one 3A CC has the ability to exercise the other.

3.05 The 3A CC obtains and executes a sequence of instructions from MAS. Two types of instructions are available, single-word and multiple-word instructions (Fig. 6). Single-word instructions are the most commonly used; however, multiple-word instructions are used when additional word length is required.

3.06 The 3A CC instructions enable reading from or writing into any of the general registers. Since most of the instructions allow any general register to be used, it is not necessary to move data to a special register to perform a function.

3.07 The general formats (Fig. 7) for 3A CC instructions are:

- RR—Register-to-register
- RN—Register and immediate operand
- RXR—References memory by adding an index register to an address register pair
- RXN—References memory by adding a number N to an address register pair
- RI—Register and immediate data
- SL—Specifies 20-bit data to load a register pair or to reference memory
- SS—Specifies 8-bit offset in branch operation.

All are single-word instructions except RI and SL types. Each instruction contains two parity bits, one branch allowed (BA) bit, and a 7-bit OP code.

3.08 All words in the system have odd parity over eight bits and two parity bits over each 16-bit word. The two parity bits of each word are parity (P_L) on the low eight bits (0 through 7) and parity (P_H) on the high eight bits (8 through 15).

3.09 The BA bit is used in program transfer or branch processes. When a branch instruction occurs, a hardware check ensures that the BA bit of the next instruction is set. If for some reason the BA bit is not set, an error is indicated in the error register of the 3A CC.

3.10 The 7-bit OP code field in all instructions specifies the function to be performed; it is used to access a set of microinstructions which accomplishes the function indicated by the OP code.

3.11 The remaining bits of both single-word and multiple-word instructions contain different arrangements and types of information depending on the particular instruction.

3.12 The instructions used in the 3A CC are broken down into classes according to functions performed. These classes are:

(a) Data transfer instructions

(b) Branch instructions

(c) Arithmetic instructions

(d) Logic instructions

(e) Bit operation instructions

(f) I/O instructions

(g) Maintenance instructions.

The instructions are described within the class to which they apply. Within each class, both types of instructions, single word or multiple word, are used.

3.13 The maintenance channel (MCH) is used to communicate between 3A CCs and to permit one 3A CC to exercise the other. The structure and functions of the MCH are discussed in 3.126. The word format of an MCH message, as shown in Fig. 8, is 22 bits of data and parity and 8 bits of command including the start bit. The data is formatted the same as the 20-bit gating bus. For 16-bit data fields, the high 4 bits of the high data field are not used. For 12-bit data fields, only the full high data field is used with the P_H.

3.14 The system status panel (SSP) is used to control and display status of the 3A processor and its parent system. It has both an I/O channel and direct cabled connections with the 3A CC to accomplish this control. The SSP circuits require a 21-bit message format over the I/O channel. Figure 9 shows the format and briefly defines the bits of an SSP I/O message.

FUNCTIONS OF 3A CC CIRCUIT

3.15 The 3A CC, together with a 3A CC control panel, forms the basic control equipment of a 3A processor (Fig. 10). The 3A CC circuits and their major functions are:

(a) System clock provides basic timing for the 3A CC. It is the source of timing signals whenever an operation requires clocking.

(b) General register portion of central control registers provides a high-speed scratch pad memory that may be used to store data, results of logic operations, etc. Each of the 16 registers consists of 16 data bits plus 2 parity bits. The registers are numbered R0 through R15. In addition to normal register functions, registers R9, R10, and R11 are also used to interface with the I/O channels. Register pairs are used to store addresses or data when the word to be stored exceeds the 16-bit length of a single register.

(c) Special registers are used for dedicated purposes as follows:

(1) CR (C register), a general purpose register, is generally used by the microprogram control to buffer results of data manipulation logic (DML) operations.

(2) ER (error register) buffers the output of check circuits. Each bit represents a specific fault.

(3) HG (hold-get) register is used as a pointer to the hold-get table area in main memory. The hold-get area is used to store the state of selected 3A CC registers and a return

address at the time of a subroutine or interrupt execution.

(4) IM (interrupt mask) register contains the masking bits for IS register. When a mask bit is set, the interrupt corresponding to that bit in IS register is blocked.

(5) IS (interrupt set) register buffers incoming interrupts until each is acted on by 3A CC.

(6) MS (maintenance state) register buffers maintenance states of 3A CC. This allows the diagnostic program to set up conditions which enable the diagnostic program to test the 3A CC.

(7) SS (system status) register buffers various status states of the 3A CC; each bit represents a different condition.

(d) Microprogram control initiates a sequence of microinstructions which translates each instruction fetched from main memory. Microprogram control contains a read-only microstore, control registers, and decoders that decode microinstructions. Each main memory instruction points to a starting address in microstore of a series of microinstructions which actually carry out the desired operation. Each microinstruction contains a TO and a FROM field which contain the information necessary to determine where data is to go and from where it is to be obtained. The TO, FROM, and miscellaneous decoders attached to the microstore instruction register (MIR) TO and FROM fields decode the information into control signals which control the data flow. The microinstruction also contains a next address (NA) field, which normally provides the microstore address of the next microinstruction in the series. This address field may contain data that is to be gated to other registers. Then, the next microinstruction will be located at the present address +1. In addition to microinstruction processing, microcontrol contains a microcontrol status register (MCS) which saves such states as adder status, I/O status, data ready, etc. Microcontrol examines this status information in modifying its operation when addressing the next microinstruction. Microprogram store is a read-only memory of 3072(3K) words of 32 bits each. It can be expanded by 512 word increments up to a maximum of 4096(4K) words.

(e) DML (data manipulation logic) consists of a series of registers with attached arithmetic logic to perform addition, subtraction, and Boolean operations. It also contains logic to perform shifts and rotations of the contents of a register. The DML computes parity on its outputs so that data may be gated to other portions of the 3A CC without causing an error.

(f) Interrupt facility consists of, in addition to the IS and IM registers, logic which responds to status conditions that initiate interrupts. When interrupts are recognized, a hard-wired address is gated to the microcontrol memory address register (MAR), thereby linking to the interrupt processing portion of the microcode.

(g) Main memory interface consists of registers and logic to control communications between the MAS and 3A CC. The store address register (SAR) holds the address of the word in MAS that is to be accessed. Program instructions are buffered in the store instruction register (SIR) when read from MAS. Data is buffered in the store data register (SDR) for use in either a read or a write operation. The main memory status (MMS) register controls the mode of operation for each store cycle.

(h) The I/O channels provide the communication links between the 3A CC and its peripheral devices. A serial channel is provided through which the 3A CC can exchange data with slow speed devices such as teletypewriters. Parallel buses provide the communication links with direct memory access (DMA) and parallel channel (PCH) units which interface with high speed devices such as magnetic tape units. A 3A CC is capable of addressing up to 20 main channels. Each main channel can address up to 20 serial subchannels or 8 parallel subchannels.

(i) Maintenance channel is a serial data link over which the duplexed 3A CCs communicate with each other independently of other peripheral activities. The primary purposes are to:

(1) Provide a means of switching off-line 3A CC on-line and removing on-line 3A CC from service if a fatal error occurs in the on-line machine.

(2) Allow on-line machine to send diagnostic orders to off-line machine to locate a fault.

(3) Provide a path for on-line machine to monitor state of off-line machine.

(j) Console functions apply primarily to those associated with the interface between the 3A CC and its control panel. Operation of the panel EXECUTE key causes an interrupt to be generated to notify microprogram control that a panel function is required. Under microprogram control, contents of the panel switches are selectively gated into display buffer (DB). Once in DB, switch entry is interpreted and requested action initiated. Address and data matches can be performed on any desired MAS cycle by match logic included in the 3A CC. The SSP also interfaces with the console function group to provide direct actions such as lock off- or on-line, initialize, etc.

(k) Data transfer check circuits are a set of function checkers which will activate when the function is in error. The TO field and FROM field decoders, branch allowed (BA) checker, gating bus parity check, I/O channel checker, etc, are among items whose operation is checked.

A. System Clock

3.16 The system clock (Fig. 11) is composed of a crystal oscillator, a phase splitting circuit, and a set of counter circuits. The basic functions are to provide timing signals for operation that require synchronization within 3A CC and to generate timed interval interrupts. A subfunction of the latter includes initiation of CC switchover if call processing execution breaks down. The crystal oscillator generates a 26.67-MHz sine wave signal which is then passed through a squaring circuit to obtain a square wave with a period of 37.5 nanoseconds. This signal is applied to a 4-phase clock circuit which generates four clock phase signals. These signals (P0, P1, P2, and P3) constitute one complete basic cycle with a period of 150 nanoseconds. The four clock phases are used directly to synchronize other 3A CC functions.

3.17 Clock phase 1 drives a binary counter to generate timing intervals of 1.2 microseconds, 19.2 microseconds, 38.4 microseconds, 76.8 microseconds, and 153.6 microseconds. These intervals are used to control miscellaneous functions within the 3A CC. The timing (TI) register is driven by the 19.2-microsecond intervals from the counter to provide counts for timed interrupts at intervals of 1.25, 5.0, 10.0 and 25.0 milliseconds (Fig. 12). An RC-controlled oscillator provides a 625-Hz signal used as a time-out clock in maintenance and test modes.

B. Central Control Registers and Interrupt Facility

3.18 There are 16 general registers designated R0 through R15 in the 3A CC. These registers are 18 bits wide, with 16 data bits and 2 parity bits. The general registers are used primarily as "scratch pads" or temporary storage for main memory programs. All general registers have access to the gating bus to effect transfer of data to and from the registers. This access is controlled by instructions read out of microstore which, when decoded, open and close gates to and from the selected source and destination registers. A technique referred to as bit-slicing is used to increase the probability of detecting faults in the registers. Bit-slicing is the process of placing consecutive bit cells of a register on different circuit packs so that failure of a single circuit pack will cause failure of only one bit in each byte. This will cause a parity error to be immediately detected. Figure 13 provides a block diagram of the general registers and of bit-slicing. Table A is a definition of terms used in the description of instructions.

3.19 Registers R9, R10, and R11 are used for communicating with I/O channels (Fig. 14). Register 9 is used to contain the I/O channel, subchannel, and device address. Register 10 normally contains data to be sent to a peripheral device, and R11 receives data from a peripheral device. Register pairs can be established which can be used when storage is required of data or addresses with lengths of more than 16 bits. Table B is a listing of 3A CC instructions.

3.20 The 3A CC also contains a set of special registers (Fig. 15) which is used for dedicated functions. The special registers access the gating bus in the same manner as general registers. Names and functions of these registers are:

ER — Error register. Stores all error signals from check circuits in the 3A CC and main store.

HG — Hold-get register. Saves pointers into an area in main store where contents of the general registers are saved when subroutines using HG are executed.

MS — Maintenance states register. Stores maintenance states that are used to set up test conditions within the 3A CC for maintenance purposes.

IS — Interrupt status register. Stores interrupts as they occur.

IM — Interrupt mask register. Provides mask for the IS register so that higher priority interrupts can be serviced first.

CR — C register. Serves as a general purpose register for the microprogrammer. Generally used to buffer results of data manipulation logic operations.

SS — System status register. A 20-bit register that contains system state information.

Table C defines the addressable special registers.

3.21 Interrupt mask (IM) register, interrupt set (IS) register, and interrupt allow logic comprise the interrupt facility. This facility allows a maximum of 16 interrupts to be buffered in the IS register. Each of these interrupt bits has a corresponding masking bit in the IM register which, if set, will block recognition of the interrupt. Interrupt levels have been assigned as shown in Table D. Priority level follows bit order with bit 0 having the highest priority and bit 15 the lowest. When an interrupt occurs, the corresponding bit is set in the IS register. All unmasked bits of IS are ORed to generate an interrupt present (INTPRS) signal. When another interrupt is not already being serviced, the interrupt allow logic generates an interrupt signal (INTRP) using INTPRS and phase 2 of the system clock. Interrupts are serviced at the end of microinstruction sequences. End of a sequence is signalled by the next address field of the last microinstruction being set to all zeros. When the all-zero condition is detected, the all-zero signal and the INTRP signal load the beginning address of the interrupt servicing microcode sequence into the microstore address register.

3.22 The interrupt facility in the 3A auxiliary processor is used not only as part of the maintenance and error reporting function, but also as a part of the normal data transfer routines. When a peripheral device which uses interrupts needs to communicate with the 3A CC, it sets the interrupt (INTP) lead, which sets the associated bit in IS. PCH 0, PCH 1, and DMA/PCH are each associated with a particular bit in IS, but the user has the option of specifying the priority ranking of the peripheral devices by means of strapping between INTP 0 through INTP 7 leads and INTP A through INTP H leads to IS. Refer to Fig. 16 for a simplified block diagram of interrupt hardware, and to Fig. 17 for the 3A CC/channel unit interface. Table D is a listing of priority level assignments.

3.23 When the 3A CC accepts an interrupt for servicing, ACKI is generated and returned to the interrupting device to indicate that the request is being processed and that the device should identify itself. The bit set in IS identifies only the channel unit. Any one of a possible 16 devices attached to the channel unit could have generated the interrupt; therefore, further identification is necessary. Each peripheral device is assigned a specific INF lead to be used for identification purposes. When ACKI is received, the interrupting device sets its INF lead to a logic 1, while all other devices set their INF leads to 0. The 3A CC then effects the data transfer by loading the channel and device addresses in R9, loading output data into R10, or gating input data into R11.

3.24 Maintenance state register (MSR) is used to condition the 3A CC for testing purposes. During normal processing, MSR is cleared to eliminate processing errors which would otherwise occur because of attempted execution of maintenance tests. A miscellaneous decoder cross point (zero maintenance state register—ZMS) is available to clear the register. Most of the MSR bits have independent implementation signal paths except for the three clock test leads (MSR bits 1, 2 and 3). (Refer to 3.18). Table E lists MSR bits and their function.

3.25 The system status register (SS) contains control bits which establish operating conditions of the 3A CC. A separate set of leads interfaces the SS with the maintenance channel. A signal (gate SS to maintenance channel - MCSSMCH) gates the contents of the SS directly to the maintenance channel transmit receive register. Table F identifies the SS bits and their functions.

3.26 The error register (Table G) buffers errors detected by check circuits in the 3A CC as well as some errors from MAS. Each bit of the error register can be set by its check circuit. In addition, it has normal register access to the gating bus as well as its special clear signal, and a signal from maintenance channel that permits the contents of error register to be gated into maintenance channel. This allows one 3A CC to read the error register of the other 3A CC without interfering with the one being read.

3.27 The bits in the error register are divided into groups according to the severity of the error represented and the error recovery procedure to be followed. Refer to Table G for a list of these errors. Bits 0 through 9 are classed as fatal errors. Setting any one of these bits stops the on-line 3A CC and causes a switch to the off-line 3A CC unless the block hardware check (BHC) or lock on line (LON) bits are set in the status register. Bit 10 is the read parity error from the MAS. This bit will initiate an attempt to correct the error by a double store read or a complement correction. If the attempt is successful, processing will continue without interruption.

3.28 Bits 12 and 13 are the maintenance reset function class of errors. Either of these errors will cause initialization of the 3A CC unless BHC is set in the status register.

3.29 Bits 11 and 14 through 19 and P_L and P_H form the interrupt class of errors. Any of these errors will generate a level 5 interrupt (set bit 5 of the interrupt set register.)

C. Microprogram Control Microstore

3.30 The microprogram control is basically a translator and sequencer of instructions read out of MAS. An instruction from MAS does not contain all the information required for execution. Instead, each instruction contains a pointer to an address in microprogram store which holds the first step in a sequence of microinstructions. This sequence issues detailed orders to microprogram control to execute the MAS instruction. When the sequence ends, the microprogram control returns to a state that can accept the next MAS instruction. The microprogram store (MIS) contains many such sequences used not only to translate and execute commands but also to control other functions such as interrupts, control panel inputs, program loading, error checking, etc.

3.31 Microprogram control is made up of the following functional units:

Microprogram store (MIS)

Microinstruction register (MIR)

Microaddress register (MAR)

Return address register (RAR)

Instruction buffer (IB)

Decoders and translators

Microcontrol status register (MCS)

Error return address register (ERAR).

The MIS is a read-only memory with an access time of approximately 65 nanoseconds. It has a maximum size of 4096 (4K) 32-bit words. In the TN, the MIS has 3072 (3K) words encoded in it.

3.32 Once programs are written into the MIS, they cannot be changed. If a program change is required, a new circuit pack containing the changes must be installed. However, the 3A CC can be used in many different applications without modification by installing a MIS containing the proper microprograms. MIS is accessed from MAR, and its outputs are gated to MIR.

3.33 The MIR is a 32-bit register whose inputs are normally from the MIS. It is divided into fields which are individually decoded to provide signals to control operation of the 3A CC (Fig. 18). These fields are:

Bits 0-7—FROM field—Defines source register for gating operations

Bits 8-15—TO field—Defines destination register for gating operations

Bits 16-27—NEXT ADDRESS field—Contains address of next instruction in MIS, data to be used, or control information

Bits 28-31—Control and parity bits—Define the contents of the next address field.

3.34 Outputs of the TO and FROM fields are applied to decoders whose outputs designate source and destination registers in a gating operation. Each decoder outputs one of 70 possible signals representing the positions of four 1s in the bit field. (These are called four-out-of-eight codes.) The FROM field designates the source register while the TO field designates the destination register.

3.35 The miscellaneous decoder is a matrix formed by 14 outputs from TO decoder and 8 outputs from FROM decoder. The signals produced by the ANDing of TO and FROM signals (crosspoints) are used to set and clear various flip-flops and registers and to enable dedicated gating paths.

3.36 Return address register (RAR) is a 12-bit register used to store the return address of a microsubroutine. When the program enters a subroutine, the address to which the program returns (upon completion of subroutine) is stored in RAR. Table H identifies the MCS bits and their significances.

3.37 Microcontrol status register (MCS) is used to control a number of special functions such as conditional transfers, RAR functions, data manipulation logic status, etc. MCS contains key status bits that are used to control microcode execution flow. The MCS is duplicated so that MCS 0 actually controls microprogram execution, while MCS 1 inputs to check circuits used to verify accuracy of the microprogram execution. Individual bits in MCS are tested by gating that bit into MAR 0 and performing a conditional branch.

3.38 Error return address register (ERAR) is used to store the return address to be used after completion of a store error correction routine.

3.39 Microinstruction sequences are generally initiated by fetching an instruction from MAS and loading the operation code portion into MAR. This becomes the beginning address of a microsequence that will execute the instruction. The microsequence loads new addresses into MAR by one of the following methods:

- Load MAR from NA field of MIR. This is the most common method of putting a new address in MAR.
- Increment MAR address by 1. This is normally done when NA field of MIR contains data instead of an address.
- Load MAR from RAR. This is done when returning from a subroutine.
- Load MAR with a predetermined address when an interrupt or initialization occurs. [O(120) for an interrupt and O(277) for initialization].
- Load MAR from ERAR. This is done when returning from a store error correction subroutine.

3.40 Figure 19 is a flowchart that depicts the various means of addressing MIS, and Fig. 20 is a simplified logic diagram of addressing.

3.41 The instruction buffer (IB) is used to store the OP code and operand field read from MAS. When an instruction is read from MAS, it is gated to store instruction register 0. The all-zeros detector on the NA field of the microinstruction register detects the all-zero condition which signals that the previous microsequence has ended. If data ready (DR) flip-flop is set, the instruction in SIR is gated into the instruction buffer and MAR. If DR is not set, the microcontrol loops on the all-zero location until DR is set.

3.42 When the OP code is obtained from MAS, it is loaded into the MAR and IB. Two 4-bit groups can also be loaded into the IB X and Y fields (bits 7-4 and 3-0, respectively). Outputs of the X and Y fields feed 3-out-of-6 translators, whose outputs can in turn be conditionally gated to the low 6 bits of the TO and FROM fields to the MIR. (Refer to Fig. 21.) This gating is accomplished when all zeros are detected coming out of MIS in bit positions 13-8 and 5-0. The upper two bits (15, 14 and 7,6) out of MIS are decoded into a 1-out-of-2 code which determines whether the general register set (0,1) or the maintenance register set (1,0) are to be used, or if the operands going to the TO and FROM fields of MIR are to be interchanged (1,1). As shown in Fig. 22, register substitution from IB X and Y or operand interchange can only take place when bits 13-8 and 5-0 out of microprogram store are all zeros.

3.43 Indexing is performed by gating the X or Y fields of IB into MAR 0-3 at clock phase P1. Miscellaneous decoder crosspoint (MDCOLB) (MDROW10) gates IB Y field into MAR, while crosspoint (MDCOLC) (MDROW10) gates the X field into MAR. MAR then contains an address which is the OR of MIR (NA) field and IB X or Y.

3.44 All tables in microstore must begin on 2-, 4-, 8-, or 16-word boundaries because of the technique used to increment an address. For example, a 16-word table must begin at an address whose last four bits are zeros.

3.45 Conditional branching is possible in the microcontrol. Key status bits that can be tested by microcontrol are in the MCS register. The MCS bit that is to determine the branch is gated into MAR 0 by MARP 1 and clock phase P1. Conditional branches, therefore, must occur on even-numbered addresses so that if the bit being tested is a 1, the branch address will be N+1, while if the bit is a 0, the branch address will be N.

3.46 TO and FROM field decoders function in an identical manner. These decoders are called 4-out-of-8 decoders because there are 8 input bits, 4 of which are 1s. There are 70 possible outputs from each decoder, each output corresponding to a particular pattern of 1s at the input. Outputs are re-encoded to the complement of the input, and then compared with the inputs. The output of the comparator should be zero. If not, an error signal is produced.

3.47 The 4-out-of-8 decoders are each divided into two 4-bit decoders. The 4-bit decoders are identified as being left subfield or right subfield (Fig. 23). Bits in either the left subfield or the right subfield must have one of the following combinations.

0 ones out of the 4 bits	(0O4)
1 one out of the 4 bits	(1O4)
2 ones out of the 4 bits	(2O4)
3 ones out of the 4 bits	(3O4)
4 ones out of the 4 bits	(4O4)

This provides 16 possible combinations.

0O4	1
1O4	4
2O4	6
3O4	4
4O4	1
4O4	16

3.48 Outputs from the two subfields are then paired to obtain one of 70 possible combinations. Since there is a total of four 1s in both subfields, if one of the subfields contains three 1s, the other can only have one.

3.49 TO and FROM decoders also drive the miscellaneous decoder which is used to control various flip-flops and dedicated gating paths. Fourteen of the decoded TO outputs and eight of the FROM outputs are ANDed together to form a total of 112 possible control signals or crosspoints. About 94 of these crosspoints are used. Figure 24 is a symbolic representation of the matrix, and Fig. 25 is a simplified logic diagram.

3.50 The mnemonic definitions of TO and FROM field control signals on the back plane are named to reflect the 4-out-of-8 code that produced them. For example, the 4-out-of-8 code (1100 0101) in FROM (F) field develops two leads, one from the left (L) half and one from the right (R). The two leads are F2O4LCO and F2O4R50. The leads are broken down into the following fields:

1	2	3	4	5	6
F	2O4	L	C	O	B

Field 1 F = FROM field

T = TO field

Field 2 1O4 = one 1 in the 4-bit subfield

2O4 = two 1s in the 4-bit subfield

3O4 = three 1s in the 4-bit subfield

4O4 = four 1s in the 4-bit subfield

Field 3 L = Left half 4-bit subfield

Field 3 R = Right half 4-bit subfield

Field 4 C = Hexadecimal representation of the contents of the 4-bit subfield that causes the control signal to be active.

Field 5 0 = Control signal is low when active

1 = Control signal is high when active

Field 6 Blank = only one fanout

6 A = first fanout of a multiple fanout signal

6 B = second fanout of a multiple fanout signal

6 C = third fanout of a multiple fanout signal

Figure 26 illustrates how the fields are divided and how the lead designations are formed.

3.51 The auxiliary decoder (Fig. 27) is a 2-out-of-4 decoder which is enabled when the CA and CB bits are both 1s. It is driven by bits 8, 9, 10, and 11 of the microinstruction register NA field. The following system controls are provided by the auxiliary decoder:

NA FIELD VALUE	CONTROL FUNCTION
0110 X(6)	NA (7-0) gated to function register (7-0).
1001 X(9)	I/O parity divert. Checks parity on incoming message and diverts parity while message is being received.
1100 X(C)	I/O DML match divert. Diverts a match on DML logic.
1010 X(A)	Disable bus parity check. Allows bus parity check to be turned off for one microinstruction. Used primarily in maintenance programs.
0101 X(5)	Spare
0011 X(3)	Spare

3.52 MAS includes an area reserved for storage of the contents of the 3A CC general registers when hold-get subroutine instructions are executed. This area is called the hold-get area and consists of a number of 16-word blocks (Table I). The base address [O (10760)] of the first block is stored in the hold-get (HG) counter. Each call to a subroutine decrements the HG counter by 16 to provide the address of the block to be used, and each subroutine return increments HG by 16.

3.53 Words 0 and 1 in each HG block are reserved for the return address of the subroutine, while words 2 through 15 are reserved for general registers 2 through 15, respectively. The contents of any or all the general registers except R0 and R1 may be saved. When the return from the subroutine is made, the registers are reloaded from the HG block.

3.54 Virtually any number of subroutines may be nested, the quantity being limited only by the amount of MAS available to store the 16-word blocks. Fig. 28 is a diagram of the program flow. Main program subroutines are software driven and do not require any special hardware. Subroutine instructions are executed just as any other instructions.

3.55 The microcode has subroutines also. When a microstore subroutine is to be entered, the instruction contained in MIR contains CACB bits set to 2, and the NA field contains the return address which is loaded into RAR. The TO and FROM fields contain commands to clear the RU flip-flop. This inhibits any further gating into RAR. The first address is obtained by jamming a 1 into the bit 0 position of MAR (incrementing it by 1). The subroutine instructions are then sequenced in the normal manner.

3.56 The last microinstruction in the subroutine contains all 1s in the NA field. When this instruction is gated into MIR, an all-ones detector on the NA field detects the condition and gates the return address stored in RAR into MAR, thereby initiating a normal cycle. At the same time, the RU flip-flops are set. This removes the inhibit on the RAR so that it can be updated in the normal manner. See Fig. 29 and 30 for simplified block diagrams of the microstore microsequence subroutine entry and exit.

D. Data Manipulation Logic

3.57 Data manipulation logic (DML) performs arithmetic and logical operations in the 3A CC. It consists of a group of registers and associated combinational logic which can perform subtraction, addition, and certain logic operations. Other functions are:

- Rotation
- Find low zero
- Register packing and unpacking
- Parity generation on DML results
- Matching of the two DMLs.

Figure 31 is a block diagram of DML 0 and DML 1.

3.58 The three registers in DML are function register (FR), A register (AR) and B register (BR). An operation is normally performed by loading FR with the command word for the function to be performed, and loading the operands into AR and BR. The result of the operation is then available at the output of DML and can be gated onto the gating bus in the usual manner. Some DML operations destroy parity, so parity of the result must be regenerated.

3.59 There are 16 Boolean (logical) operations that can be performed by DML, depending on the value of bits 4, 5, 6 and 7 loaded into FR (Fig. 32). These functions are tabulated in Table J. Bits 1, 2, and 3 of FR control the add function as follows:

- Bit 3—Add long. Performs 20-bit add. Overflow from bit 19 is gated to the DS flip-flop.
- Bit 2—Add short. Performs 16-bit add. Overflow from bit 15 is gated to DS flip-flop.
- Bit 1—Add +1. Adds 1 to result of add long or add short.

These three bits are always zero when a Boolean operation is being performed.

3.60 The rotate is a 2-step operation controlled by the microprogram that rotates the contents of AR by 0 to 15 positions. When the rotate operation is enabled, contents of AR are gated to BR rotated by 0, 4, 8 or 12 positions. Contents of BR are then gated back to AR rotated by 0, 1, 2, or 3 positions. This procedure is illustrated in Fig. 33.

3.61 Find Low Zero Test (FLZT) is enabled by miscellaneous decoder crosspoint MDFLZT to determine the position of the lowest order zero in AR0. The outputs of AR0 go directly to the FLZT logic. This logic develops a binary number which represents the position of the lowest zero in AR0. If no zero are found (AR0 contains all 1s, a signal FLZA0 (FLZ all ones) is produced. This signal is gated to the DS flip flop in MCS by MDFLZT. The binary number representing the position of the low order zero is gated into BR0 by MDFLZT.

3.62 Register packing is the process of placing bits 0-15 of a 20-bit register into positions 0-15 of a 16-bit register, then placing bits 16-19 of the source register into position 0-3 of the adjacent general register. Register unpacking is the reverse of the above procedure.

3.63 It is not necessary to preserve parity within DML since the outputs of DML 0 and DML 1 are matched before being gated onto the gating bus. Any mismatch will set an error flag. DML functions that are controlled by the outputs of the function register are checked for proper parity. Output of the party generator is gated onto the gating bus by a crosspoint which generates DML GB0.

3.64 The DLM matcher is a 20-bit exclusive OR. One input is from the gating bus when the contents of DML 0 are gated onto the bus, while the other input is from DML 1. Any output from the matcher is an error which sets bit 4 of ER when the matcher is enabled by DMLMTEN0.

3.65 DML also matches the output of DML 0 with DML 1. Since the DMLs perform the same operation at the same time, the outputs should also be identical. When the match is performed, contents of DML 0 are gated onto the gating bus. Contents of the bus then are compared with DML 1 in the matcher. An error sets bit 4 in the error register.

3.66 A nonmatch is a fatal error and normally results in stopping the 3A CC and switching the other 3A CC on-line. The fatal error (bit 4) can be diverted to bit PH of the error register (I/O match error) by using the auxiliary decoder on the NA field of MIR. The I/O match error is not a fatal error, so a less stringent recovery procedure can be used. I/O match error is also used to check I/O channel operation.

E. Main Memory Interface

3.67 The main memory interface (Fig. 34) consists of those registers and logic circuits in the 3A CC which provide communication links between the 3A CC and MAS. It consists primarily of the following functional units:

(a) ***Processor Bus Controller (PBC 0 or PBC 1):*** Contains the control logic to interface with the MAS. There is one PBC for My store and one for the Other store.

(b) ***Store Address Register (SAR):*** Contains address of next location to be accessed in MAS.

(c) ***Program Address Register (PA):*** Contains address of location in MAS that is presently being accessed. PA+1 logic is also attached to PA so that PA can be incremented to provide next address in a linear program.

(d) ***Store Data Register (SDR):*** Stores contents of location in MAS that has just been accessed if the contents have been defined as data.

(e) ***Store Instruction Register (SIR):*** Stores contents of location in MAS that has just been accessed if the contents have been defined as an instruction.

(f) ***Instruction Buffer (IB):*** Buffers contents of SIR so that execution of one instruction can be carried out while the next instruction is being gated into SIR.

(g) ***Main Memory Status Register (MMS):*** Contents of MMS define operating modes of MAS. There is an MMS (MMS 0 and MMS 1) in each processor bus controller (PBC 0 and PBC 1).

3.68 The logic necessary to perform a double store read (DSR) or a complement correction when a read parity error is encountered is contained in the MMS. A read parity error causes a store error C (SERC) signal to be returned from MAS instead of the store complete (SCM) signal. If the 3A CC is in the update mode (the other MAS is in standby), SERC will reverse the state of the IDL flip-flop to redirect the read from the errored on-line MAS to the same address in the standby MAS. If the 3A CC is not in the update mode, SERC causes the errored word to be complemented and written back into MAS at the same address.

3.69 Each 256K block in MAS has its own controller (MASC). Each MASC has its own clock and control circuitry and operates autonomously with respect to the 3A CC.

3.70 The primary function of PBC is to receive, buffer, and execute MAS requests from the microcontrol. This function consists of two parts: static conditions that remain fixed during a memory cycle and dynamic conditions that sequence through a series of events during a cycle.

3.71 The static conditions are loaded into MMS by the microcontrol. The static condition MMS bits are:

(a) ***IDL Bit:*** Used to disable (idle) the associated PBC. When IDL bit is clear, PBC will respond to microcontrol memory requests. One IDL bit must always be clear, and the other one must be set.

(b) ***UPD Bit:*** Indicates that the system is in an update mode. Write commands will be issued to both stores. PBC 0 and PBC 1 are both enabled, and IDL bit is overridden.

(c) ***ISO Bit:*** Used to isolate any lower priority unit and prevent it from accessing the memory.

(d) ***BDSR Bit:*** Block double store read bit determines the type of recovery technique to be used when a parity error is detected on a word being read out of memory. If BDSR equals 0, the multiple store read will be used. If BDSR equals 1, complement correction will be used.

(e) ***CW Bit:*** Complement write bit is used to establish a special maintenance condition and for the complement correction subroutine.

When CW bit is set, the last word read out of memory is complemented and written back into memory at the address present on the MAS bus.

(f) ***BEC Bit:*** Block error correction bit inhibits all error correction procedures within the 3A CC that normally occur when an error is detected on a main store read operation.

(g) ***RW Bit:*** Defines whether the memory operation is a read or a write.

(h) ***MM1, MM2:*** Provide a 2-out-of-4 code to define the type of request that has been issued to MAS. These bits are used in conjunction with the RW bit and are sent to MAS on the SC0 through SC3 control leads in the MAS bus. Refer to Tables K and L.

3.72 The dynamic states are controlled by six flip-flops. These flip-flops and their functions are:

(a) ***REQUEST Flip-Flop:*** Indicates that microcontrol has issued a request for a memory cycle. This initiates the sequence of events that results in writing a word in MAS or reading a word from MAS.

(b) ***SEIZE Flip-Flop:*** Indicates that PBC is in the process of issuing a request to MAS. SEIZE flip-flop isolates the bus from lower priority device memory requests so that they do not cause interference with the request in progress. Figure 35 is a simplified logic diagram of the SEIZE flip-flop circuitry.

(c) ***GO Flip-Flop:*** Sends SGO signal to MAS bus to indicate that address, data, and control signals on the bus are stable. When the store finishes the memory cycle in progress, it sends the store complete (SCM) signal to the 3A CC and the PBC responds by clearing GO.

(d) ***DR Flip-Flop:*** Data ready flip-flop is used to inform microprogram control that the requested memory operation is complete. The DR flip-flop is cleared when microprogram control requests a store operation. During the time that DR is clear, microprogram control cannot make further memory cycle requests. When MAS finishes a request, it sets DR. As soon as microcontrol detects that DR is set, it can issue another memory request. Figure 36 is a simplified logic diagram of the DR circuitry.

(e) ***ENREL Flip-Flop:*** Enable release flip-flop indicates that SCM signal has been received and the current PBC cycle is to be terminated. ENREL is set at the same time as DR and is cleared at the end of the PBC cycle by SEIZE when SEIZE is cleared.

(f) ***ID Flip-Flop:*** The instruction-or-data flip-flop identifies the type of memory operation, whether it is an instruction or a data cycle. When ID flip-flop is clear, the word returned from memory is gated into the store data register and is, therefore, a data word. When ID is set, the returned word is an instruction and is gated into the store instruction register.

3.73 Microcontrol performs the following sequence of events during an MAS request:

- Loads MMS
- Issues MAS request
- Tests for completion of the request and gates the retrieved data to the proper register in the 3A CC.

3.74 MMS is loaded directly from the BR register with the miscellaneous crosspoint BRXMMS. This defines the mode of operation until the microcontrol changes the MMS. When MMS is loaded, microcontrol can then issue the MAS request.

3.75 The usual method of issuing an MAS request (instruction fetch) is by using the CACB decoder. The CA and CB bits are part of each word that is brought out of microstore. Normally, the first word of each microinstruction sequence has the CA and CB bits set to the IFETCH state (0, 1 respectively) which initiates the PBC cycle to fetch the next instruction. As soon as execution of one instruction sequence out of microstore is begun, the next instruction is fetched out of MAS into SIR. This is accomplished by IFETCH setting the REQUEST, ID, and PULREQ flip-flops which gate the PA+1 output into SAR. When the DFETCH instruction is executed, ID flip-flop is cleared, and the word read out of main memory is gated into the store data register (SDR). Figure

37 is a simplified logic diagram of the request and ID flip-flop circuitry.

3.76 The SARI and SARD microinstructions are used in those cases when microcontrol is ready to issue another MAS request, but the previous request has not been completed. During execution of the SARI instruction, the data manipulation logic has computed the address in MAS of the next instruction and stored it in a source register. As long as DR is not set, SARI loops and gates the address onto the gating bus on each microcycle. The cleared DR inhibits gating the address into SAR.

3.77 When DR becomes set, the address is loaded into SAR, SARI exits the loop, MAR is incremented by one, and an instruction fetch signal is issued to PBC.

3.78 The SARD is identical to SARI except that the ID flip-flop is clear, thereby causing the returned word to be gated into SDR when DR becomes set.

3.79 The start PBC (SPBC) command is used to initiate a MAS request without affecting the state of ID flip-flop. Figure 37 illustrates the input of the SPBC command to the REQUEST flip-flop.

3.80 The DR flip-flop is set when the PBC receives the SCM signal from MAS. Microcontrol tests DR to determine whether the MAS request has been finished.

3.81 The most frequently used method of testing DR is the all-zero detector on the next address field of MIR. The last instruction of a microcode sequence is identified by having all zeros in the NA field. When this condition is detected and DR is set, the next address that is waiting in SIR is gated into MAR to initiate the next sequence. If DR is not set, the all-zero field will be gated to MAR so that the word at address zero will be retrieved. This word is also all zeros, so the microcontrol will loop at address zero until DR is set.

3.82 The SARI and SARD instructions also test DR, except that the address in NA can be any even-numbered address. The TO field of MIR is decoded, and the crosspoint gates the state of DR into bit 0 of MAR. As long as DR is clear, the MAR address remains unchanged and microcontrol loops on that address. When DR is set, the MAR address is incremented by one and the contents of the next address will be read out.

3.83 The SARD instructions function is exactly the same manner except that the ID flip-flop remains clear, directing the received word into SDR.

3.84 When microcontrol has loaded MMS and issued the MAS request, PBC carries out the operation without any further control from the 3A CC. The PBC, at this point, is completely independent except for the 3A CC clock input.

3.85 Before PBC can continue with processing the MAS request, it must examine the state of the MAS bus. If the bus is in use, PBC must wait until it becomes idle. PBC determines the state of the bus by monitoring store error C (SERC), store go (SGO), and store request (SREQ) signals. When these signals are inactive, indicating an idle bus, PBC sets the SEIZE and GO flip-flops. The MAS request is then gated onto the bus. The setting of GO flip-flop and the resulting SGO signal initiates MASC to execute the memory request.

3.86 During normal operation the on line 3A CC reads from its associated MAS via the PBC. The IDL flip flop associated with the on line PBC will be set, while the other IDL flip flop will be clear. This prevents the read request from being issued to the off line MAS. If a read parity error (SERC) is returned from on line store, the states of the two IDL flip-flops are reversed thereby directing the read request to the off line store.

3.87 The off line MAS responds with a normal read cycle at the same address at which the on line store was read. There is no effect on 3A CC operation except that bit 10 of ER will be set, and the memory cycle length will be two microseconds long instead of one.

3.88 Since memory operation, and therefore the SERC is asynchronous to the 3A CC, SERC is first gated into ERCB buffer. The output of ERCB then goes to either double store read (DSR) logic or to complement correction logic depending on the state of BDSR flip-flop. BDSR is controlled by system software and is usually clear so that DSR is enabled when the system is in an update mode. When the system is not in update mode, BDSR is set and complement correction is used to correct read parity errors.

3.89 When DSR is required, the SERC.DSR flip flop is set at clock phase P2, and cleared at the next P2. The output of SERC.DSR sets SERCROS flip flop. Outputs of SERC.DSR and SERC.ROS are combined to produce EROSERC which reverses the states of the IDL flip flops. The output of SERC.DSR also produces TSERCBOA and TSERCBOB which gates the errored data from the on line store into the DK register, and the store address into AK. In addition, the good data from the off-line store is also gated into the DI register as well as into SIR or SDR.

3.90 The PBC now waits for the store complete (SCM) signal from MAS which indicates that MAS has terminated the operation. Every microcycle, the state of the SCM lead is gated to the enable SCM (ENSCM) flip-flop by clock phase P2. The next clock phase, P3, gates ENSCM to ENREL and DR flip flops. Therefore, on the next P2 after the arrival of SCM, ENSCM is set. DR and ENREL are set 37 nanoseconds later by P3, thus completing the request (Fig. 38). Setting of ENREL clears GO which then clears SGO. When SGO disappears, the MASC clears SCM. This resets the PBC and SEIZE flip-flop. MASC is then ready to issue another request.

F. Input/Output Channels

3.91 The 3A CC communicates with peripheral devices by means of parallel gating buses and a serial I/O interface. The 3A CC can address up to 20 main channels. Up to 20 serial subchannels can be addressed through each main channel or as many as 8 parallel subchannels may be addressed through each main channel. If DMA is connected to a main channel, four parallel subchannels can be addressed through DMA. Each application using the 3A Processor may assign these channels according to its own requirements.

Serial Channels

3.92 All the main channels are identical, and each main channel consists of I/O status register (IOS), I/O data register (IOD) start code register (STRT), sequence and control logic, error check circuitry, and gating paths in the 3A CC.

3.93 A subchannel consists of a dedicated cable driver and cable receiver. Each subchannel has its own address within its main channel. A main channel can communicate with only one of its subchannels at a time since the subchannels share the common address and control circuits.

3.94 I/O serial channels transfer data to and from peripheral devices in a serial format at a rate of 6.67 megabits per second. Serial format means that the bits are transmitted one at a time. The bits are transmitted as bipolar ac pulses whose phase determines whether the pulse is a 1 or a 0. The ac transmission permits longer signal path lengths without serious degradation of the signal, while the bipolar pulses maintain average signal level at a zero volt level with respect to ground.

3.95 Figure 39 shows the interface between the 3A CC and the serial I/O channels, Fig. 40 depicts typical subchannel connections, and Table M lists typical subchannel assignments. Subchannels will be assigned according to parent system requirements.

3.96 The 3A CC controls the I/O channels with nine miscellaneous decoder signals (Table N). These signals operate on the main channel whose address is contained in R-9, bits 10 through 15 (MCS field). Figure 41 is a functional block diagram of the I/O main channel, and Fig. 42 is a simplified logic diagram of the bipolar transmitters and receivers.

3.97 Serial data pulses transmitted over the I/O channel between the 3A CC and the peripheral device are in bipolar form. Each data bit is represented by a pair of pulses of equal amplitude and opposite polarity. The phase relationship of the pulses determines whether the data bit is a 1 or a 0.

3.98 Contents of 3A CC registers R9 and R10 are gated into IOS and IOD, respectively. The start channel microinstruction terminates the gating period, sets the normal start code (011) into the start register, and initiates the sequencer.

3.99 In the normal mode of operation, a 21-bit serial message is transmitted to the designated peripheral device, and a 21-bit serial message is received in response. These messages contain a 3-bit start code and an 18-bit data field. The data field includes two parity bits which maintain odd parity over their respective 8-bit data fields.

3.100 The two legal start codes for an I/O message are 011 and 101. The leading bit of the start code must always be a 1 so that the receiver can identify the start of a message. One bit must be a 0 to be sure that the shift register can transmit both 1s and 0s, and the remaining bit maintains even parity over the start code field. (Refer to Table O).

3.101 Start code 011 is called the normal start code, and 101 is called the maintenance start code although the two codes are handled in the same manner.

3.102 Incoming I/O messages contain the same start code formats as output messages. Illegal start codes are detected and error flags set to indicate the error.

3.103 The 3A CC interfaces with the main channel by means of 3A CC general registers R9, R10, and R11. Register R9 is the I/O control buffer and contains the main and subchannel addresses and a 4-bit control field. Each main channel is assigned a unique main channel select (MCS) 3-out-of-6 code. This code is contained in bits 10 through 15 of R9 (Table P).

3.104 Register R10 contains data to be transmitted through the main channel and the subchannel to the peripheral device. Register R11 contains data that is received from the peripheral device. Each of the three registers communicates with the 3A CC via dedicated gating buses. Gating is accomplished in the same manner as the other general registers in the 3A CC.

3.105 Contents of R9, R10, and the IOD in the selected main channel are always present on the output buses. Gating from the bus to the selected peripheral device is performed at the device. This assures that the data is present and stable before the destination registers are enabled to receive it.

3.106 The IOD has normal gating access to the R11 gating bus, controlled by MCS enables. The IOS also has access to the R11 bus for maintenance purposes. This access is allowed only by a sequence of microinstructions from microcontrol.

3.107 Refer to Fig. 43 for a block diagram of the IOS, and to Fig. 44 for a block diagram of the IOD. Table Q defines the significance of each bit in IOS.

3.108 The IOS is a 12-bit register which consists of two 6-bit fields: the subchannel select (SCS) field and sequencer control (SCTL) field. The SCS field is loaded from 3A CC register R9 and contains the address of the subchannel to be used. The SCTL field may be set to specific 3-out-of-6 codes which identify the start of a control sequence to be followed. Bits 0 through 3 of the SCTL may be loaded from R9. (Refer to Fig. 43.) In this case, bits 4 and 5 are set to 1 and 0, respectively, in order to form a valid 3-out-of-6 code.

3.109 The IOD is an 18-bit shift register which functions as the transmit receive buffer for the data portion of an I/O message. It may be parallel-loaded from 3A CC register R10, and its contents may be loaded into 3A CC register R11.

3.110 The STRT register is a 3-bit shift register which is attached to the low end of the IOD. It is set to specific bit patterns by the control signals which initiate the control sequences to be followed. STRT is an internal IOC register and is not externally accessible.

3.111 The I/O sequencer controls the entire transmit/receive operation once it is initiated by a command from the 3A CC.

3.112 The IOC can perform two major transmit sequences: transmit normal and transmit maintenance. It passes through four major states during the execution of a message sequence: idle, transmit, receive, and message complete. IOC sequence logic consists of a state code generator which generates a series of 3-out-of-6 codes identifying the sequences and the states within the sequence, a state decoder which translates the 3-out-of-6 codes to a 1-out-of-20 set of signals, and the state change logic. The state code generator is bits 0 through 5 of the IOS register (the SCTL field). This generator is a circular shift register which produces a sequence of codes in response to the shift signals. A particular control sequence is initiated by the first code in the sequence being set into the SCTL field. The next state codes are generated by shifting the field left one bit position at a time.

3.113 The state change logic in the IOC sequencer generates signals which initiate the SCTL field shifts described in the preceding paragraph. Figure 45 is a simplified diagram of state change logic, and Fig. 46 is the state diagram depicting states through which the sequencer cycles. Each state change requires that a state code be present and that a particular set of external conditions be satisfied. When all these conditions are present, the ADVSTA signal is produced which causes the SCTL shift register to be shifted to the next state.

3.114 A send I/O (SIO) main memory instruction initiates a series of microstore program instructions which initialize the main channel specified by the contents of the MCS field of 3A CC register R9. The SCTL field is set to the idle state (000111), and the main channel is enabled to accept additional microinstructions.

3.115 When the sequencer receives the MD4STCH start channel signal, it begins to transmit the data stored in the IOD register. Bit 0 is transmitted first, followed in sequence by the remaining two bits in the start code and the 18-bit data message. As each bit is transmitted and contents of the IOD shifted to the right, a trailing zero is added to the left end of the IOD. This process continues until the 21-bit IOD/STRT register contains all zeros. When the all-zero detector detects this condition, the sequencer is advanced to the receive state while continuing to transmit zeros.

3.116 When IOC advances to the receive state, it enables the serial input to the high end of the IOD. The received message is shifted in bit-by-bit until the leading 1 of the start code is detected in bit 0 position of the IOD/STRT register. At that time, the state change logic inhibits further shift pulses, resynchronizes to the 3A CC clock, and advances the sequencer to the message complete state.

3.117 The message complete halts timing signals to the peripheral device, decodes the 3-bit start code, and sets the appropriate 3A CC flag. After completion of the transmit sequence, IOC remains in the message complete state until IOD is unloaded and initialized by program action (MDOICHO command from microcontrol).

CTI/Power Unit

3.118 The CTI (collector diffusion isolation [CDI] to transistor-transistor logic [TTL] interface) unit is located at levels 62 to 70 of the 3A auxiliary processor frame. It provides the logic level shifting between 3-volt CDI integrated circuits used in the 3A CC and 5-volt transistor-transistor logic integrated circuits used in the direct memory access (DMA) and parallel channel (PCH) units. In addition, the CTI unit contains three type 132M power supplies which furnish 5-volt power to the DMA, PCH 0, and PCH 1, as well as fuses and power alarm circuits. Figure 47 is a front view of the CTI unit, and Fig. 48 is a block diagram.

3.119 Inputs to the CTI are gated from R9 and R10 gating buses in the 3A CC by miscellaneous decoder crosspoints. Logic levels of these inputs are shifted to the higher level required by DMA and PCH and are placed on the DMA and PCH input bus.

3.120 Information being returned from DMA and PCH to the 3A CC is shifted to lower voltage levels required by the 3A CC logic in the CTI and is gated onto the R11 gating bus.

3.121 The CTI unit requires +3 volt power which is supplied by a 3-volt dc-to-dc converter located in the 3A auxiliary processor frame power unit. When a DMA unit is installed, the 132M power supply which provides +5 volts for DMA also supplies +5 volts for the CTI unit. In installations that do not use DMA, CTI is connected to the PCH 0 +5 volt power supply.

3.122 Additional information on the CTI, RCH, and DMA units is contained in Section 254-100-130.

Parallel Channels

3.123 In addition to the serial I/O channel described in 3.79, the 3A CC has the capability of servicing high speed peripheral devices through a parallel I/O channel. Space is provided in the 3A processor frame for two parallel channel units, PCH 0 and PCH 1, each of which can support up to eight subparallel channels. Each subparallel channel can control up to 16 peripheral devices.

3.124 A DMA channel can be installed to provide a peripheral device such as a magnetic tape transport with the capability of transferring blocks of data to or from main store without requiring intervention of the 3A CC. Figure 49 depicts the I/O structure of the 3A processor.

3.125 General registers R9, R10, and R11 serve the same purpose in the parallel channel as in the serial channel. R9 contains the address of the selected channel and peripheral device, R10 contains data to be transmitted to the device, and R11 receives data returned from the peripheral device. Figure 50 depicts R9 address format, Table P lists the channel number assignments, and Table Q lists the MPCH status word format. Figure 51 illustrates the signal interface between the 3A CC and the CTI unit.

G. Maintenance Channel Controller

3.126 The maintenance channel (Fig. 52) consists of the following functional groups:

(a) MCHB - The maintenance channel buffer register (MCHB) is a 22-bit register which serves as a temporary storage register for buffering incoming or outgoing data. Its contents may be gated to or from any internal register within the processor. The MCHB is also used in the originating MCH controller to buffer all returned data.

(b) MCHTR - The maintenance channel transmit receive register (MCHTR) is a 22-bit shift register which may be parallel-loaded from the program gating bus via miscellaneous decoder control signal PGBXMCHT/R. This register accepts the data portion of the MCH message, and its contents may be gated to the MCHB register via control point 1 from the command decoder. In addition, there are input paths into this register from MIRL, MIRI, CC error register, system status register, and miscellaneous bits. These bits are activated by outputs from the command decoder, permitting direct access to a vital set of registers within the processor, the contents of which are necessary to determine the sanity of the system.

(c) MCHC - The maintenance channel command register (MCHC) is an 8-bit shift register which has parallel access to and from the program gating bus (via gating signals MCCXPGB and PGBXMCC, respectively). This register contains the start bit and a 3-out-of-7 operation code for the MCH order to be executed. The operation code portion of this register also feeds the command decoder which decodes the command portion of a received message.

(d) Main Sequencer - The main sequencer (Fig. 53) consists of a 3-bit state code generator (rotating shift register), a state decoder, a 2-bit MCH state register, and the logic necessary to produce state change signals and control the overall operation of the channel. The contents of the state code generator always define the state of the channel (Table A). This sequencer handles all normal transmissions and receptions and operates autonomously once started (by activating the STMCH miscellaneous decoder crosspoint). The maintenance channel may be overridden at any time by initializing the sequencer and returning it to the rest state (MONITOR).

(e) Switch Sequencer - The switch sequencer, when activated, is responsible for controlling the automatic transmission of a switch message to the other MCH controller. It has the ability to override the main sequencer and depends on the 3A CC for clock signals and the activation signal. The activation signal is generated by the hardware initialization errors in the 3A CC error register or directly via a microinstruction. Figure 54 is a simplified logic diagram of the switch sequencer, and Figure 55 is the state diagram.

(f) Message Format - The MCH message format is a 30-bit field containing a start bit, a 7-bit command field, and a 22-bit data field. The data is in the format as data on the 20-bit processor gating bus, ie, 20 bits of data and 2 parity bits.

3.127 The MCH controller is completely autonomous except for the start-up procedure. In normal operation, a transmission is initiated by software, the command register (MCHC) and transmit/receive register (MCHTR) are loaded, the start bit (STRT) is set, and the MDSTMCHO microinstruction is issued. When the command register is loaded, the controller is removed from the IDLE state at the same time by the sequencer being set to the master transmit (MXMIT) state.

3.128 As soon as the MDSTMCHO instruction is received, data stored in the MCHTR is transmitted a bit at a time (serial transmission), beginning with the start bit. As each bit is transmitted, data in the register is shifted right one position and the leftmost bit position is filled with a zero. This process continues until the all-zero detector across the MCHC and MCHTR detects all zeros. This condition causes the sequencer to advance to the master receive (MREC) state. A continuous stream of zeros is still being transmitted to the other 3A CC. As incoming serial data begins to enter the MCHTR of the receiving 3A CC, contents of the register are shifted right as each data bit is received until the start bit is detected in the start bit position of the MCHC. This inhibits any further clocking in of data into the MCHTR and advances the slave controller to the slave execute (SXEQ) state. The command in the command field of the MCHC is decoded and executed. When the end of command signal is received, the slave controller sets the return message code in the MCHC, sets the start bit to 1, checks for errors, and advances the slave controller to the slave transmit (SXMIT) state.

3.129 The master controller is still in the MREC state and is still transmitting zeros. The incoming serial message from the slave controller is clocked into the master MCHTR until the start bit is detected in the start bit position. When the controller detects the start bit in its proper position, it gates the MCHTR onto the MCHB and sets the return (RTN) bit in the MCH status register. If an error signal has been returned, the error list (MCHERR) in the MCH status register is set. The stream of zeros is also stopped and then the controller advances to the WAIT state. The slave controller detects the termination of the zero signals and goes to the MONITOR state. The stream of zeros from the slave controller is stopped when the controller enters the MONITOR state. The master controller then detects the no-signal condition and initializes to the MONITOR state.

3.130 Data on the maintenance channel is transmitted in serial bipolar form in the same manner as the I/O channel (refer to the discussion on the I/O channel, 3.91). The bipolar pulses are generated, transmitted, and regenerated into data pulses as in the I/O circuits.

3.131 The command translator decodes the commands in bits 0 through 6 of the MCHC. The command field contains a 3-out-of-7 code.

3.132 There are two types of MCH commands: those which require the system clock in the slave 3A CC and those which can be executed without use of the system clock. As a result of these two types of commands, the slave sequence has two modes of operation in the SXEQ state: the bit stream timing mode (BSTM), and the clock timing mode (CTM). BST mode derives its timing from the incoming bit stream while CT mode uses the system clock.

3.133 The first bit of the command field is the mode select bit. All BSTM commands have the first bit set to 1, while the CTM commands have the first bit set to 0. Table R is a description of the maintenance channel commands.

3.134 The second operational mode of the MCH is the automatic generation and transmission of a switch message. When a fatal error occurs in the on-line 3A CC, system reconfiguration is necessary to remove it from service and functionally replace it with the off-line unit.

3.135 Detection of the fatal error causes a switch signal to be generated and sent to the switch sequencer. The MCH responds automatically with a switch message to the off-line 3A CC which causes it to change to on-line status and assume on-line processing tasks.

3.136 A fatal error may occur at any time; therefore, the switch signal exercises absolute priority to abort an in-progress MCH transmission, initialize the main sequencer, set the MCHC to the switch code, set the start bit, and initiate the switch message transmission without any external control. Each of these functions is performed by the switch sequencer (Fig. 54).

3.137 Normally, the switch sequencer is in the ACTIVE state (Fig. 55). A switch signal (fatal error or software switch command) advances the sequencer to the INITIALIZE MAIN SEQUENCER state. After the main sequencer is initialized, the switch sequencer moves to the LOAD AND GO state, in which it sets MCHC to the switch code (00001111), sets the start bit, sets the main sequence start code generator to the MXMIT state, and activates the main sequencer.

3.138 The main sequencer now assumes control of the transmission process and transmits the switch message to the off-line 3A CC. During transmission, the switch sequencer advances to the I/O DISABLE state where it remains until an idle switch sequencer (IDLSWSQ) message is received from the 3A CC that has just been switched on-line. This state prevents transmission of repetitive switch messages.

3.139 The main sequencer of the formerly on-line 3A CC advances to the MREC state after transmitting the switch message, where it waits, expecting a normal return message which will advance it to the MONITOR state.

3.140 The newly on-line 3A CC is initialized by the switch message except that its MCH is not reset until a microstore initialization sequence is performed. This delay permits the newly on-line 3A CC to begin transmitting a stream of zeros to the formerly on-line 3A CC. When the MCH in the newly on-line 3A CC is reset by the microcode initialization (thereby placing it in MONITOR state), it halts transmission of zeros. Absence of incoming data (stream of zeros) is detected by the off-line unit, resulting in a reset and an advance to the MONITOR state.

3.141 Both 3A CCs are now in the MONITOR state, and normal communication between them may be resumed.

3.142 The MCH performs the following error checks:

(a) Loop test

(b) Multiple output check

(c) Simultaneous transmission check

(d) Parity check.

3.143 ***Loop Test:*** A loop test is inherent in every MCH transmission. The correct operation of the test ensures that a message was transmitted from the master controller, received by the slave controller where a return message was formed, and transmitted back to the master controller. The successful completion of this sequence causes the return bit in the MCH status register to be set. This bit may later be checked by the active 3A CC.

3.144 ***Multiple Output Check:*** A multiple output check of the command decoder is performed in the slave controller upon decoding the command field of the received message. A command field with less than three 1s will result in no output from the decoder. This situation can be detected functionally; ie, the desired command is not performed. If a multiple output is detected, an indication to this effect is included in the return message transmitted to the master controller. The master channel controller decodes this field and sets the error bit in its MCH status register. This bit must be detected by software as no automatic error indication is generated.

3.145 ***Simultaneous Transmission Check:*** Either MCH controller is free to initiate a transmission independent of the state of the other maintenance channel. The 3A CC assumes that if its own MCH is not busy (ie, in the MONITOR state), then the other MCH is not busy. Thus, it is possible that both MCH controllers could initiate simultaneous transmissions. If this occurs, an incoming bit stream will be detected by each MCH controller while it is in the MXMIT state. Detection of these dual bit streams will cause bit 1 in the MCH status register to be set. Neither controller has priority over the other, so when both attempt to transmit at the same time neither can receive and both transmissions will be lost. Both controllers will wait in the receive state (MREC) with their error status bits set and their return status bits cleared. They will remain in MREC until the software program interrogates the MCH status register to determine MCH status (Table S) and issues a reset command to return them to the MONITOR state.

3.146 ***Parity:*** Parity is maintained within the data field of each message and is checked via the normal parity checker when the data is gated onto the processor gating bus. Parity is also maintained over the MCHC. Valid data in this field consists of a 3-out-of-7 command plus a start bit resulting in even parity. When this field is gated to the processor gating bus, both P_H and P_L are forced to 1s to maintain odd parity over the two parity fields.

H. Console Functions

3.147 The 3A CC control panel and the related console functions (Fig. 56) provide one of the major means of communicating with the 3A CC. The four major console functions are:

- To receive control signals from and send status signals to the system status panel
- To provide hardware initialization including enabling of power turn-on and turn-off
- To permit the 3A CCs to be forced on-line or off-line, and to enable the maintenance reset functions
- To buffer the external interrupts.

3.148 The major items of console function hardware include the display buffer (DB), address input register (AI), address mask register (AK), data input register (DI), data mask register (DK), control panel status register (CPSR), address matcher (AM), and data matcher (DM).

(a) ***Display Buffer (DB):*** The DB is a 22-bit register used to store data to be displayed by the indicator lamps on the 3A CC control panel. It can be program loaded from the gating bus or from the program address bus, or directly from the control panel switches.

(b) ***Address Input Register (AI):*** The AI is a 22-bit register used to store a memory address. Its contents are combined with the contents of the address mask (AK) register to produce a 22-bit address which can be matched with the MAS address in the SAR.

(c) ***Address Mask Register (AK):*** The AK is a 22-bit register which can be loaded with a mask word to be used in conjunction with the contents of the AI register when an address mask is desired.

(d) ***Data Input Register (DI):*** The DI is an 18-bit register whose contents are combined with the contents of the DK register. This produces an 18-bit word which can be matched with the data on MASB at the end of each main store request.

(e) ***Data Mask Register (DK):*** Same as AK (3.124) except that DK is 18 bits long.

(f) ***Address Matcher (AM):*** The AM compares the address generated by the combination of AI and AK registers with the contents of the SAR. When a match is detected and the address match enable (bit 0) in the SS register is set, the matcher sets bit 3 in the IS register to cause a 3A CC interrupt.

(g) ***Data Matcher (DM):*** The DM compares the data word generated by the combination of DI and DK registers with the information on the MASB at the end of a store request. When a match occurs and DM is enabled by the setting of the data match enable (bit 4) in the SS register, the matcher sets bit 3 in the IS register to cause a 3A CC interrupt.

3.149 The 3A CC control panel can be operated only when it is in manual mode and the 3A CC is off-line or is in test mode. Figure 57 is a simplified logic diagram of the control panel interface logic.

3.150 Most control panel functions are executed out of a panel-halt loop. When the 3A CC is off-line and the MANUAL switch on the panel is depressed, operation of the RESET CIRCUITS switch will enable the panel-halt microcode sequence.

3.151 This loop continuously interrogates the state of the panel interrupt bit (interrupt level 13). When the interrupt bit is set by operating the EXECUTE switch, the states of the panel switches are interrogated, and the functions designated by the switches are performed. The REJECT indicator lamp will be lighted when an invalid function is requested. After the requested function is executed, the 3A CC lights the panel HALTED lamp and returns to the panel halt loop when the panel HALT switch is activated. When the panel HALT switch is not activated, activation of the EXECUTE switch causes the address in the PA register to be loaded into the SAR and the main store instruction at the address to be executed.

3.152 The control panel switches are grouped into three registers. Most of the control switches are in register 1, the register select switches are in register 2, and the data input switches are in register 3. Table T lists the switches and their groupings.

3.153 In normal operation, the microcontrol gates the register 1 states to the data bus, then to the DML where the states are tested to determine what function has been requested. An invalid function request causes the microcontrol to light the REJECT lamp on the control panel and to gate all 1s to the data buffer, thereby lighting all data display lamps on the control panel.

3.154 A valid request results in the states of the register 2 switches being gated to the data buffer and then to the instruction buffer. The microcontrol decodes the switch states to determine which register has been selected, then establishes the to-from decoder fields to enable the gating paths to the register.

3.155 Next, the states of the data input switches are gated into the data buffer and into the register that has been previously selected. The data lamps on the control panel are also lighted since they are driven directly from the display buffer (Fig. 58).

3.156 Parity on the input data may be generated automatically or manually. This mode is selected by activating the MANUAL PARITY switch in switch register 1.

3.157 Power can be removed from the 3A CC only under specified conditions. The 3A CC must be in manual mode and in test mode, or manual mode and either locked off-line or disabled in order for the power to be turned off.

3.158 When the 3A CC is off and the power switch is turned on, the power-on sequence is initiated. Processor frame power circuits are initialized; power is applied to circuit pack FA1034; DISABLED, ISC, and CC flip-flops are cleared; and the program timer is also cleared. In addition, an MRF is generated which causes the microcontrol to enter the panel halt loop.

3.159 Maintenance reset functions (MRF) can be initiated by a number of system conditions, and results in system reinitialization. These conditions are:

- MCH MRF
- Reset circuits function from CC panel
- System status panel MRF
- Program timer time-out
- Hardware MRF error
- Branch allowed (BA) error
- Write protect error
- My store fast time-out error.

3.160 When an MRF is detected, the following sequence of events is initiated:

- Clear MCH switch sequencer
- Set block hardware check bit
- Set block timer check bit
- Clear microinterpret (MINT) bit
- Clear maintenance state register
- Initialize I/O enable
- Clear stop flip-flop
- Load starting address of MRF sequence [O(277)] into the MAR
- Initialize 3A CC clock to phase P3.

3.161 Certain MRFs cause the 3A CC to halt and bring the other 3A CC on-line (stop and switch) if the on-line processor is not locked on-line. If the processor is locked on-line, the MRF will take place in the processor that originates it. These functions are:

- Hardware fatal error. (All major 3A CC and main store errors are in this category.)
- Stop and switch microinstruction.
- Idle MCH.

3.162 Stop and switch operation is carried out by the maintenance channel. The 3A CC microcontrol initiates a switch message to the MCH, sets the stop flip-flop, and clears the microinterpret flip-flop.

3.163 Figure 59 is a simplified logic diagram of the initialization and MRF circuitry.

I. Data Transfer Check Circuits

3.164 The data transfer check circuits perform a number of check functions for the 3A CC. When enabled, they verify correct parity of information placed on the program gating bus, verify valid four-out-of-eight codes in the MIR TO and FROM fields, and assure that only one valid main serial I/O channel is addressed on execution of an I/O operation.

3.165 ***Organization:*** There is a functional organization of the data transfer check circuits. They are combined for discussion purposes only and consist of five separate check circuits.

(a) TO checker

(b) FROM checker

(c) Gating bus parity checker

(d) Special TO checker maintenance state

(e) I/O main channel decoder check circuit.

3.166 ***Gating Bus (GB) Parity Checker:*** The purpose of this circuit is to verify that, on every microcycle, parity of the data word gated onto the GB is correct. Correct parity means that odd parity exists over bits 0 through 7 plus the P_L bit, checked as one group, and over bits 8 through 15 plus the P_H bit, checked as another group.

3.167 This circuit consists of a 22-bit buffer register whose outputs are fed to a parity check circuit. The two outputs of the parity check circuit are fed to the 3A CC error register where they will set either bit 3 or P_H.

3.168 The GB parity checker operates by strobing the contents of GB into its buffer register at clock phase P3 of every microcycle. By placing the GB contents in a buffer register, sufficient time is allowed for the contents to ripple through the parity trees. The output of the parity check circuits is not gated to the error register until clock phase P2 of the next microcycle. This circuit also contains logic to allow inhibiting the GB parity check. This is accomplished by activating a control signal which in turn buffers the request in a buffer flip-flop. These control functions are used to indicate that incorrect parity may be placed on GB. This provides the ability to access registers via GB whose contents for one reason or another may contain invalid parity. Inhibiting of the GB parity check is normally valid for only one microcycle. However, there are some control signals which permanently disable the GB parity check. For example, the block bus parity check bit in SS register is typically set during initialization to allow the machine to be initialized and to purge registers of possible bad parity throughout the machine. The following is a list of control functions that can inhibit the output GB parity checker.

(a) ***Stop*** flip-flop (bit 14 of the SS register) being set will inhibit the output of the GB parity check circuits.

(b) ***Block bus parity*** check flip-flop (1 bit 16 of SS register) being set will inhibit the output GB parity check circuit until it is cleared.

(c) ***Freeze*** flip-flop (bit 6 of the maintenance state register) being set will inhibit the output of the GB parity check circuit until it is cleared.

(d) The NOP microinstruction causes the microcontrol to be inactive for one microcycle. This results in no data (effectively all zeros) being gated to the GB. The GB parity checker interprets this condition as a parity error; therefore, the checker must be inhibited for the NOP microcycle to prevent an erroneous parity error signal from being generated.

(e) Activation of a miscellaneous decoder crosspoint must also inhibit the GB parity checker because data is not gated to the GB during the microcycle. The lack of data is interpreted as a field of zeros by the parity checker, and it generates a parity error if it is not inhibited.

(f) Activation of ***SIRC*** or ***SDRC*** from-field crosspoints will inhibit the output of the GB parity check circuits. This is necessary since the crosspoints gate data from the SIR or SDR, respectively, to the GB only if a memory operation is complete. If the memory operation is not complete, no data is gated from SIR to GB, and the GB parity checker must be inhibited.

(g) Activation of the next address-block parity check signal from the next address field auxiliary decoder will inhibit the output of the

GB parity check circuit for one microcycle. This typically allows a register with bad parity to be gated to the DML where parity can be computed.

There is also another set of control signals that, instead of inhibiting the GB parity check, divert the error output to bit PH of the error register. This diversion permits more graceful error recovery rather than the normal stop and switch which will occur on a GB parity error. This diverted error is called an I/O parity error and will cause an error interrupt. The control sources that divert the GB parity error to an I/O parity error are enabled from the MIR next address field auxiliary decoder.

4. THEORY—POWER AND ALARM CIRCUITS

4.01 The station power furnishes -48 vdc to the processor frame as primary power for dc-to-dc converters which provide 3-volt and 5-volt power for the logic circuits. The frame also receives +24 vdc which is used in the alarm system, for light emitting diodes (LEDs) used as indicator lamps, and the +12 volt reference voltage converter.

4.02 The 3-volt and 5-volt dc-to-dc converters are conventional transistorized power oscillators which convert the -48 vdc into ac which is rectified, filtered, and regulated at the 3- or 5-volt level.

4.03 The 1A technology circuit packs used in the 3A CC require +3 vdc for operation. All of the transistor-transistor logic (TTL) circuit elements use +5 vdc. Logic networks provide the voltage level change necessary whenever the 1A technology circuits must interface with the TTL circuits.

4.04 The 3-volt converter provides +3 vdc nominally regulated to ±1 percent. Power alarms should be activated at a high voltage of +3.07 to 3.11 volts. Low voltage alarm should activate at 2.95 through 2.99 volts. Overvoltage shutdown should occur at a maximum of 4.3 volts, while overcurrent protection should activate at 5.25 ±0.75 amperes. The output voltage is also clamped to a maximum voltage of +5.2 volts by a zener diode shunted across the output.

4.05 The 5-volt converter provides +5 vdc nominally regulated to 5.0 ±0.110 volts. The high voltage power alarm should occur between 5.16 and 5.36 volts. The low voltage alarm should activate between 4.66 and 4.86 volts. The overvoltage shutdown occurs at a maximum of 6.0 volts. Overcurrent protection occurs at 5.22 ±.84 amperes. The output voltage is clamped to a maximum of 7.0 volts by a zener diode shunted across the output.

4.06 The 12-volt regulator provides a +12 volt reference which is used by the 3- and 5-volt regulators to establish the overvoltage and overcurrent alarm and shutdown levels. A +3 volt reference voltage is also supplied to the converters to accomplish the output voltage regulation. Figure 60 is a functional block diagram of the power system.

4.07 When the output voltages of the converters exceed the limits, the power alarm activates from 5 to 200 milliseconds later. If the voltage returns to its in-tolerance level, the alarm is removed within 50 milliseconds. Facilities are provided to test the alarm circuitry either manually or automatically.

4.08 Refer to Section 254-100-140 for additional information on the power and alarm subsystems.

5. MAINTENANCE

5.01 The 3A CC is designed as a redundant self-checking processor. The bit slicing technique used in the memory and many registers insures that undetected bit failures will be highly unlikely because two circuit packs would have to fail at the same time in order to produce compensating failures.

5.02 By using the maintenance channel and the microinterpret mode of operation, one 3A CC can cause the other to execute instructions and report results for diagnostic purposes. The diagnostic programs stored on magnetic tape can also be run under either local or remote control.

5.03 The microinterpret mode (Fig. 61) is designed to minimize the amount of microcode needed for maintenance testing. This is accomplished by storing the maintenance microsequences in main store instead of microstore. When the microinterpret mode is invoked, the maintenance microinstructions are fetched from main store and loaded into SIR. The instructions are gated from SIR into MIR where they are decoded as in any other microinstruction.

5.04 The microinterpret mode is initiated by an MI (multiple cycle) or MIS (single cycle) command. The two commands are almost identical except that MIS begins by zeroing the TR1 flip-flop while MI sets it. Microstore address O(0426) is the entry point for an MIS command, and address O(0427) is the entry for an MI. Figure 62 is a flow diagram of the execution of the MIS and MI commands, and Fig. 61 is a simplified logic diagram of the microinterpret gating.

6. REFERENCES

6.01 The following documents contain additional information applicable to the 3A CC.

254-100-150 Main Store and Supplemental Store Theory and Description Common Systems

250-100-130 I/O Interfaces, Common Systems

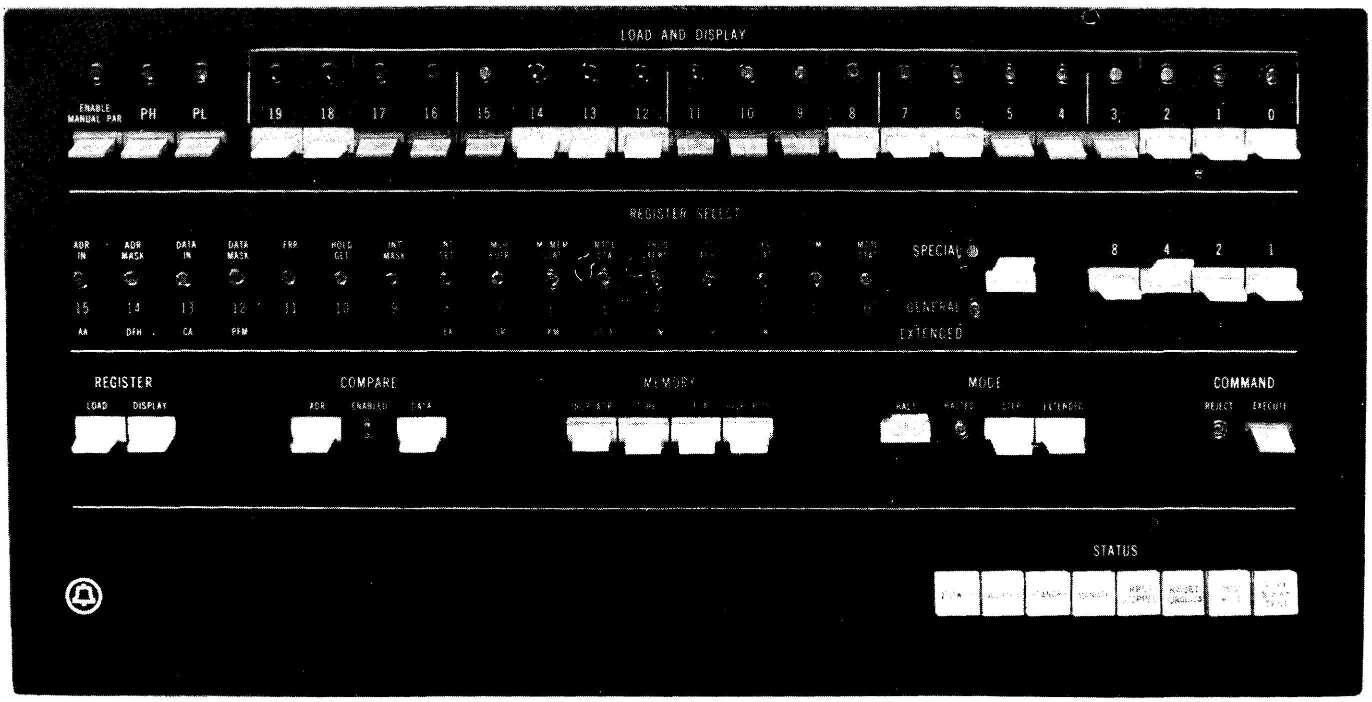


Fig. 1—3A CC Control Panel

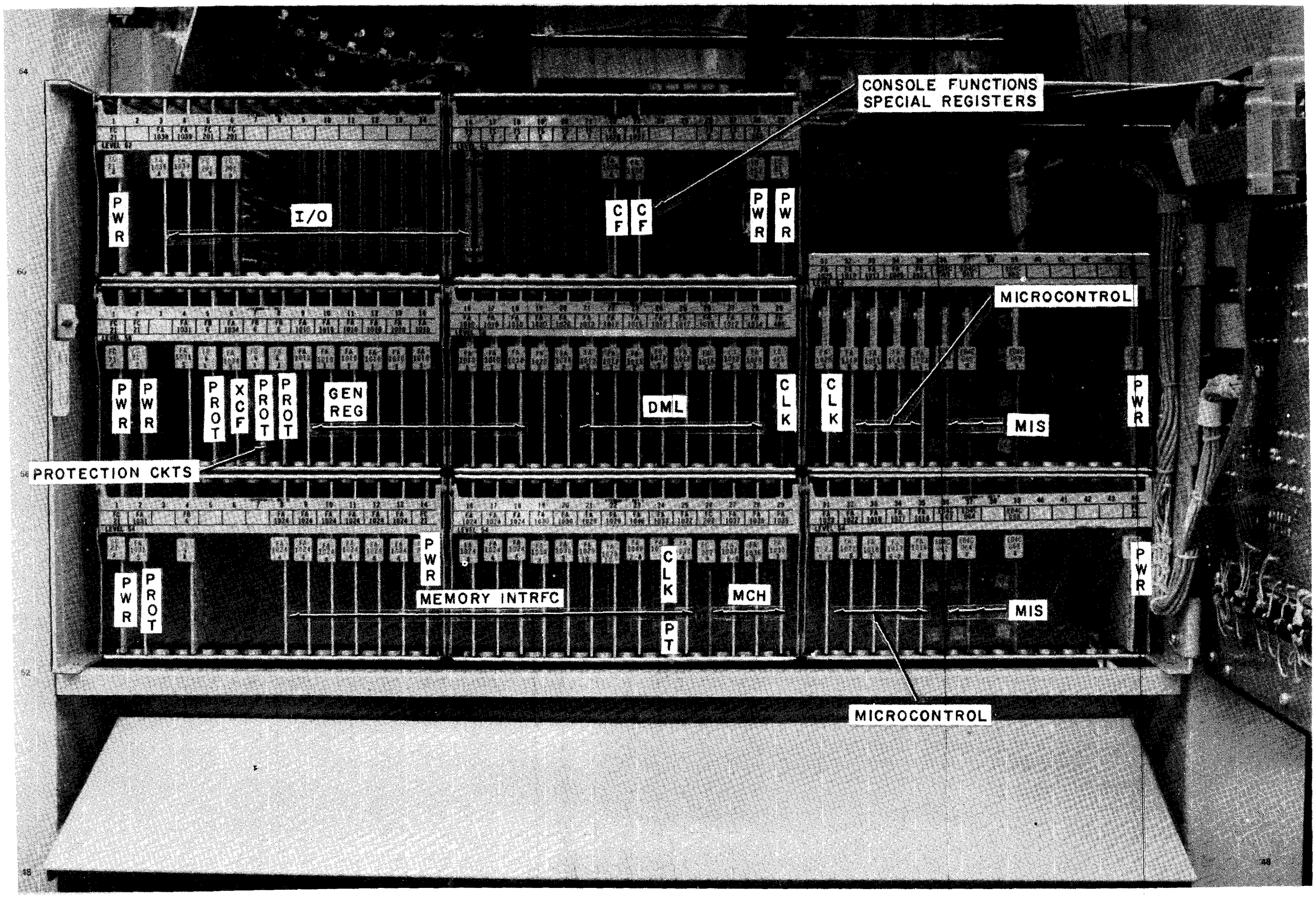


Fig. 2—3A Central Control (Panel Open)

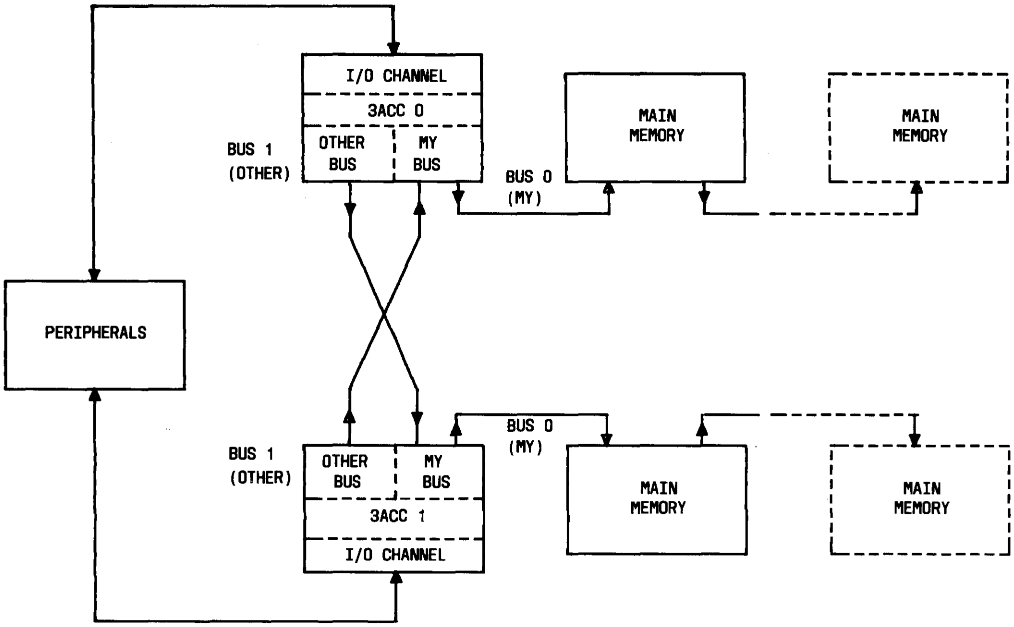


Fig. 3—3A CC Duplex Configuration

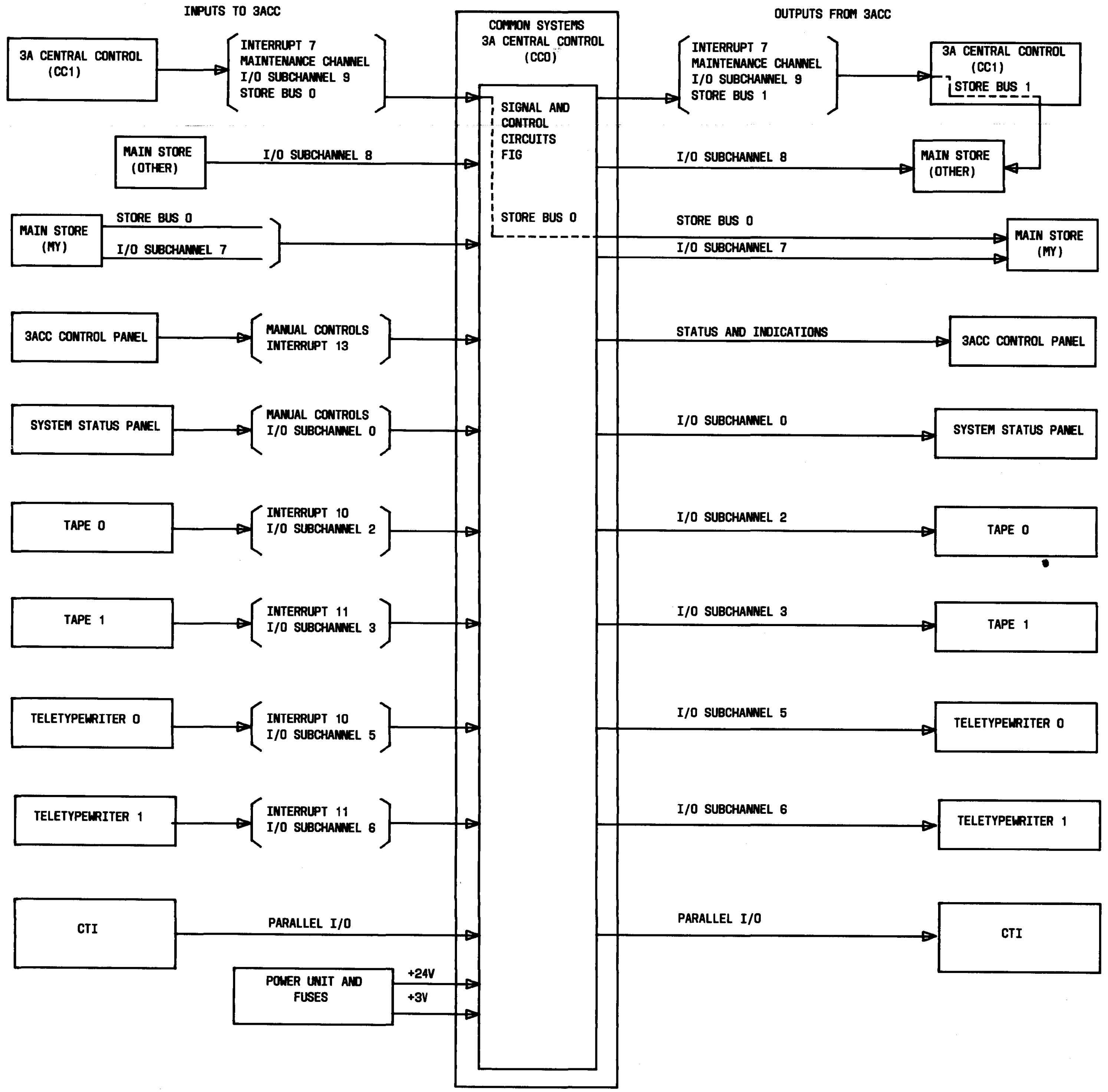


Fig. 4—Typical 3A CC Interface Diagram

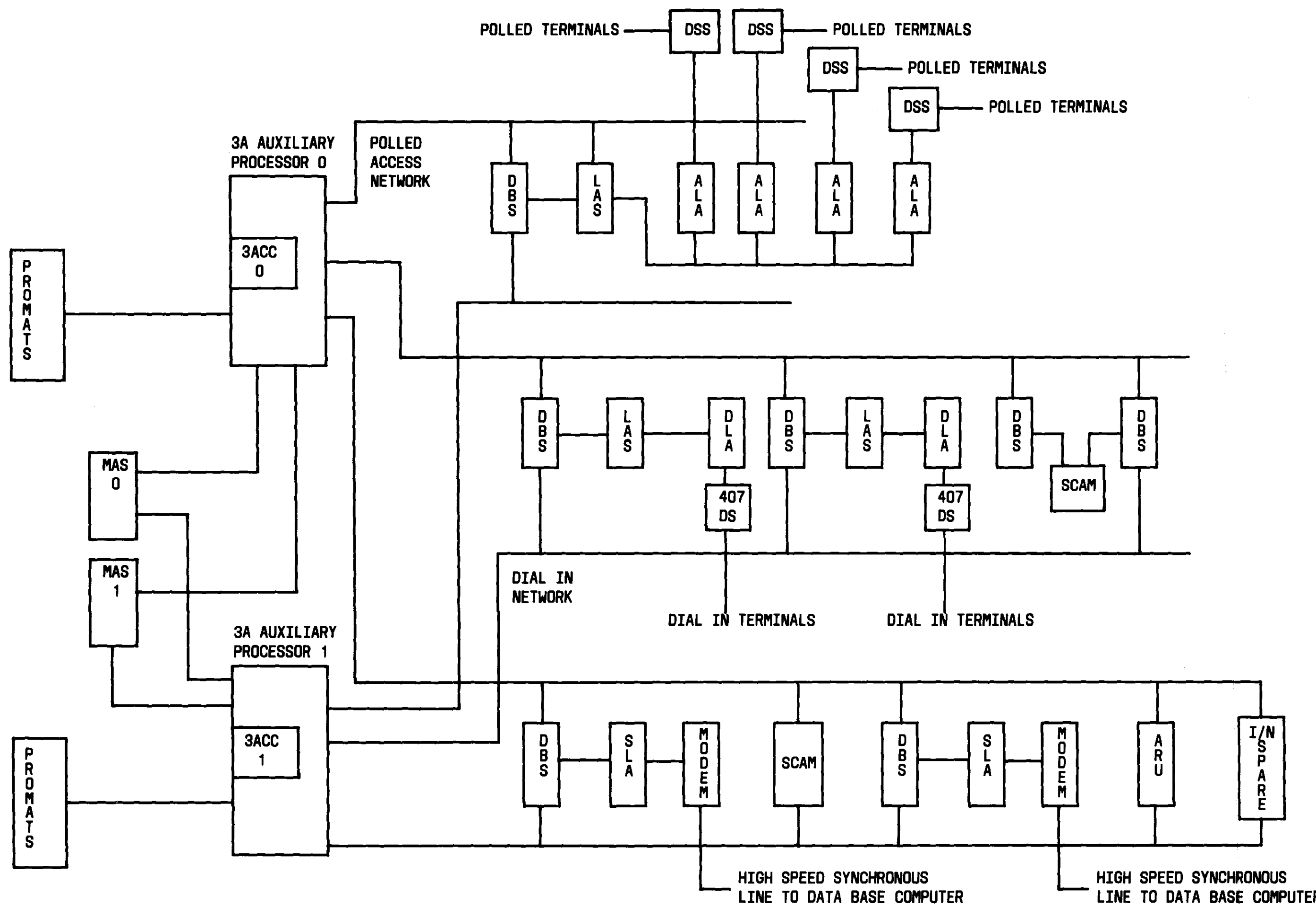


Fig. 5—Transaction Network—Block Diagram

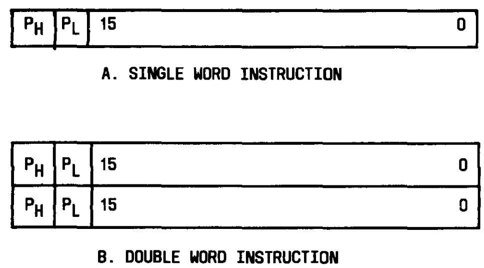


Fig. 6—Types of Instructions in Basic Set

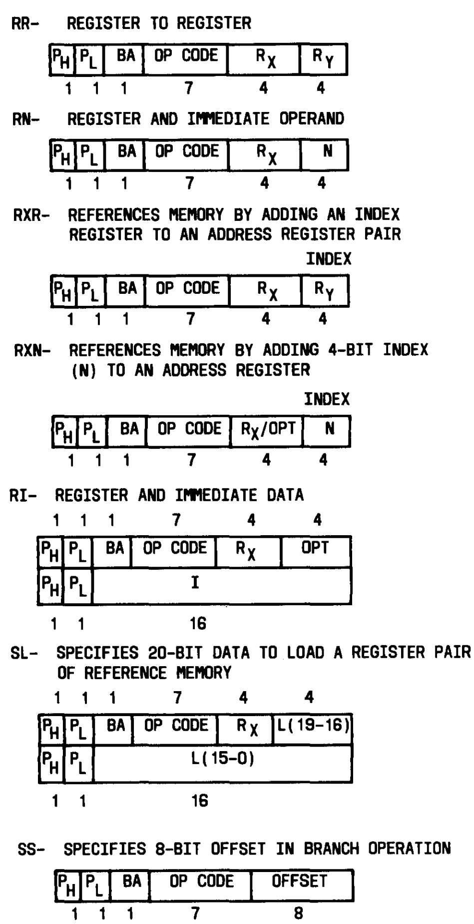


Fig. 7—General Format of the Instruction Set

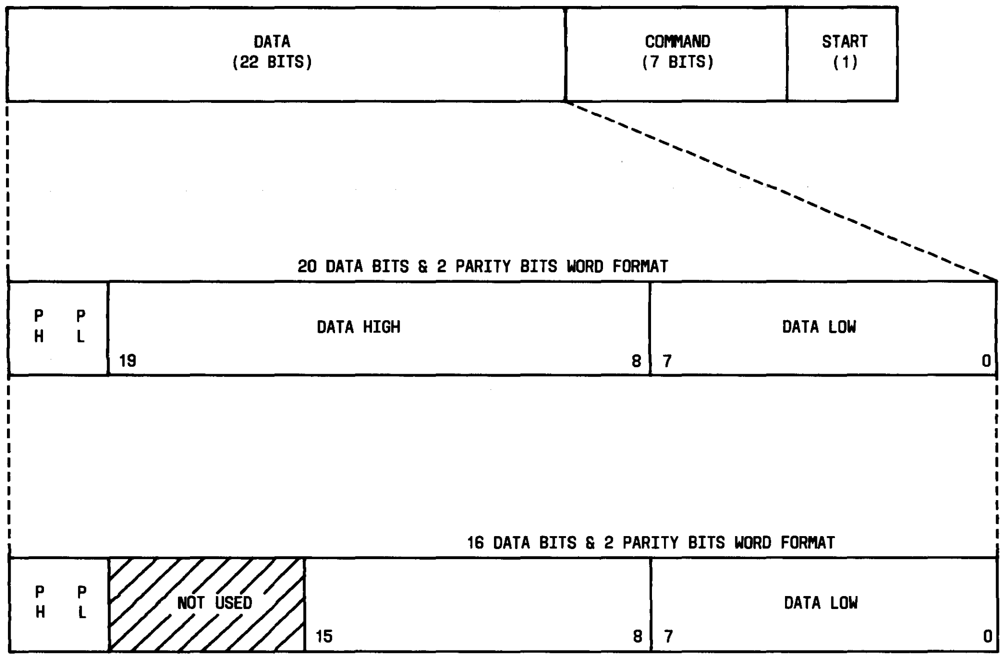
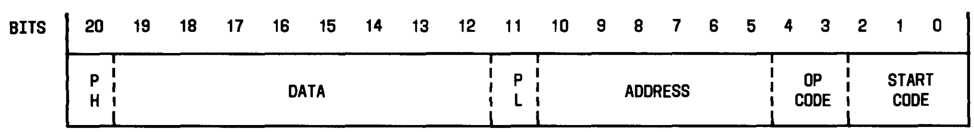


Fig. 8—Maintenance Channel Word Format



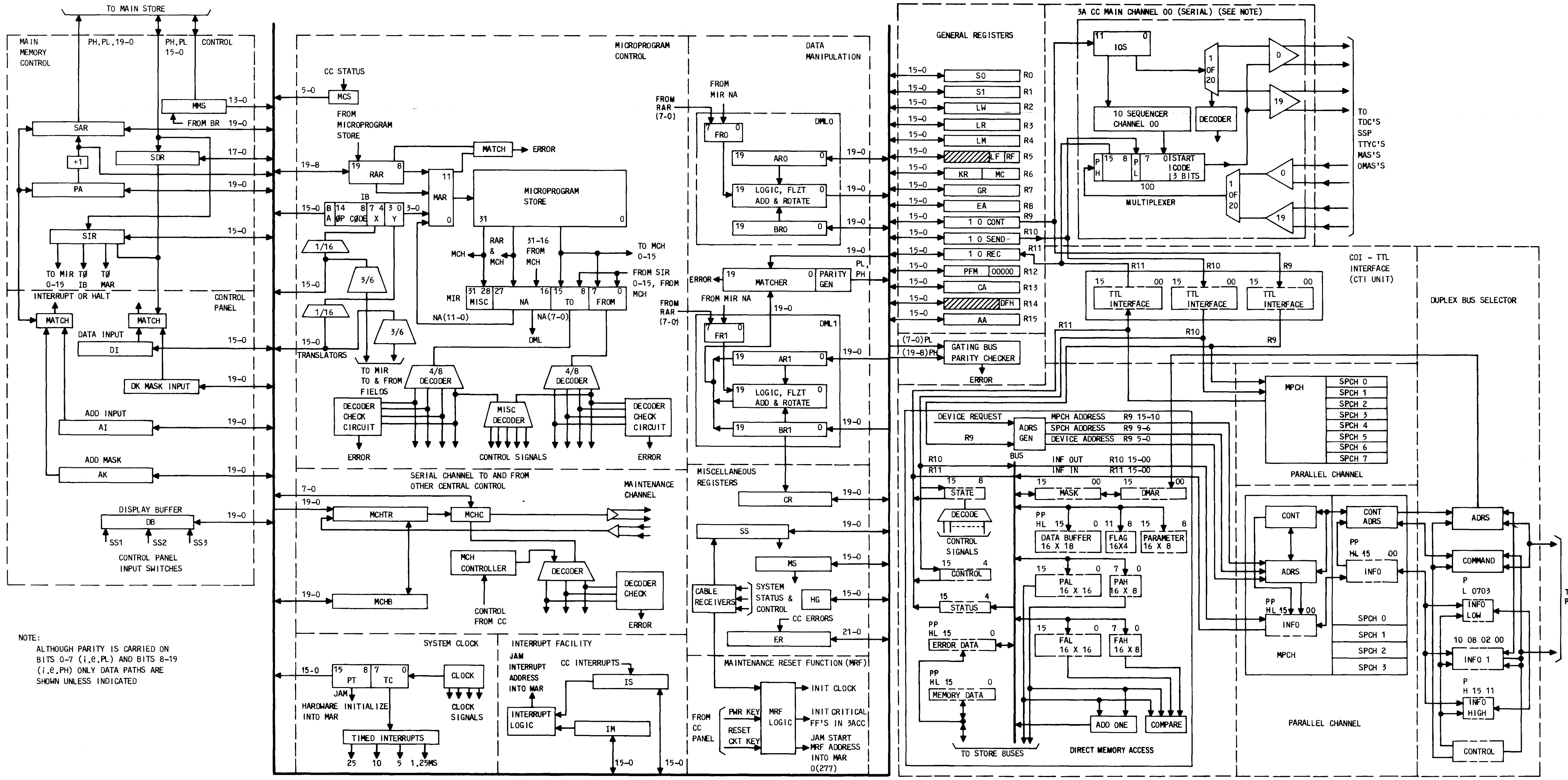
START CODE - ALWAYS 011

OP CODE - 01, WRITE-READ OPERATION (DATA WRITTEN INTO SSP FLIP-FLOP MEMORY AT ADDRESS SPECIFIED AND IMMEDIATELY READ BACK TO THE CC.
- 10, READ ONLY OPERATION.

ADDRESS - OF FLIP-FLOP MEMORY LOCATION IN SSP CONTROLLER

DATA - VALUE TO BE INSERTED OR OBTAINED FROM ADDRESS SELECTED LOCATION.

Fig. 9—System Status Panel Word Format



NOTE:
ALTHOUGH PARITY IS CARRIED ON BITS 0-7 (I.E.PL) AND BITS 8-19 (I.E.PH) ONLY DATA PATHS ARE SHOWN UNLESS INDICATED

Fig. 10—3A CC Functional Block Diagram

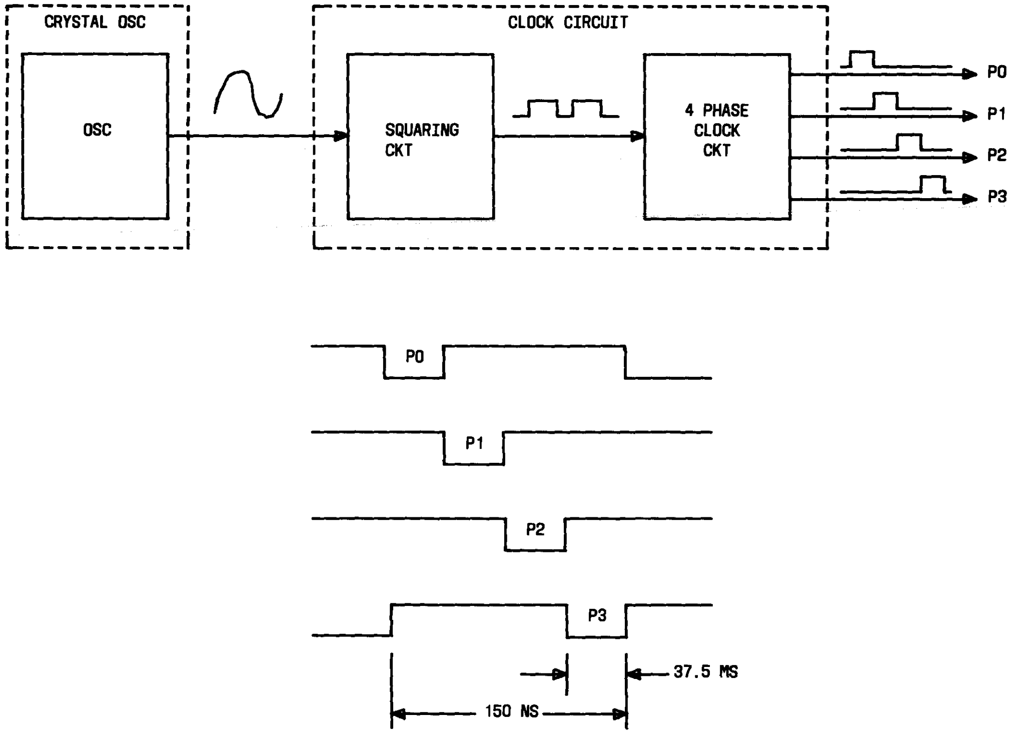
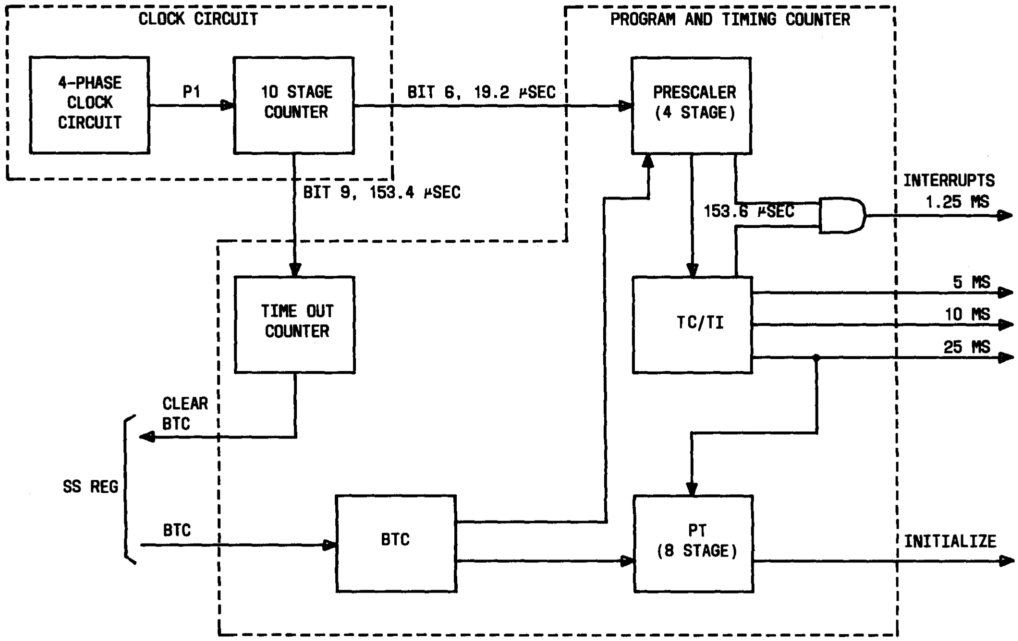


Fig. 11—Clock and Pulse Structure



BTC - BLOCK TIMER CHECK
TC - TIMING COUNTER
PT - PROGRAM TIMER
TI - TIMING REGISTER

Fig. 12—Prescaler, Timing Counter, Program Timer

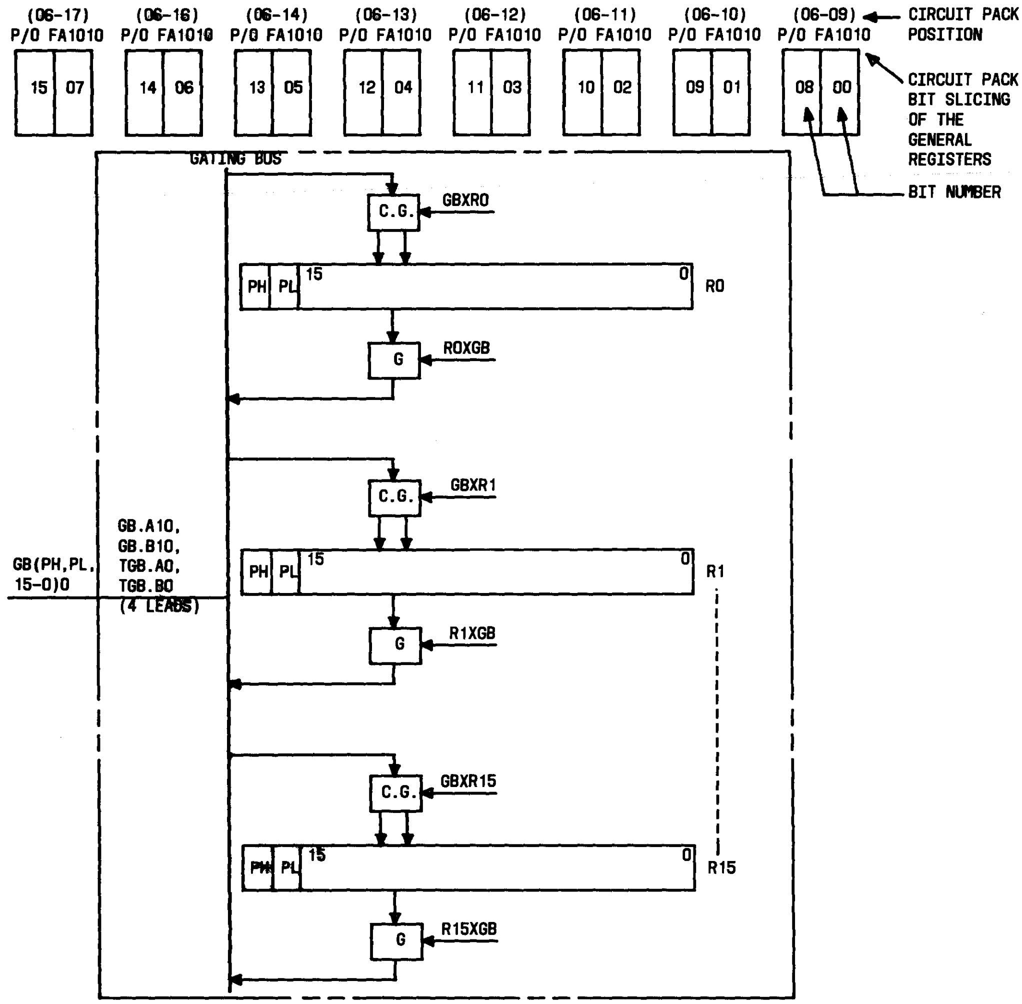


Fig. 13—General Registers R0 through R15

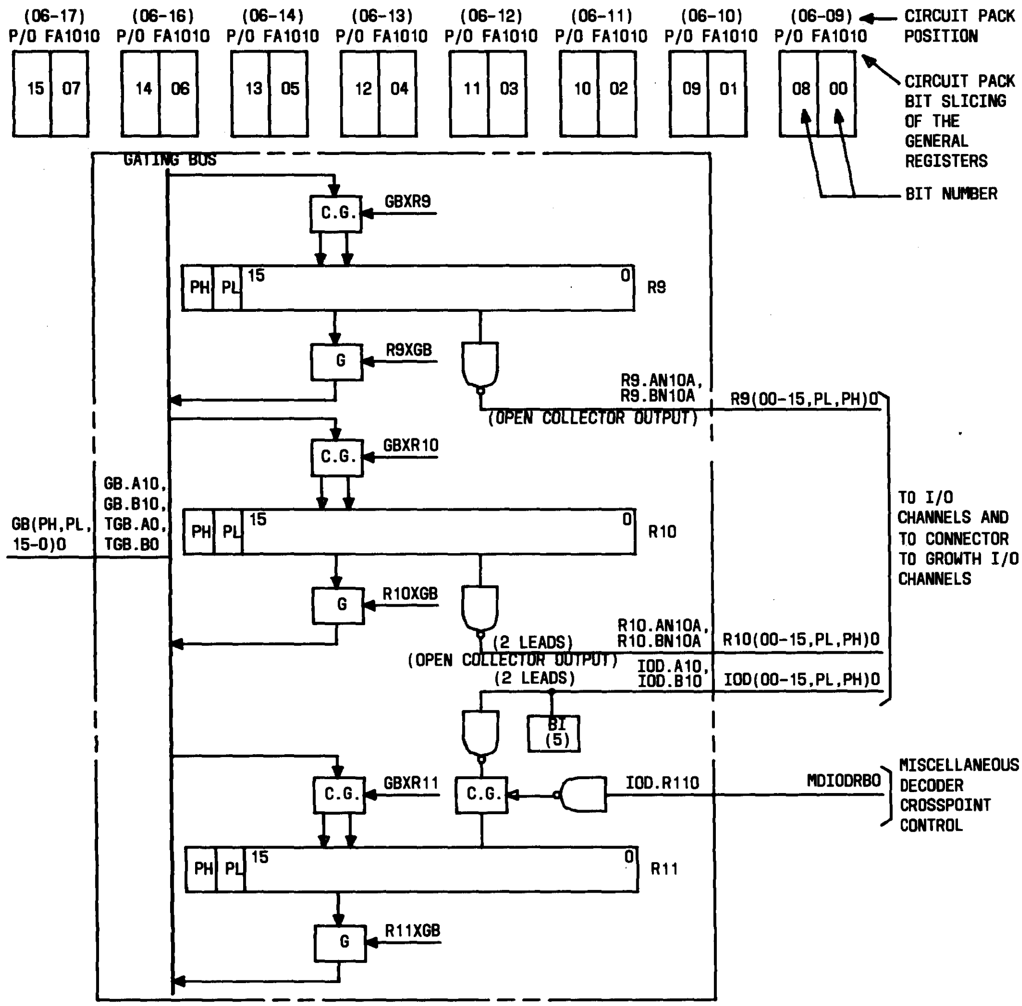


Fig. 14—General Registers R9, R10, R11

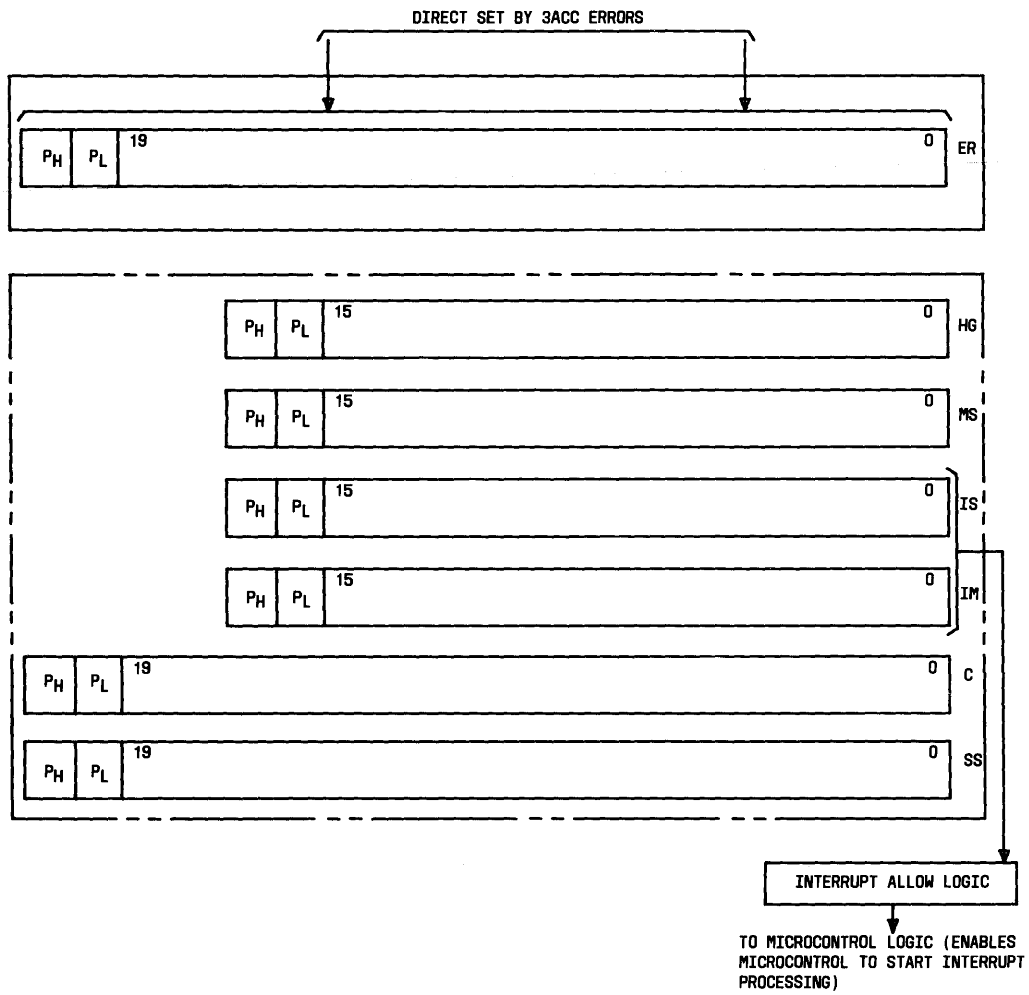


Fig. 15—Special Registers

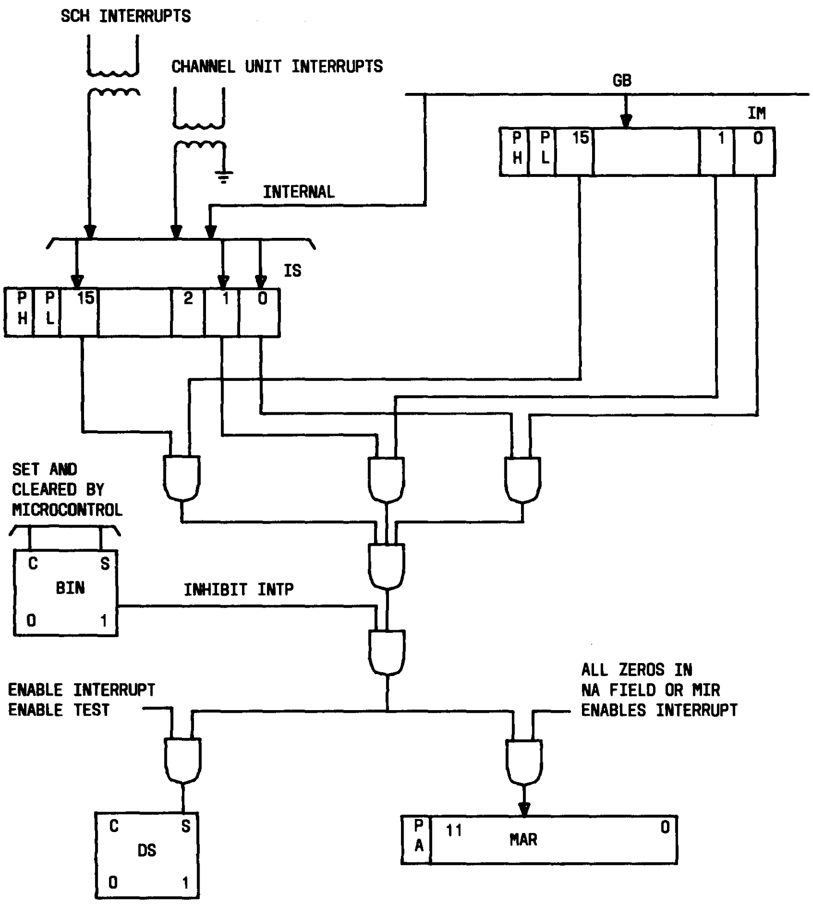


Fig. 16—Interrupt Hardware—Functional Block Diagram

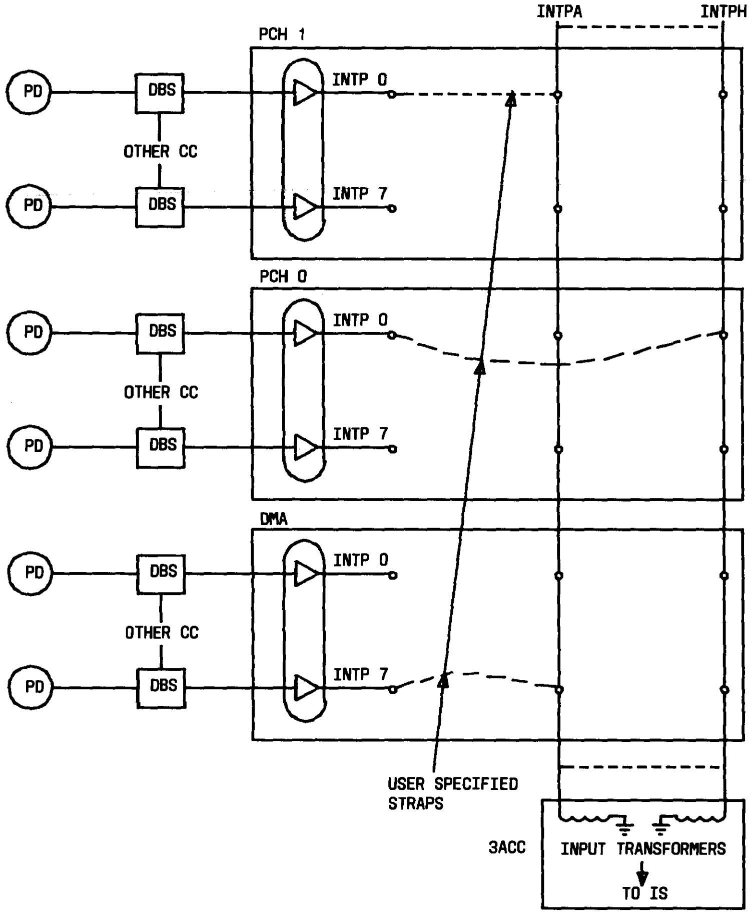


Fig. 17—3A CC/Channel Unit Interface

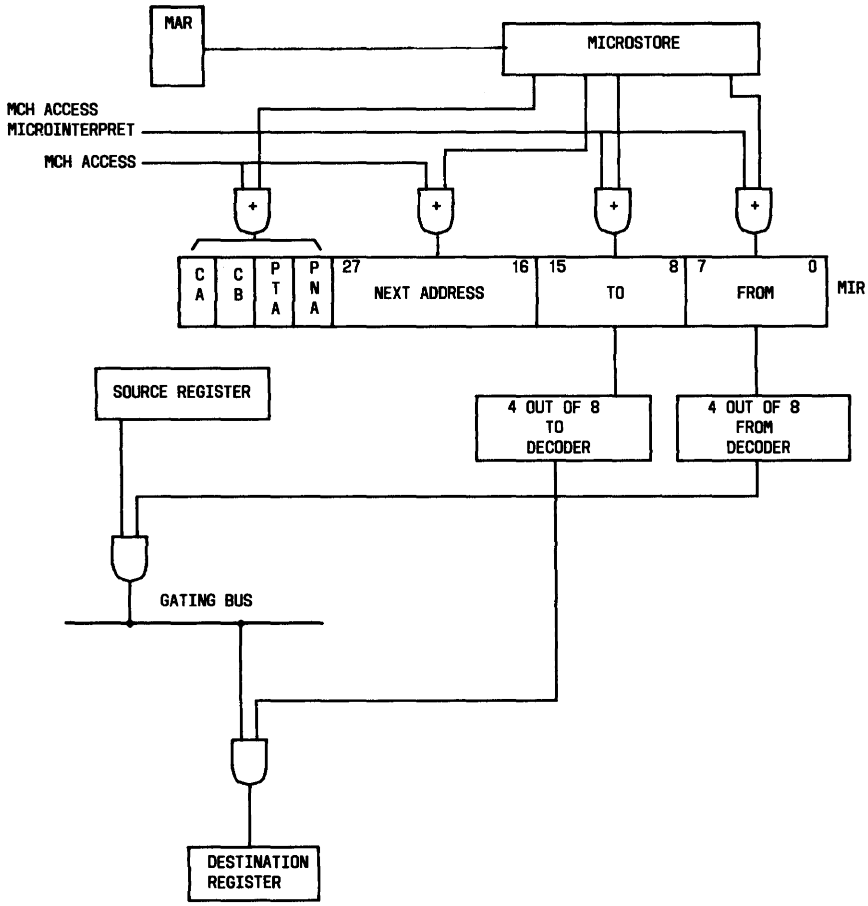


Fig. 18—Microcontrol—Simplified Logic Diagram

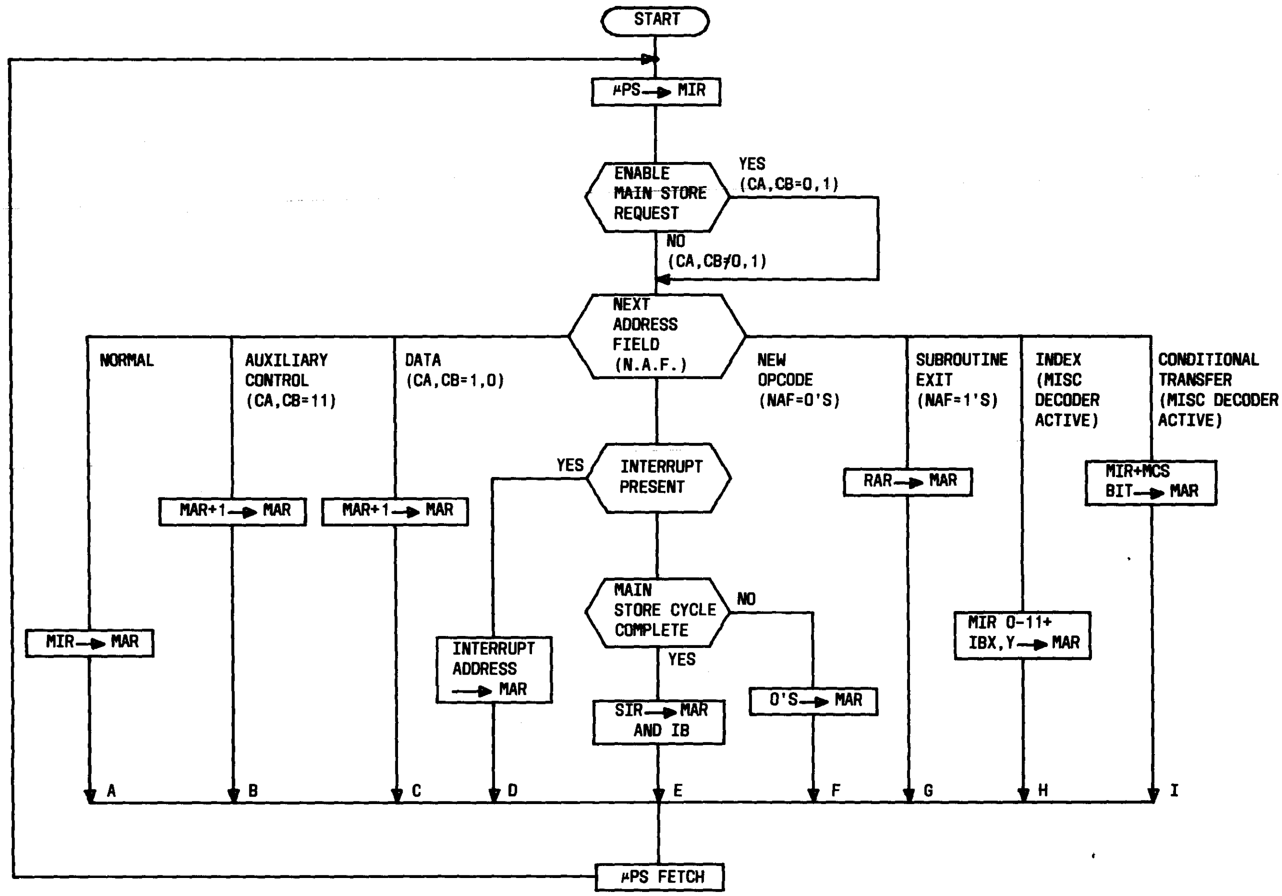


Fig. 19—Microcontrol Addressing Flowchart

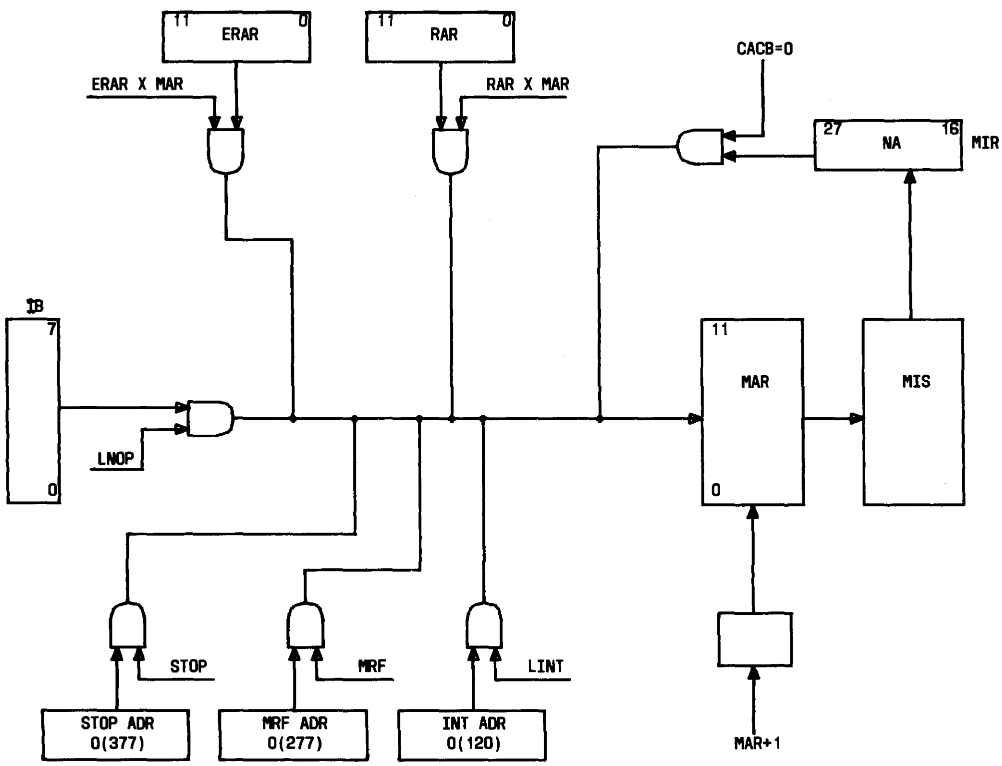


Fig. 20—Address Gating Into MAR—Simplified Logic Diagram

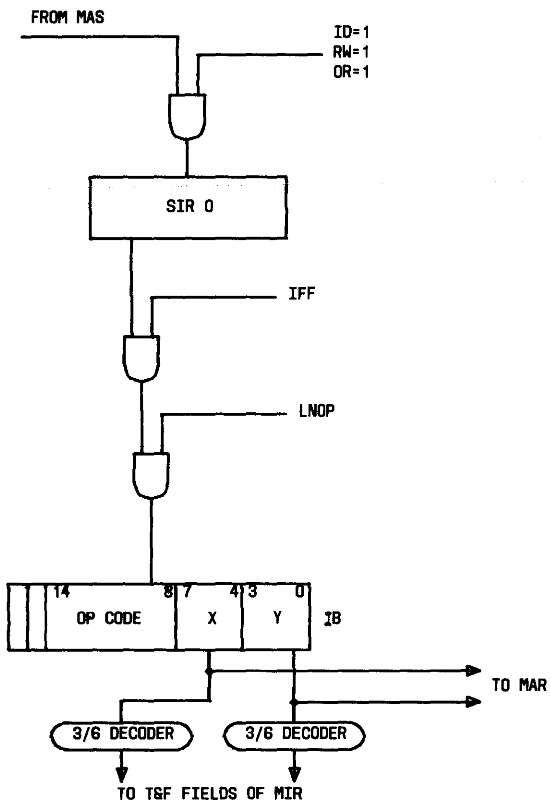


Fig. 21—Instruction to IB Gating—Simplified Logic Diagram

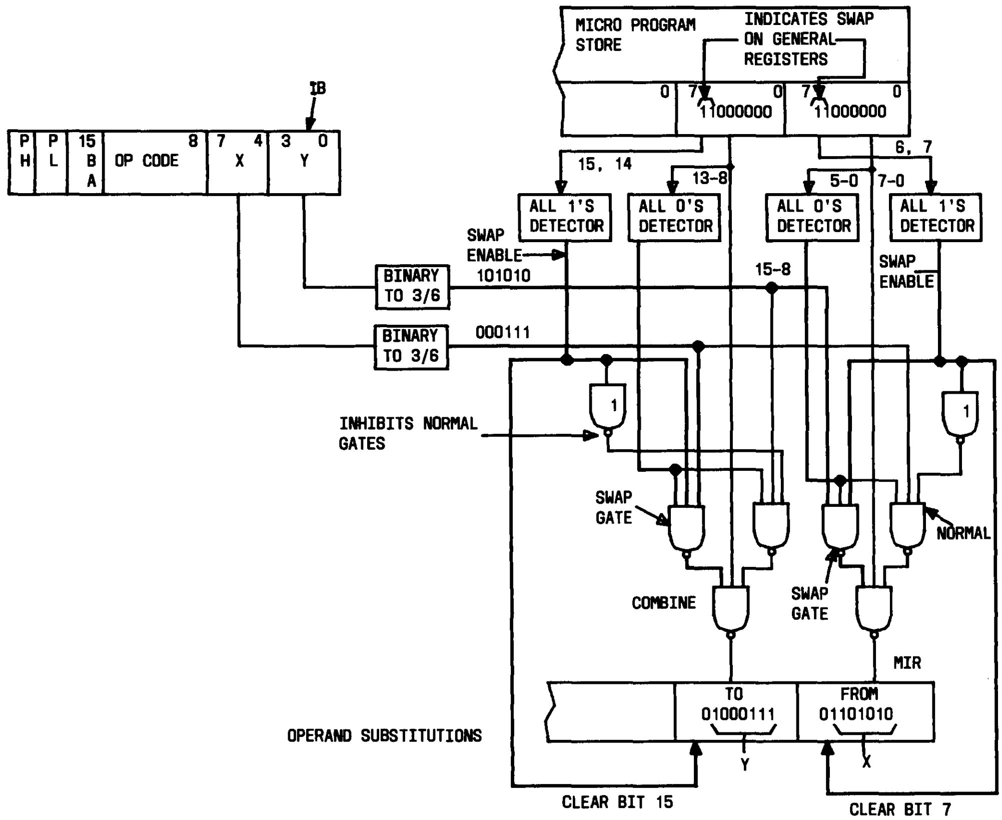


Fig. 22—Operand Substitutions

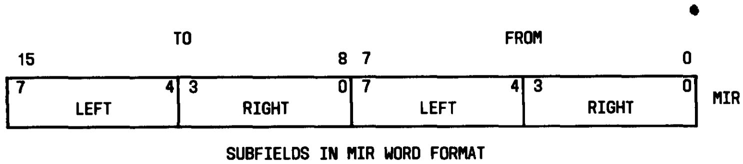


Fig. 23—Decoder Fields

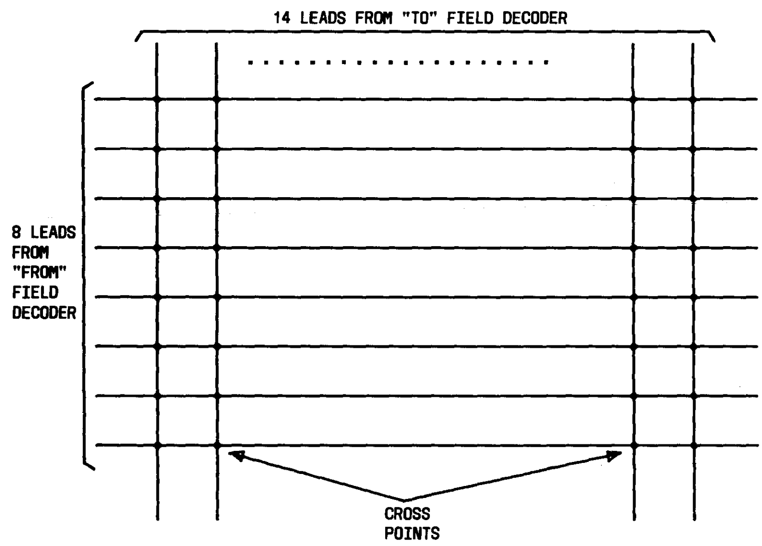


Fig. 24—Symbolic Representation of Miscellaneous Decoder Matrix

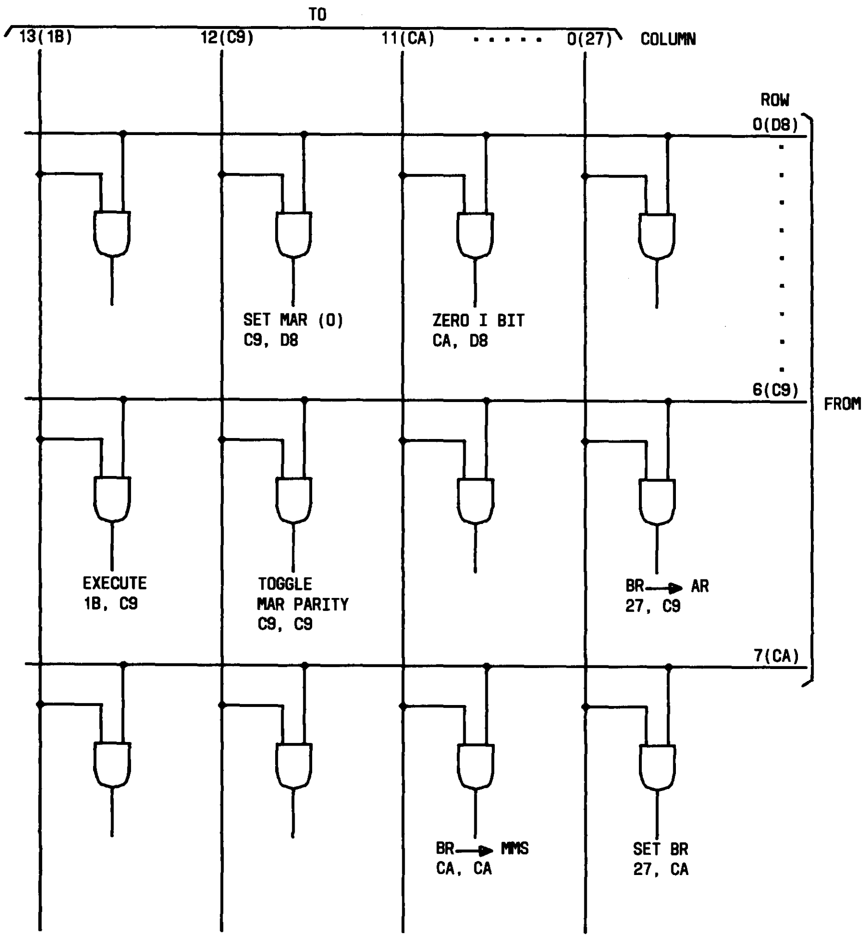


Fig. 25—Miscellaneous Decoder—Simplified Logic Diagram

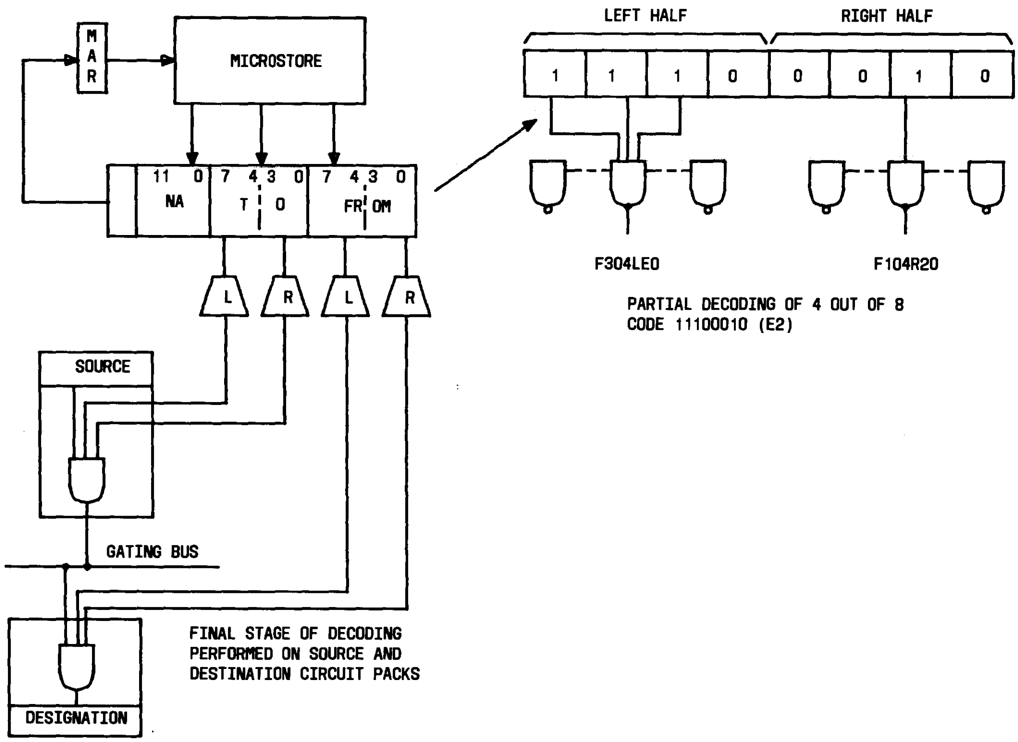


Fig. 26—Mnemonic Definition of TO-FROM Decoding

tpa610

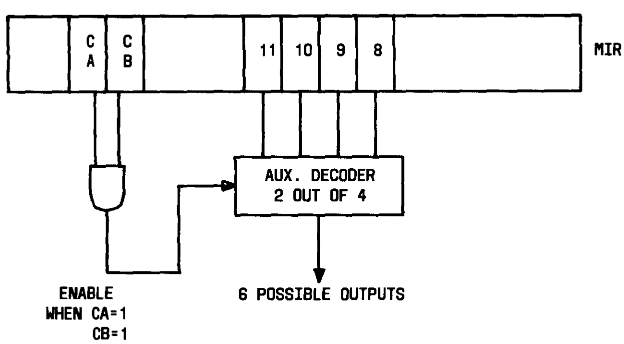


Fig. 27—Auxiliary Decoder—Simplified Logic Diagram

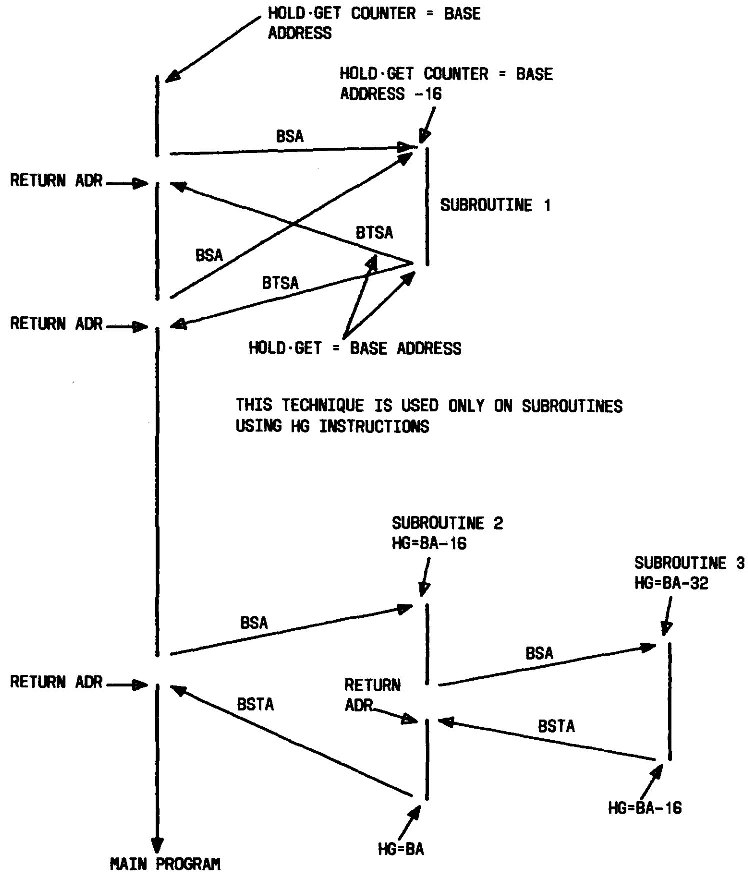


Fig. 28—Subroutine Program Flow

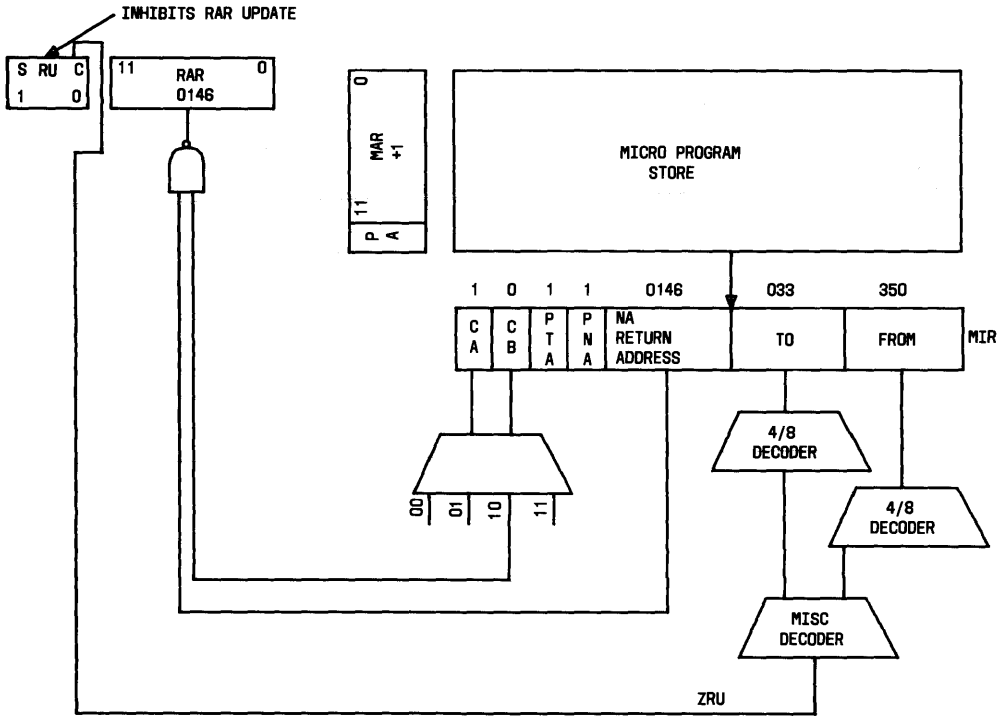


Fig. 29—Subroutine Entry

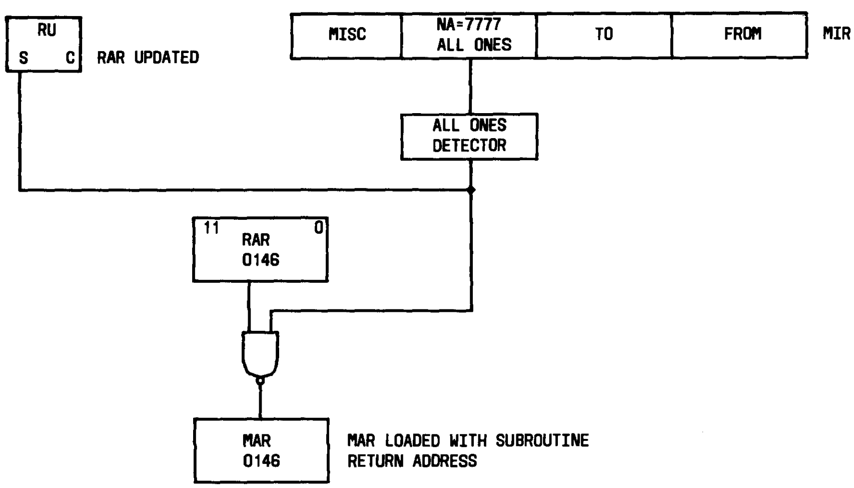


Fig. 30—Subroutine Exit

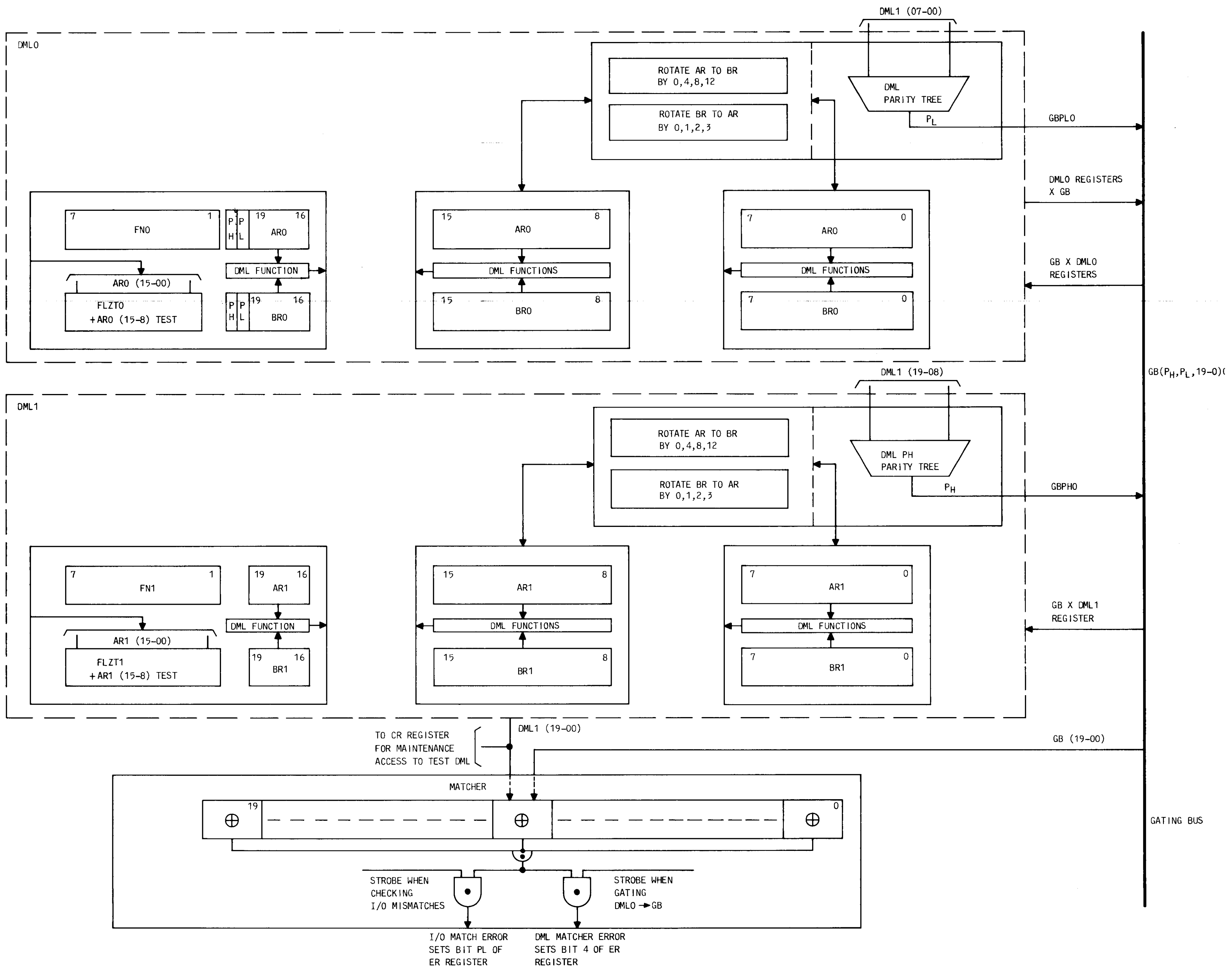


Fig. 31—DML0 and DML1—Functional Block Diagram

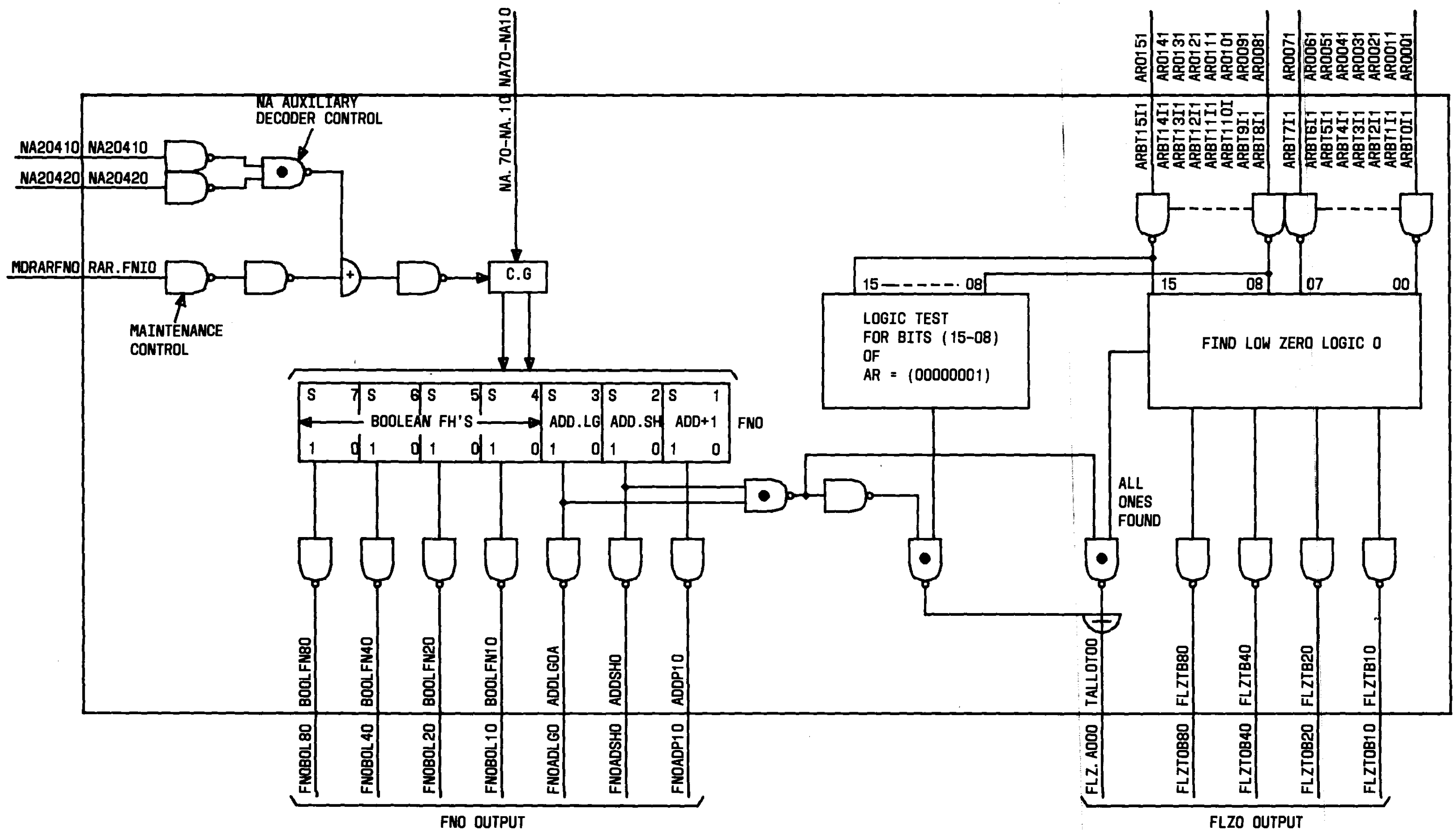


Fig. 32—DML, Function Register, and FLZT—Functional Block Diagram

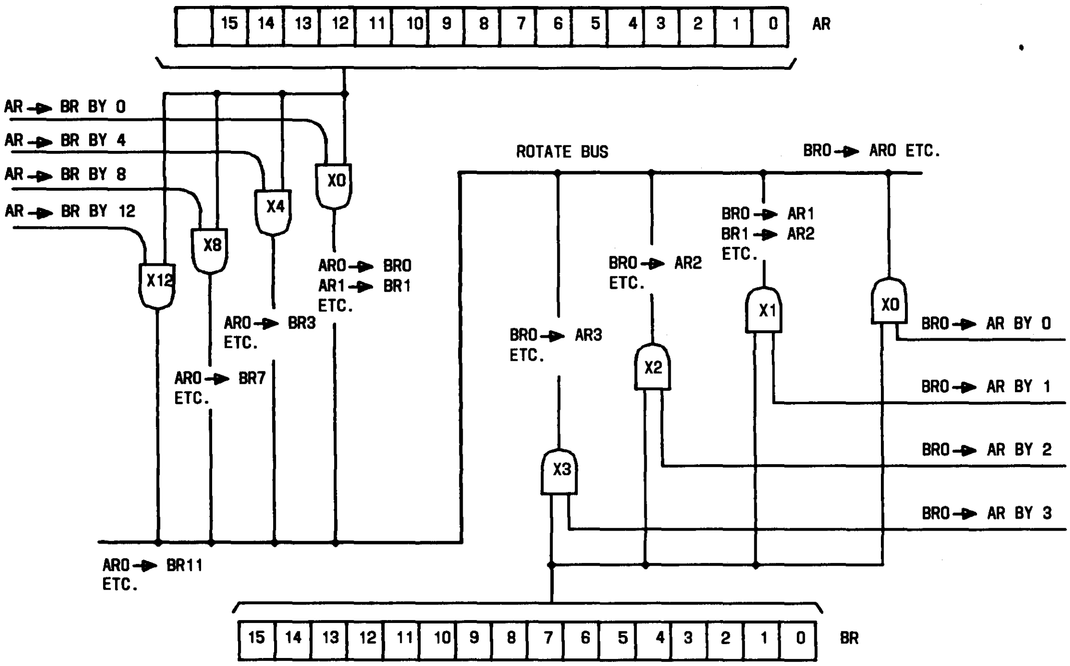


Fig. 33—Rotate Gating—Simplified Logic Diagram

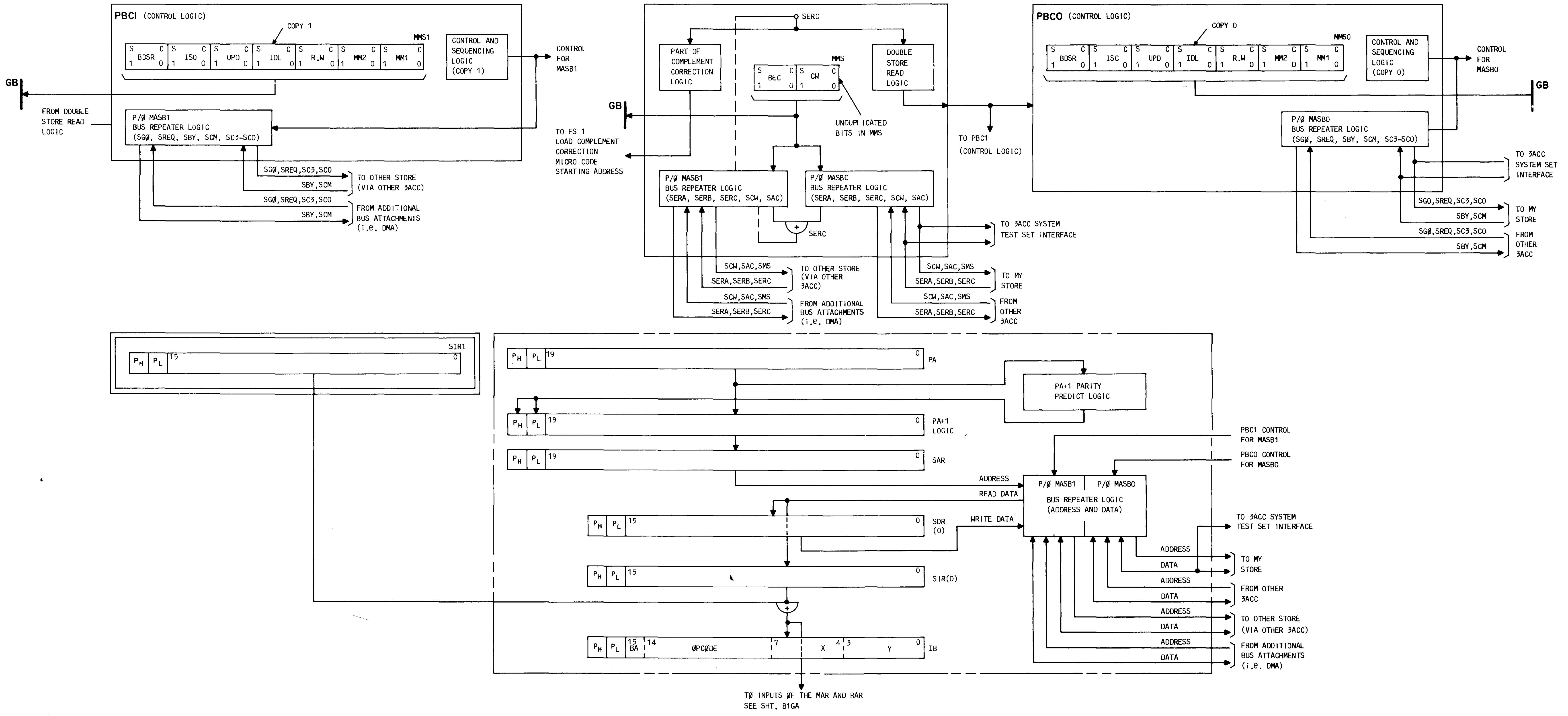


Fig. 34—Main Memory Interface Logic

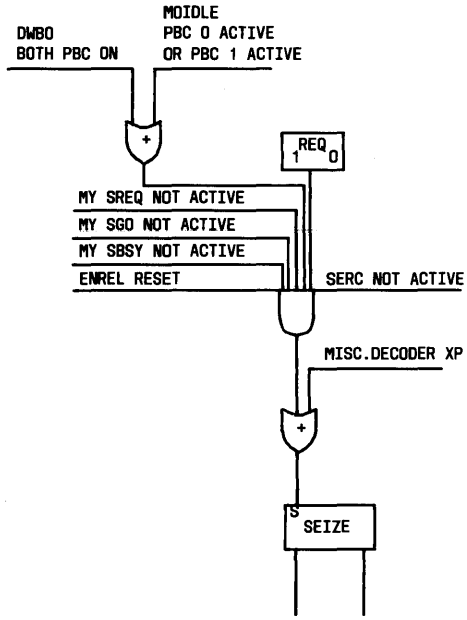


Fig. 35—SEIZE Flip-Flop—Simplified Logic Diagram

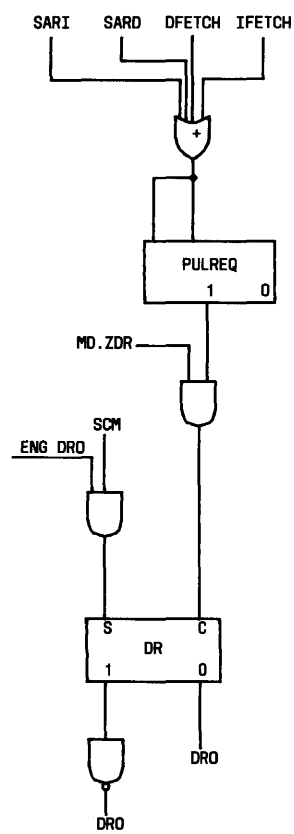
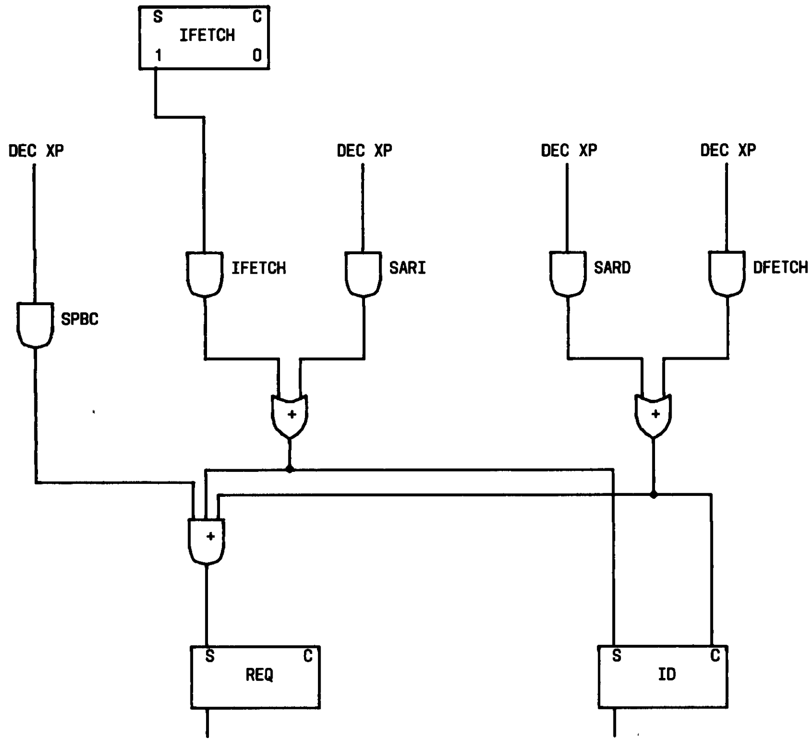


Fig. 36—DR Flip-Flop—Simplified Logic Diagram



FUNCTION	DECODER CROSSPOINTS (DEC XP)			
SAR1	T204LC0B	T204R50C	-	
SARD	T204LC0B	T204R30C	-	
DFETCH	T204L30B	T204RC0C	F204LC0B	F204RA0B
IFETCH	-		-	
SPBC	T104L10B	T304RB0B	F204LC0B	F204RA0B

Fig. 37—REQUEST and ID Flip-Flops—Simplified Logic Diagram

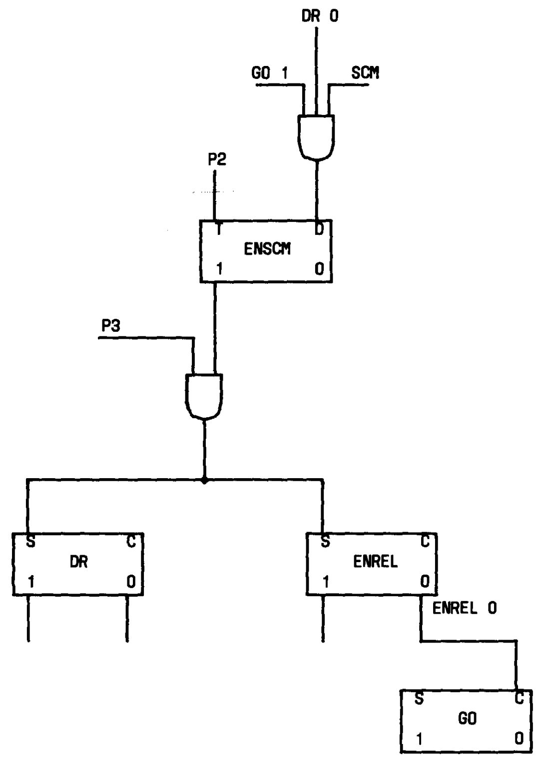


Fig. 38—PBC Control Flip-Flops—Simplified Logic Diagram

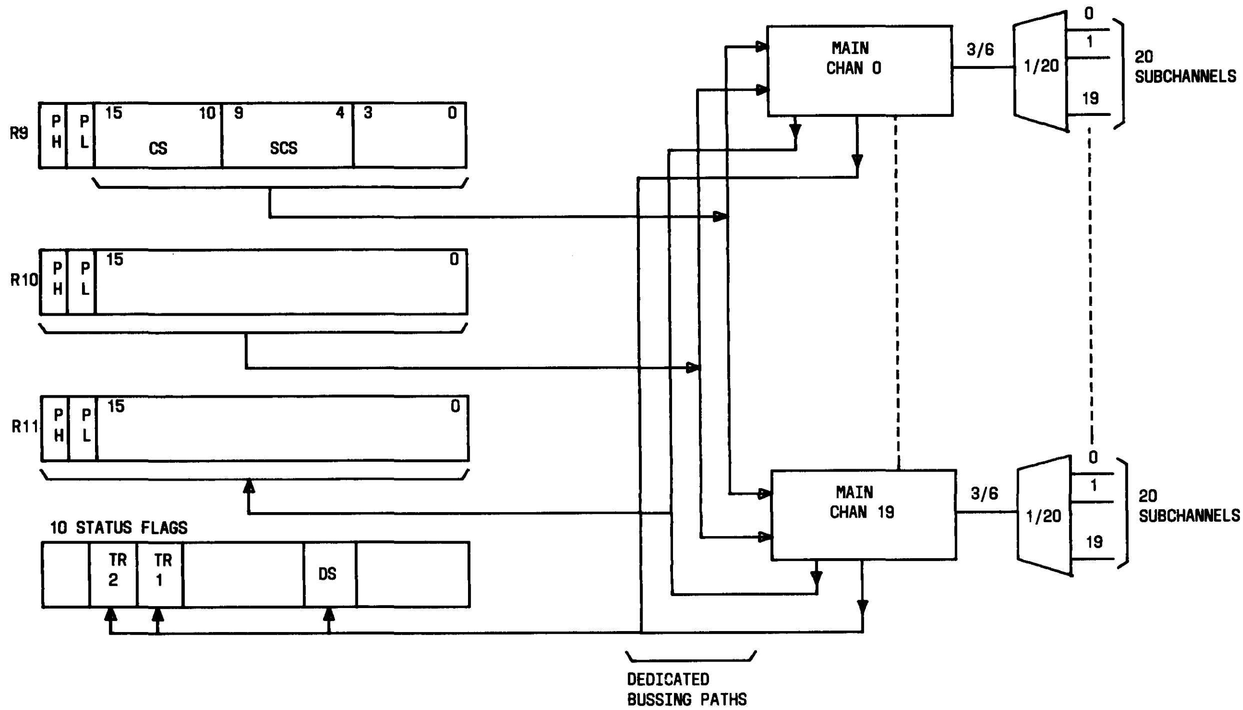


Fig. 39—3A CC/IO Serial Interface

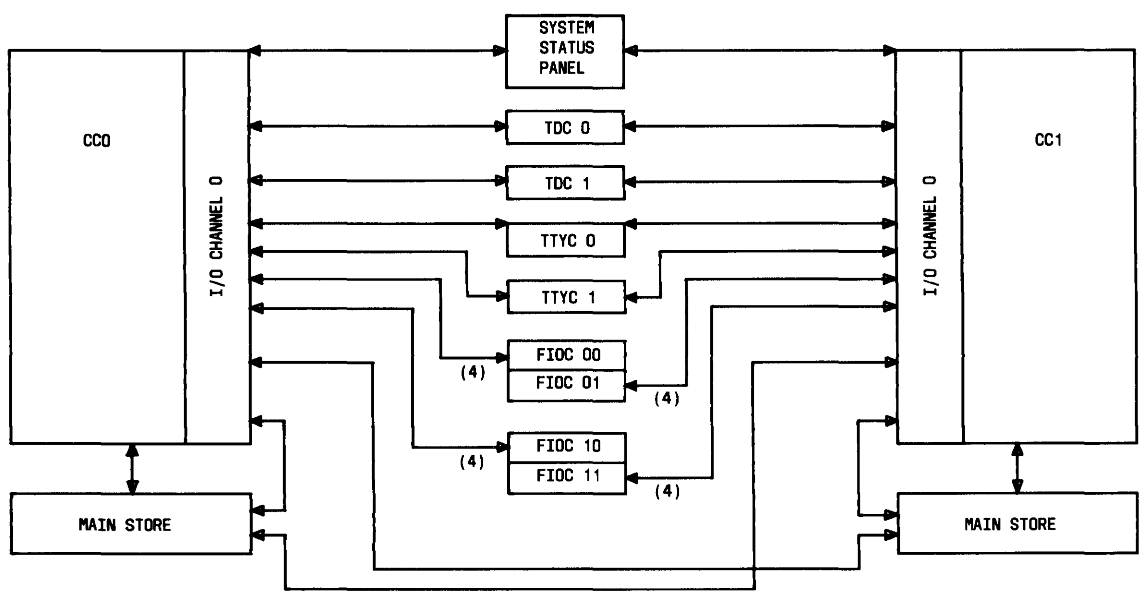


Fig. 40—I/O Subchannel Interconnectioons

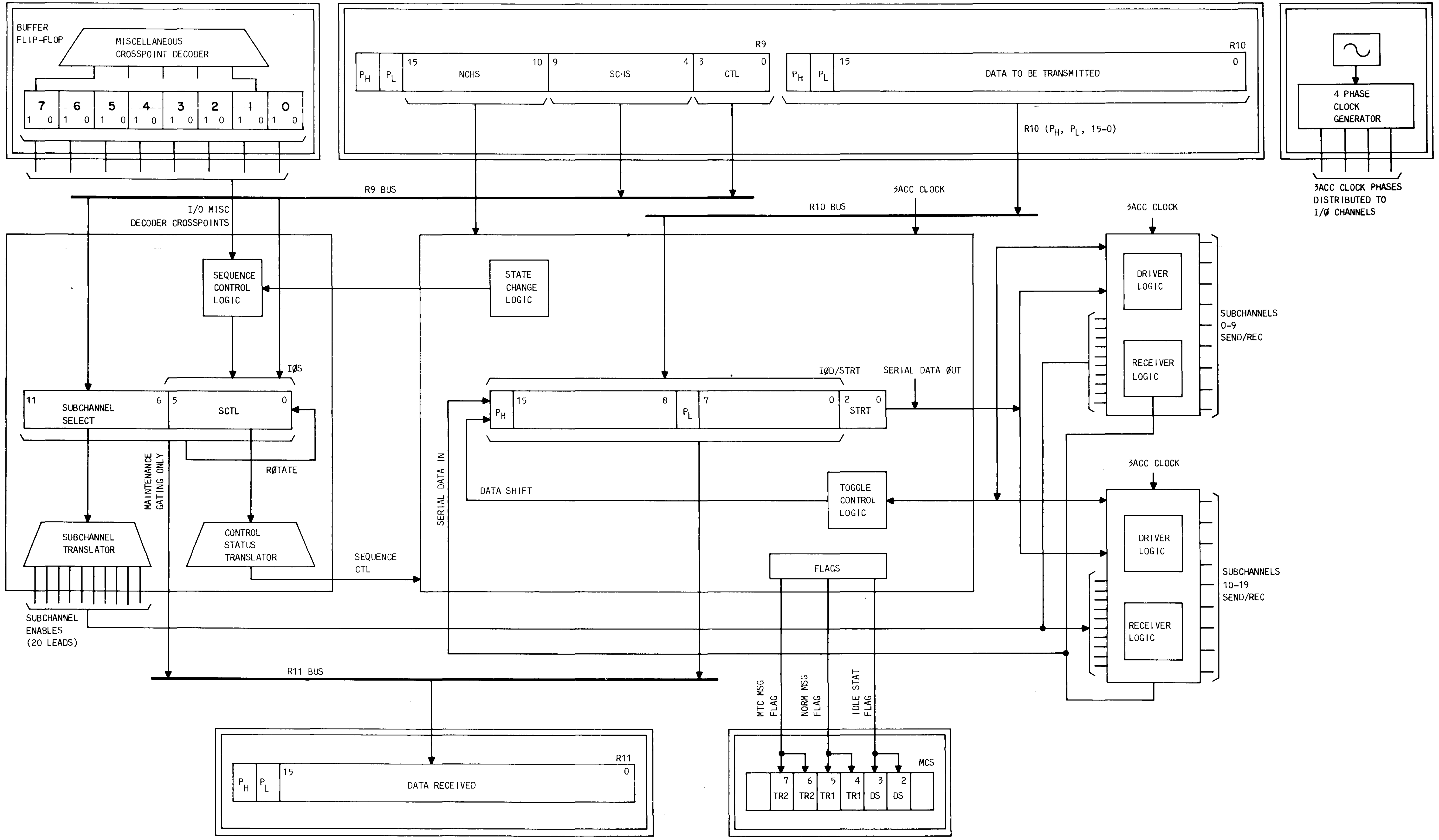


Fig. 41—Serial I/O Main Channel—Functional Block Diagram

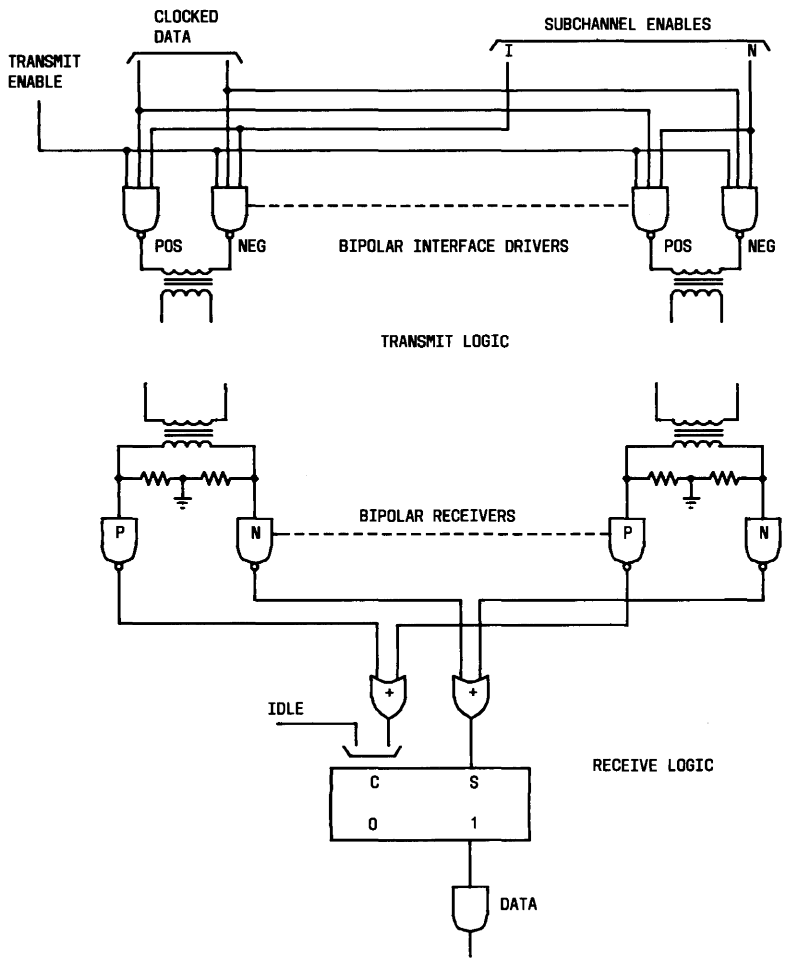
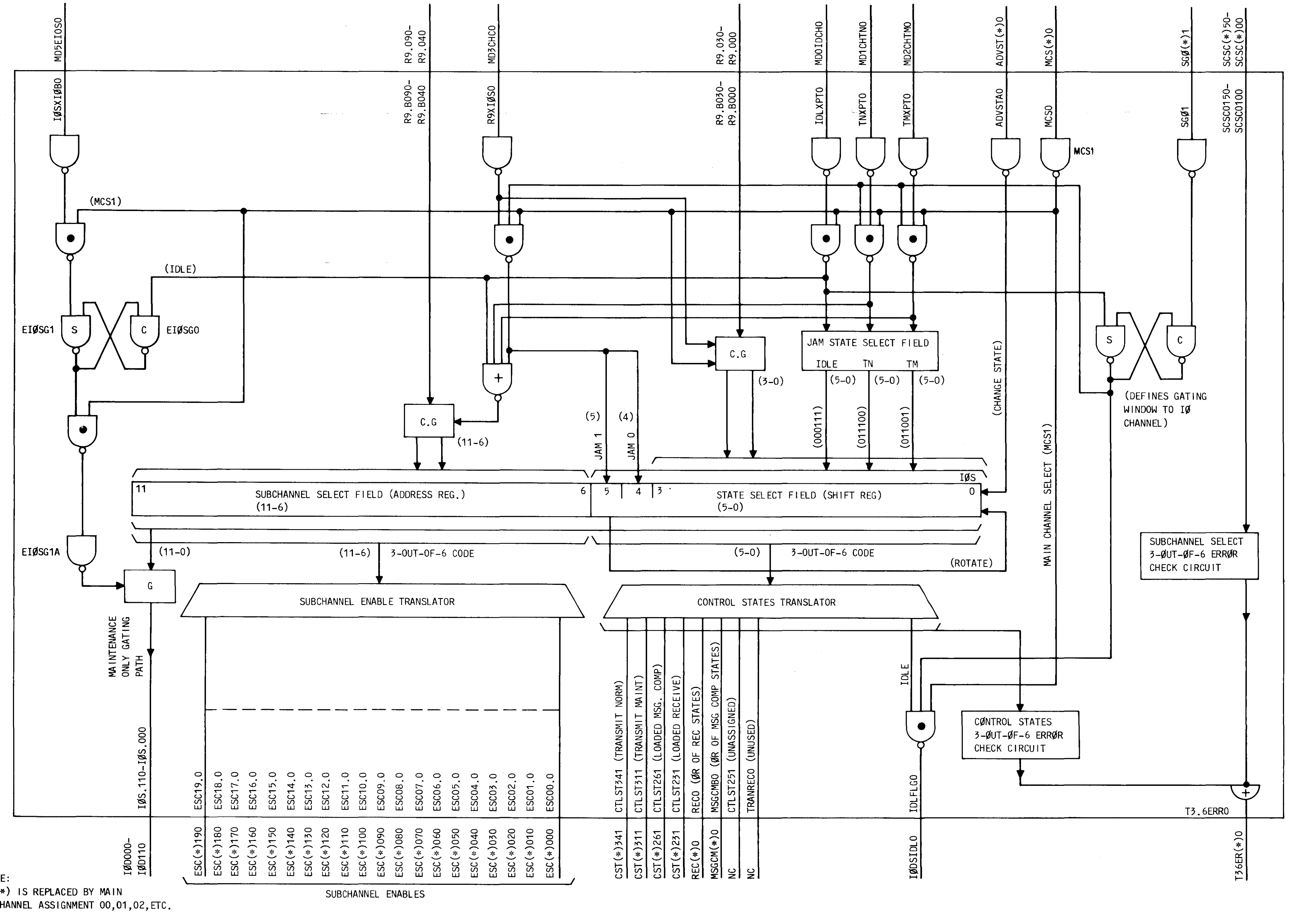
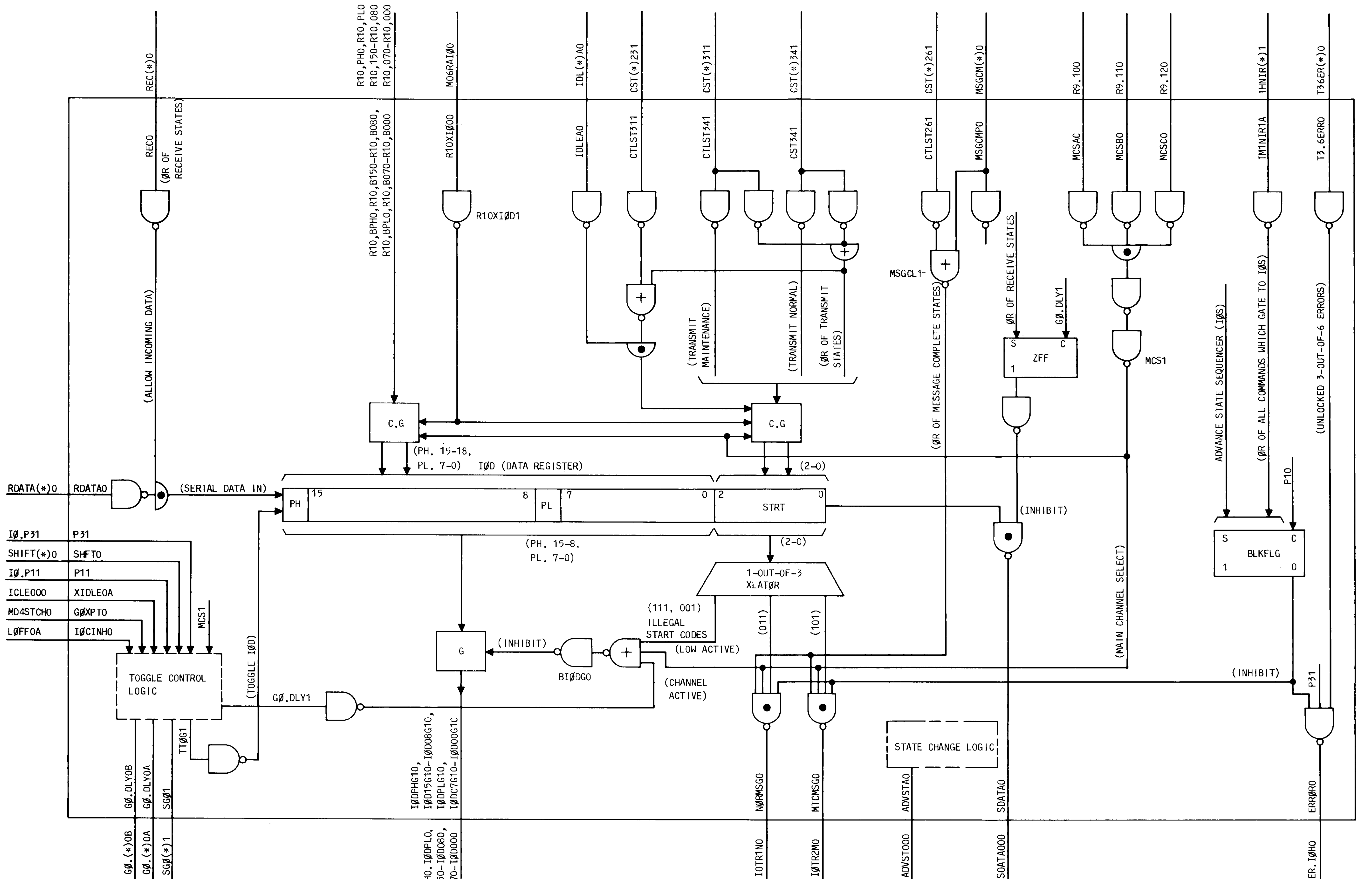


Fig. 42—Transmit/Receive Logic—Simplified Logic Diagram



NOTE:
(*) IS REPLACED BY MAIN CHANNEL ASSIGNMENT 00,01,02,ETC.

Fig. 43—I/O Status Register



NOTE:
(*) IS REPLACED BY MAIN CHANNEL ASSIGNMENT NUMBER 00,01,02, ETC.

Fig. 44—I/O Data Register

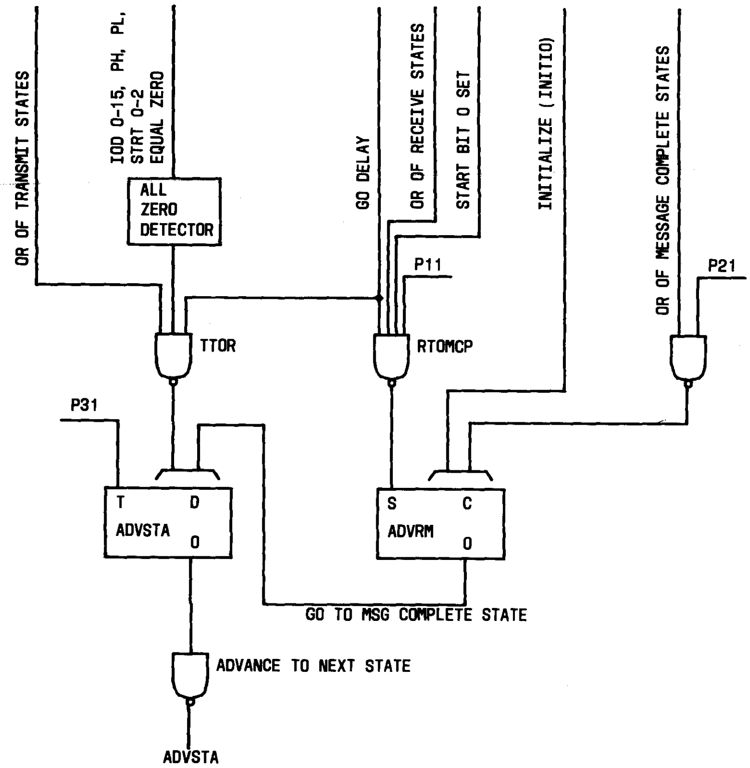


Fig. 45—I/O State Change Logic

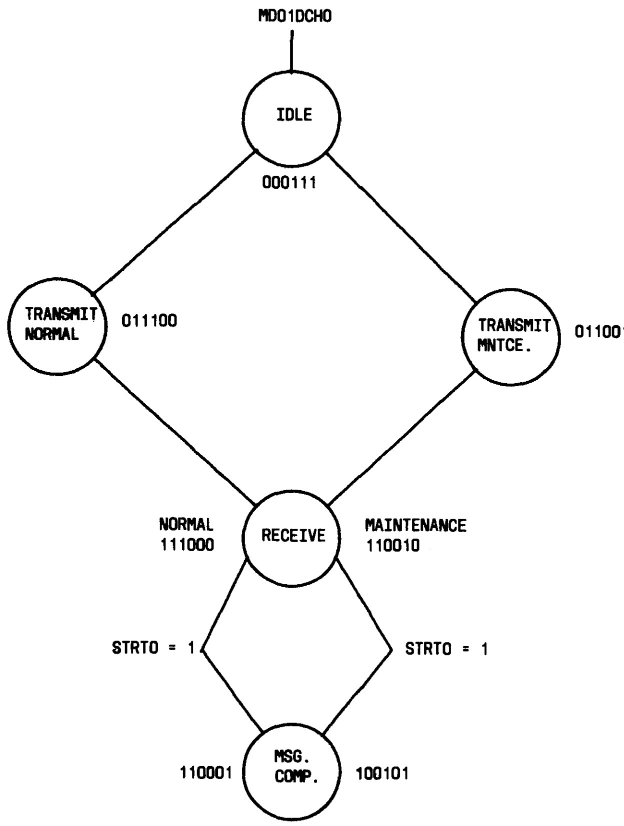


Fig. 46—Serial I/O Sequence State Diagram

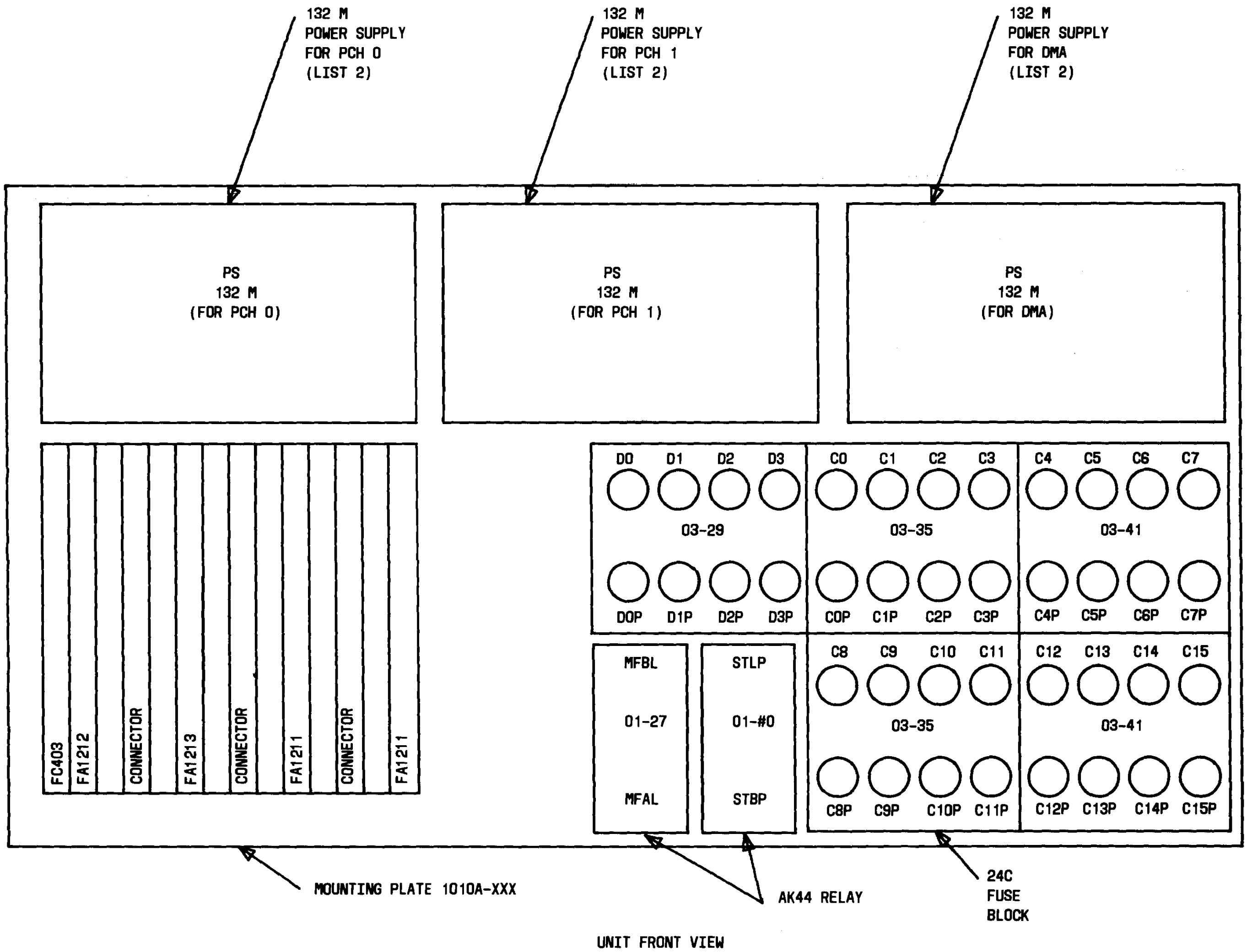


Fig. 47—CTI Unit—Front View

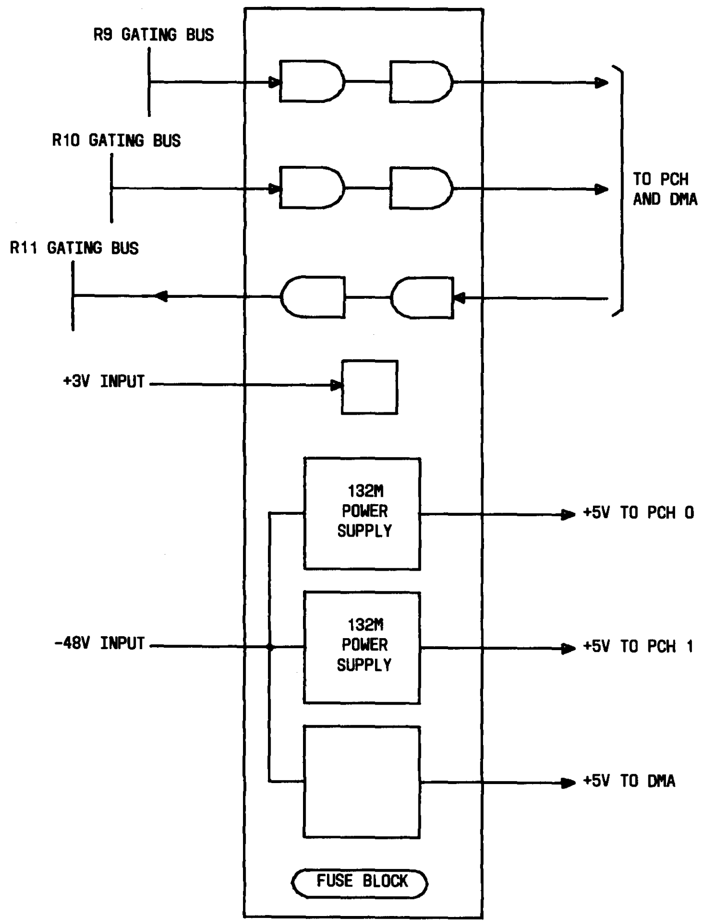
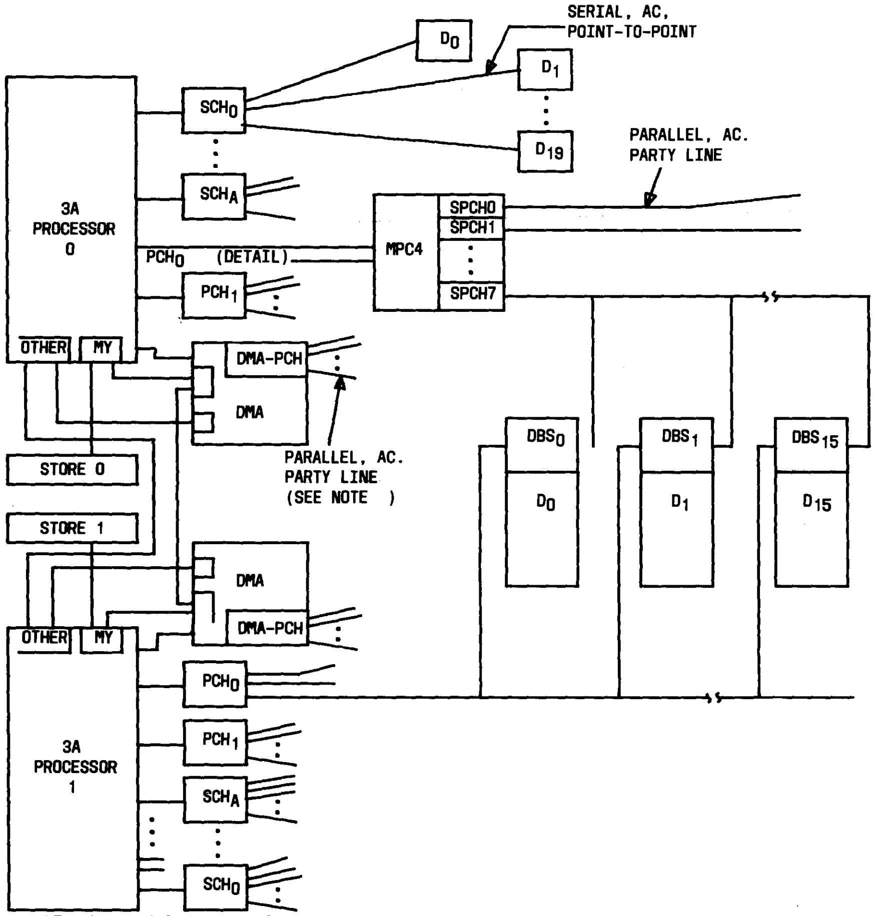
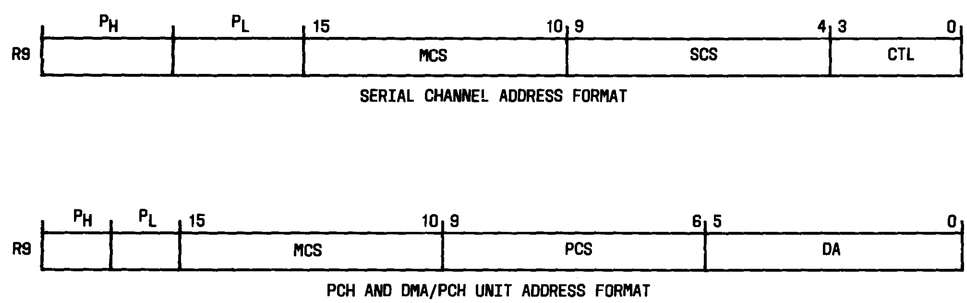


Fig. 48—CTI Unit—Block Diagram



NOTE: THE DMA/PCH CAN SUPPORT UP TO FOUR SPCH'S.

Fig. 49—I/O Structure



MCS - MAIN CHANNEL SELECT (3/6 CODE)
SCS - SERIAL CHANNEL SELECT (3/6 CODE)
CTL - OPTIONAL CONTROL FIELD
PCS - PARALLEL CHANNEL SELECT (3 BIT BINARY)
DA - DEVICE ADDRESS (6 BIT BINARY)

Fig. 50—Peripheral Address Formats

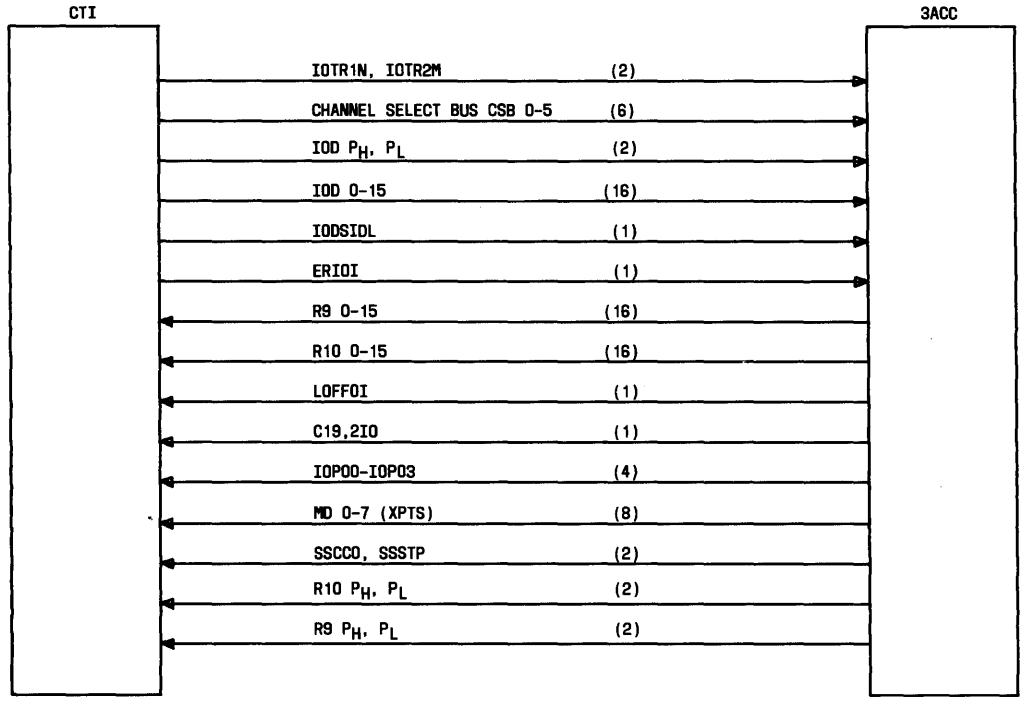


Fig. 51—CTI/3A CC Interface

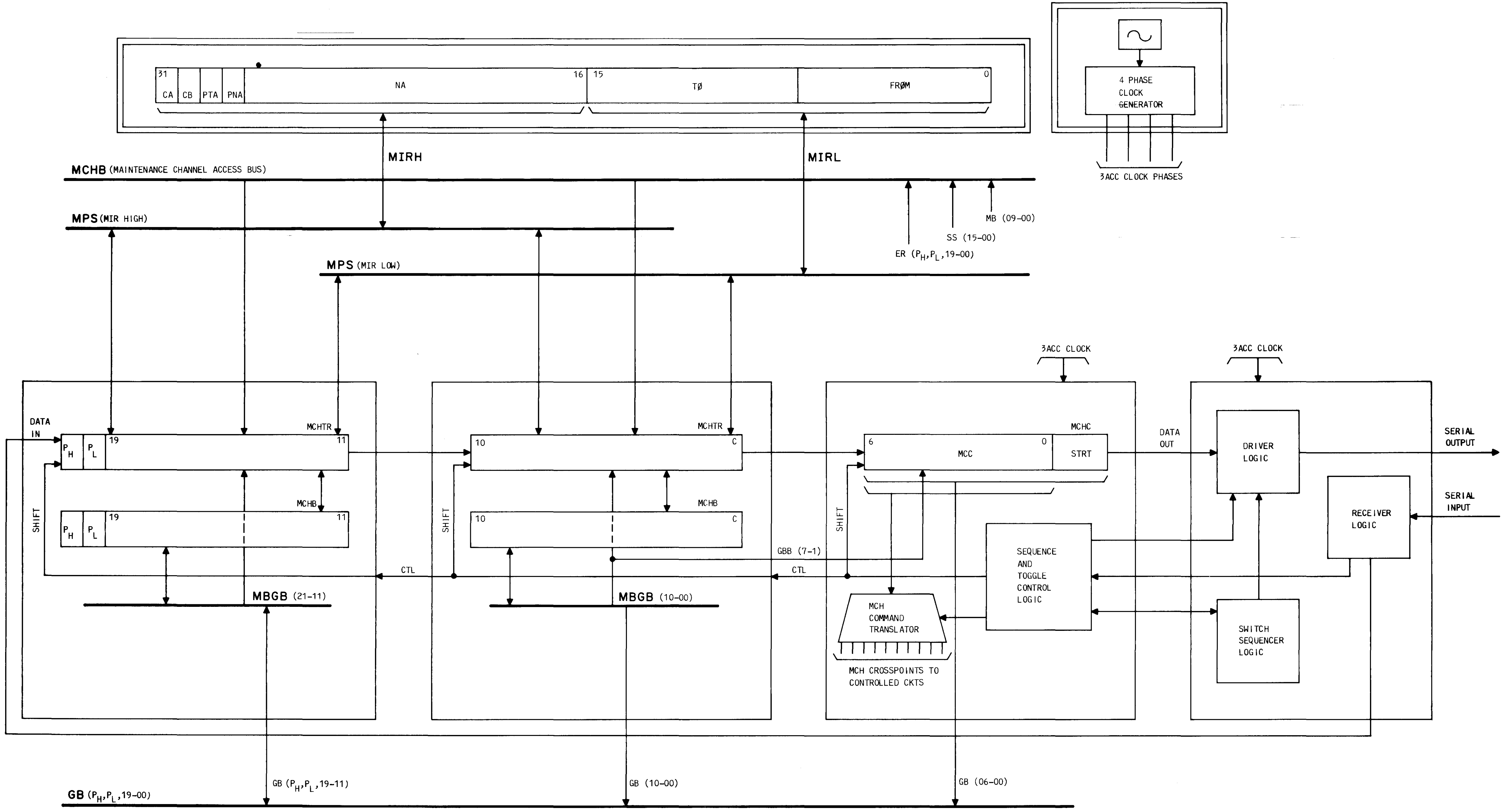


Fig. 52—Maintenance Channel—Functional Block Diagram

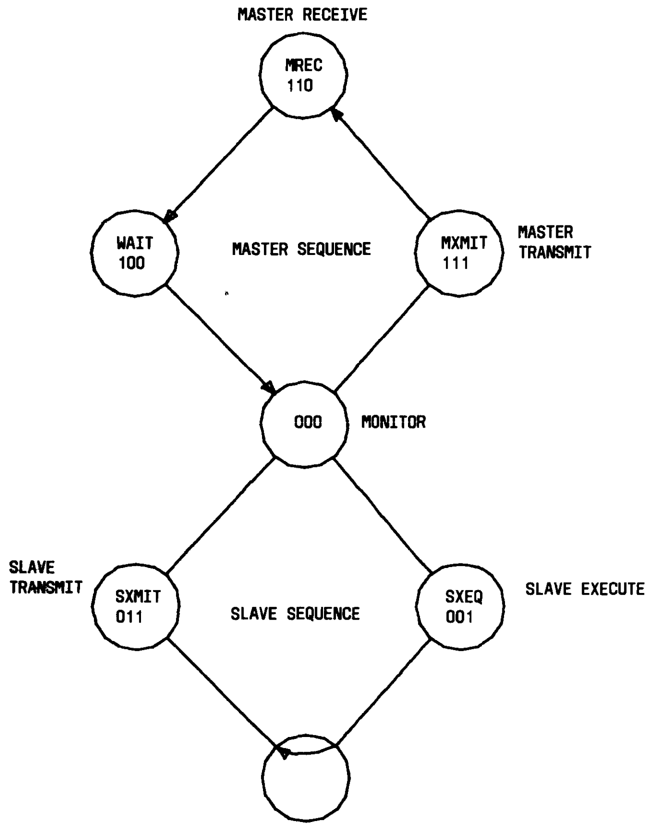


Fig. 53—Main Sequencer State Diagram

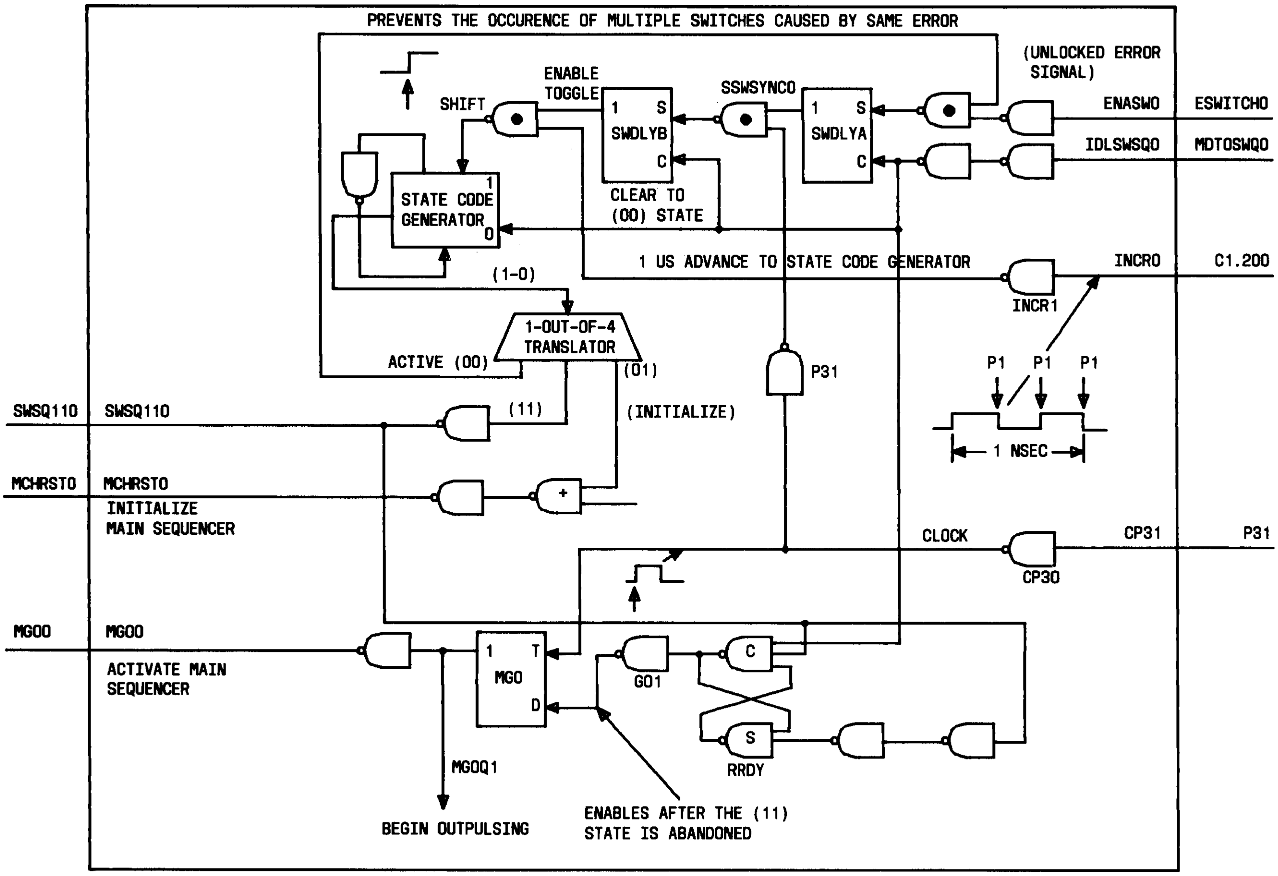


Fig. 54—Switch Sequencer—Simplified Logic Diagram

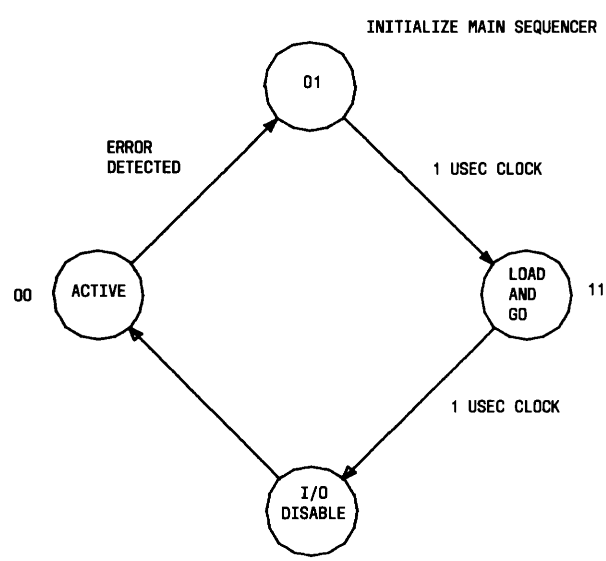


Fig. 55—Switch Sequencer State Diagram

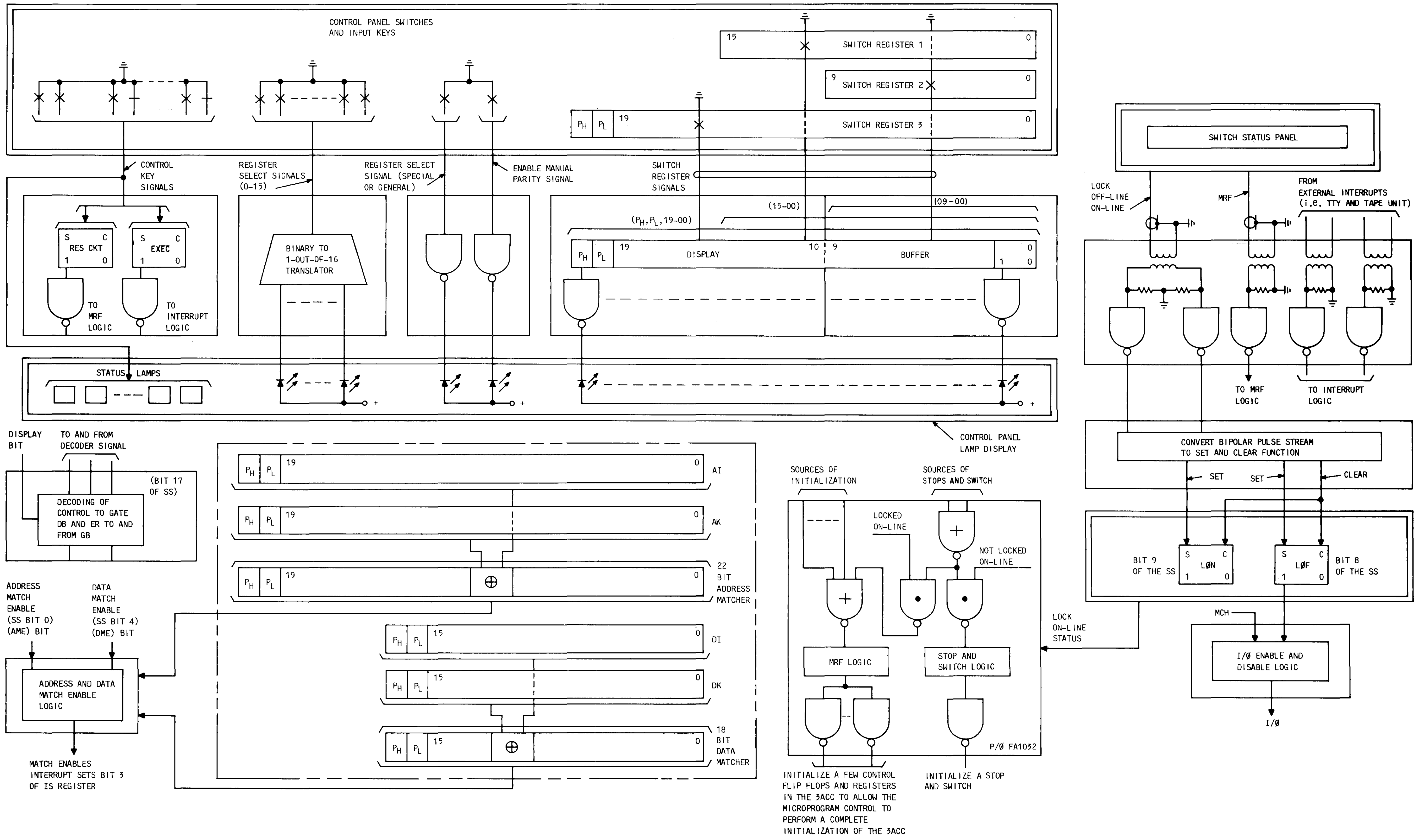


Fig. 56—3A CC Console Functions—Functional Block Diagram

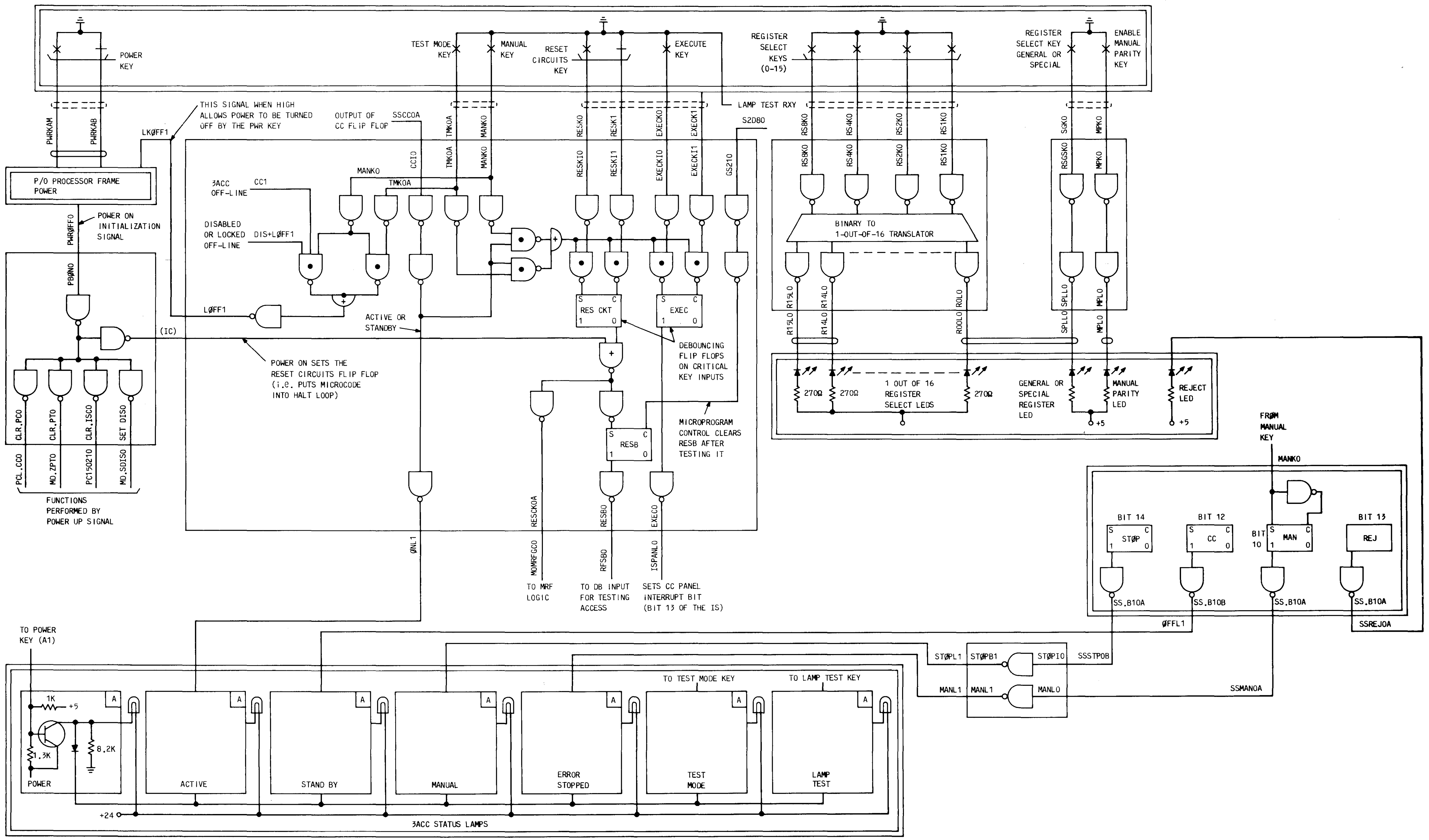


Fig. 57—3A CC Control Panel Interface—Simplified Logic Diagram

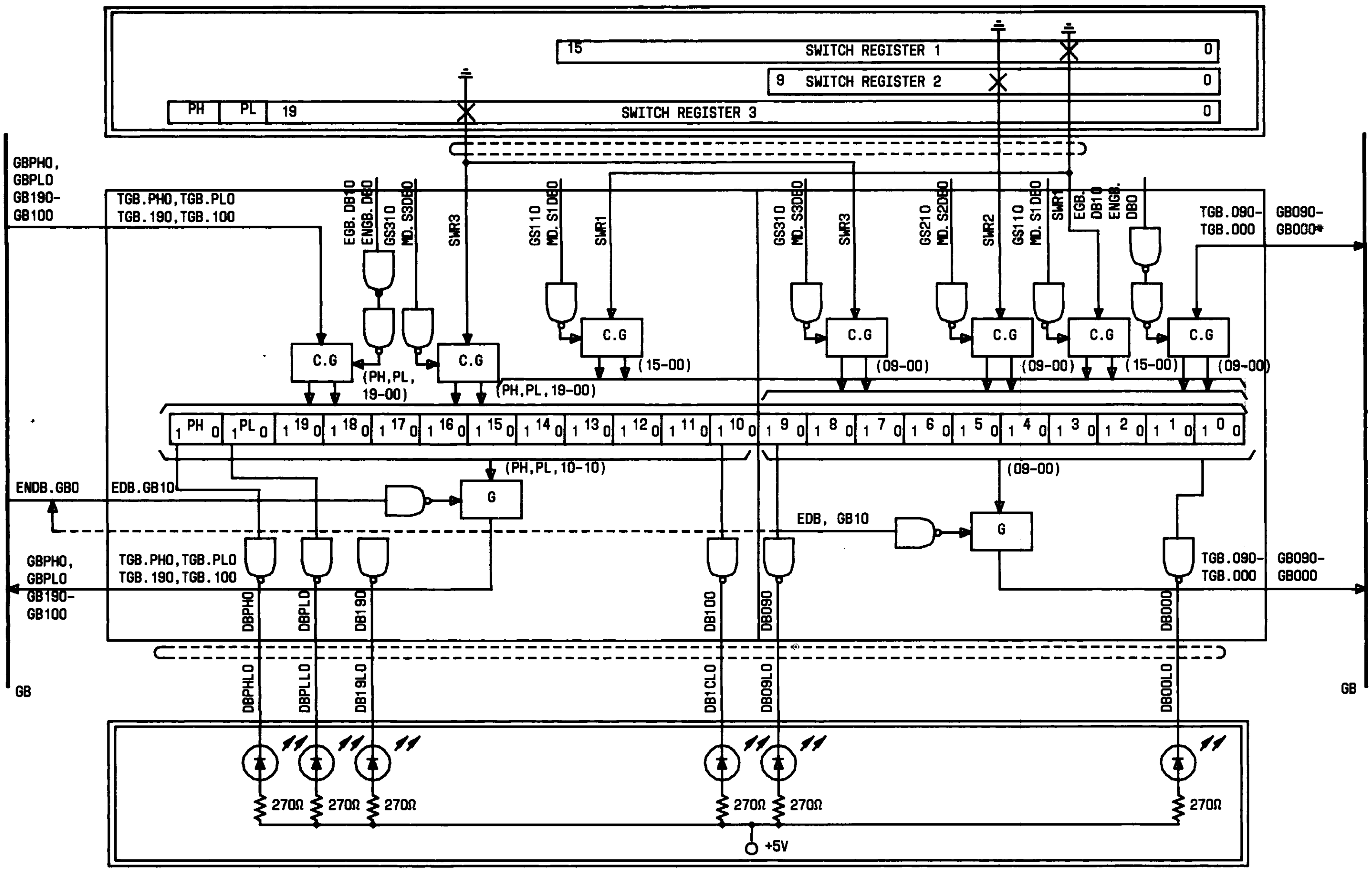
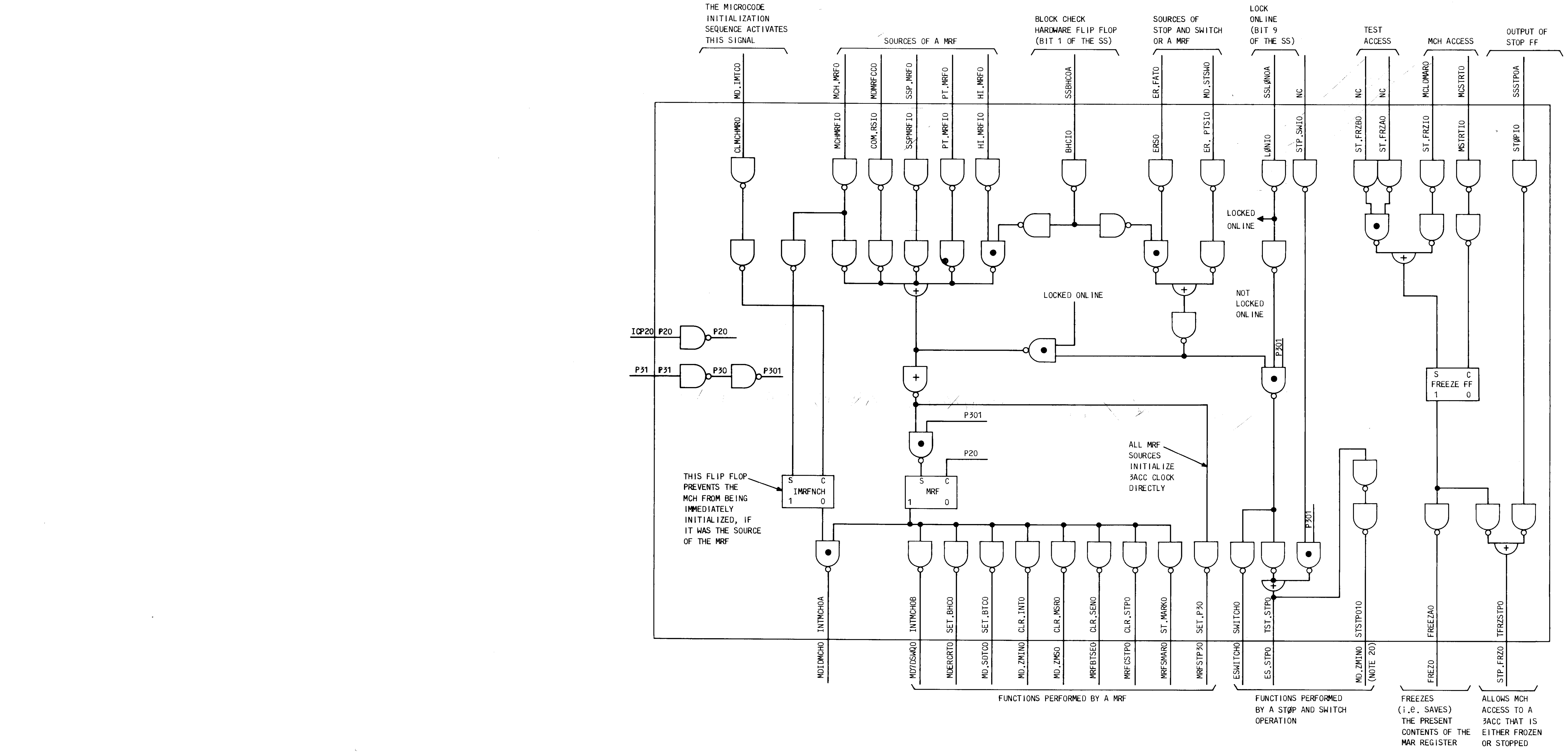


Fig. 58—Display Buffer—Functional Block Diagram



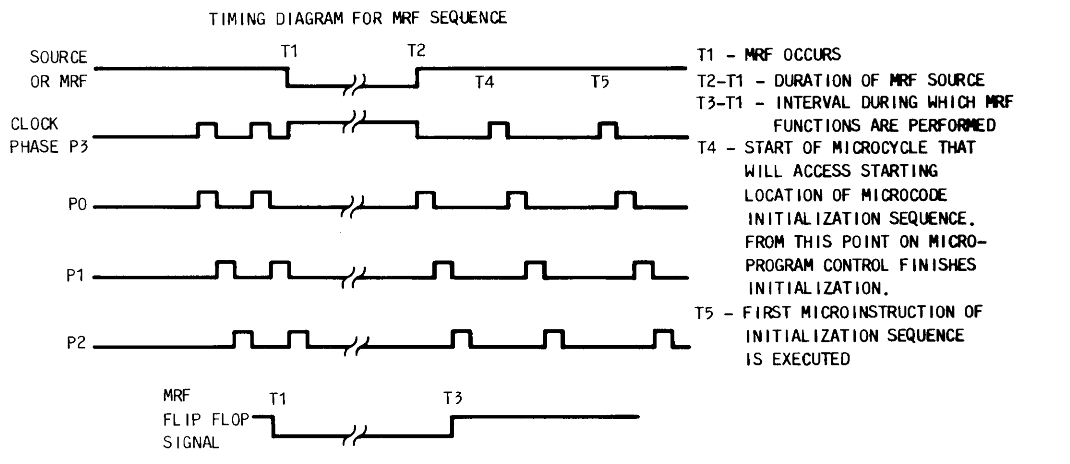


Fig. 59—Initialization and MRF Circuitry—Simplified Logic Diagram

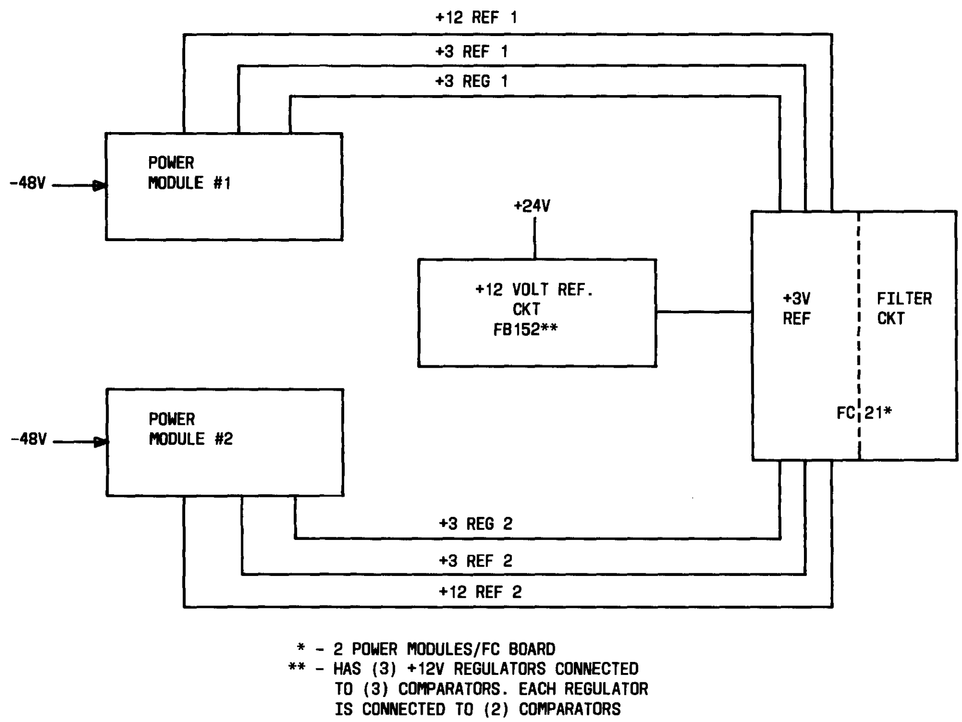


Fig. 60—Power System—Functional Block Diagram

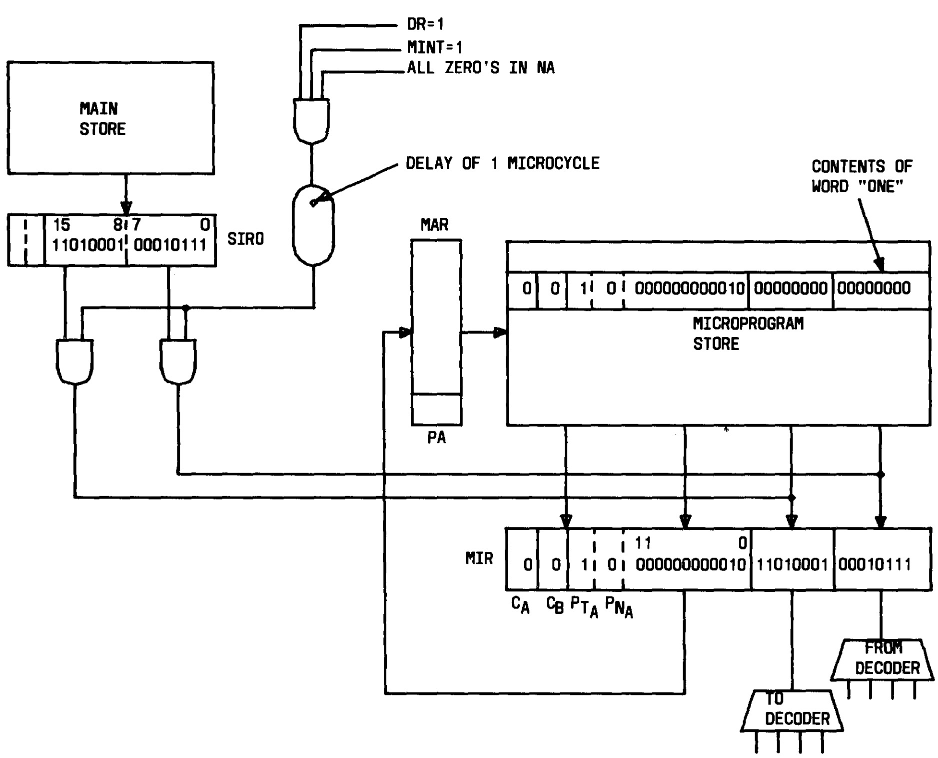


Fig. 61—Microinterpret Mode—Simplified Logic Diagram

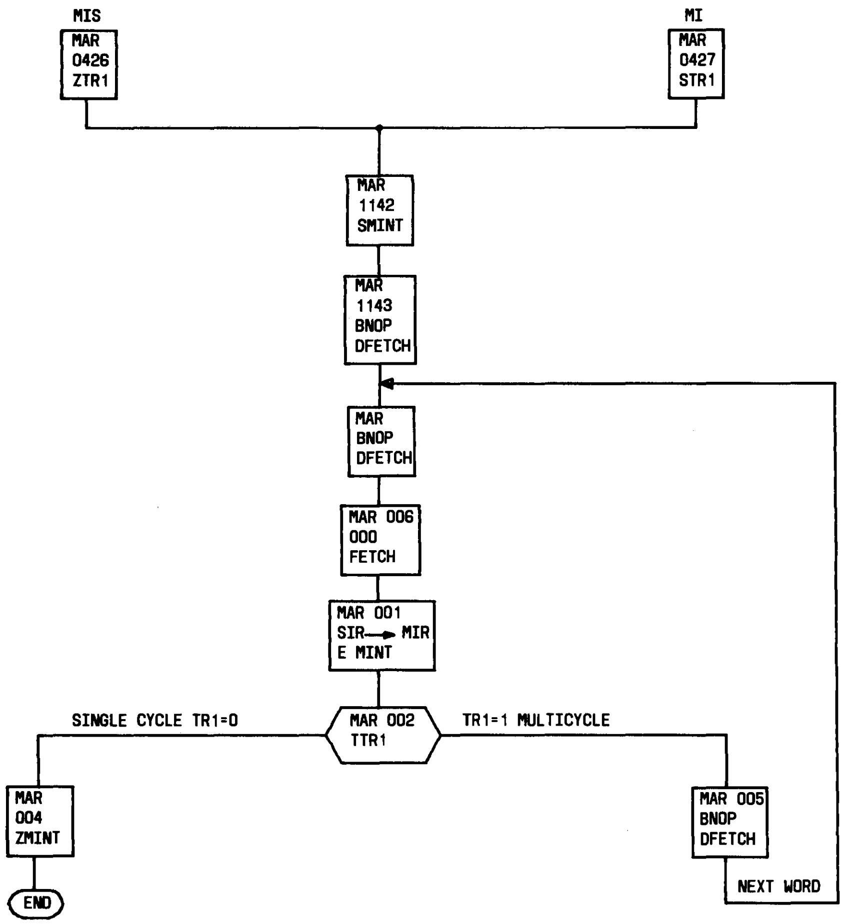


Fig. 62—Flowchart of Microinterpret Microsequence

TABLE A

DEFINITIONS OF TERMS USED IN DESCRIPTION OF INSTRUCTIONS

TERM	MEANING
RX or RY	One of the 16 general registers (R0-R15) within the CC. Definition of addressable 16-bit general purpose registers. 0 Register 0 1 Register 1 2 Register 2 3 Register 3 4 Register 4 5 Register 5 6 Register 6 7 Register 7 8 Register 8 9 Register 9 also used for input/output. 10 Register 10 also used for input/output. 11 Register 11 also used for input/output. 12 Register 12 also used for memory addressing. 13 Register 13 also used for memory addressing. 14 Register 14 also used for memory addressing. 15 Register 15 also used for memory addressing.
RA	Either the pair of general registers R12 and R13 or the pair of general registers R14 and R15 are used to reference a 20-bit address.
RA = R12 (RA0)	General register address pair of R12 and R13 are used to contain a 20-bit address (bits 3-0 of R12 and bits 15-0 of R13).
RA = R14 (RA1)	General register address pair of R14 and R15 are used to contain a 20-bit address (bits 3-0 of R14 and bits 15-0 of R15).
I	Immediate data

TABLE A (Contd)

DEFINITION OF TERMS USED IN DESCRIPTION OF INSTRUCTIONS

TERM	MEANING
OFFSET	8-bit index number
N	4-bit number (used as immediate data or index number)
M	Mask data bits
X	Low eight bits of an address
RS	May be any one of the special registers listed below: DB: Display Buffer SAR: Store Address Register PA: Program Address Register MCHTR: Maintenance Channel Transmit Receive Register (Load Only) AK: Address Mask AI: Address Input ER: Error Register SS: System Status Register MCHB: Maintenance Channel Buffer Register

TABLE B

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
System	SVC N	6.60	Branch to SVC entrance in OS, PO must point to message block.
Application Subroutine	BAL RP, Y	3.45	Branch to location Y and save return address in RP.
	BALX RP1, R1 (RP2)	4.25	Branch to location RP2 indexed by R1 and save return address in RP1.
	BALK RP1, K (RP2)	4.65	Branch to location RP2 indexed by K and save return address in RP1.
	RET RP	3.60	Return via address in register pair RP set DV IF (P0) = 0.
	RETSN RP, N	3.60	Set register 0 to N and return as in RET.
Load and Store	L R1, N (RA)	2.55	Load R1 with the contents of memory RA indexed by N and update RA.
	LA R1, N (RA)	2.55	Load R1 from RA indexed by N and update RA.
	LAL R1, R, RA	3.90	Load R1 with the contents of memory at location Y and set RA to Y.
	LAX R1, R2 (RA)	2.55	Load R1 from RA indexed by R1 update RA.
	LI R1, I	2.40	Load R1 with 16 bits of immediate data, I.
	LL R1, Y	3.75	Load R1 with the contents of memory at location Y.
	LN R1, N	1.20	Load R1 with 4 bits of immediate data, N.
	LR R1, R2	1.20	Load R1 with the contents of R2.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Load and Store (Contd)			
	LRM R1, R2, M	2.40	Load R1 from R2 under immediate mask, M.
	LRS R1, RS	1.20	Load R1 with the contents of RS.
	LX R1, R2 (RA)	2.55	Load R1 with the contents of memory at location RA indexed by R2.
	ST R1, N (RA)	2.70	Store R1 in location RA indexed by N.
	STA R1, N (RA)	2.70	Store R1 in memory at location RA indexed by N and update RA.
	STAL R1, Y, RA	4.05	Store R1 in memory at location Y and set RA to Y.
	STAX R1, R2 (RA)	2.70	Store R1 in memory at location RA indexed by R2 and update RA.
	STL R1, Y	3.75	Store R1 in memory at location Y.
	STM R1, N (RA), M	5.40	Insert R1 under mask into memory at location RA indexed by N.
	STVM R1, N (RA)	5.55	Insert R1 under variable mask into memory at location RA indexed by N.
	STX R1, R2 (RA)	2.70	Store R1 in location RA indexed by R2.
	EXR R1, R2	1.20	Exchange the contents of R1 with the contents of R2.
	ZR R1	1.20	Zero R1.
	LMP Y	5.25 + 1.5N*	Load multiple register specified by a data word immediately following this instruction from memory starting at the effective address.

*N = Number of words transferred.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Load and Store (Contd)			
	LMPX R1 (RP) LMPK K (RP)	6.30 + 1.5N 6.30 + 1.5N	Load multiple registers from location R indexed by R1 plus the positions of ones in mask.
	STMP Y	5.25 + 1.5N	Registers loaded correspond to the positions of ones in data word.
	STMPX R1 (RP) STMPK K (RP)	6.15 + 1.5N 6.15 + 1.5N	Store multiple registers specified by a data word which follows.
	LALL Y	27.90	Load all 16 general purpose registers starting at location Y.
	STALL Y	28.50	Store all 16 general purpose registers starting a location Y.
	LRPX RP1, R1 (RP2)	7.05	Load 20-bit address only in pair RP1 from location RP2 indexed by K.
	LRPK RP1, K (RP2)	7.05	Load 20-bit address only in register pair RP1 from location RP2 indexed by K.
	STRPX RP1, R1 (RP2)	7.50	Store 20-bit address only in memory at location RP2 indexed by R1 from RP1.
	STRPK RP1, K (RP2)	7.50	Store 20-bit address only in memory at location RP2 indexed by K from RP1.
	LRPI RP, IY	2.70	Load 20-bit immediate data IY.
	LRP RP, Y	6.75	Load 20-bit address from location Y.
	STRP RP, Y	7.20	Store 20-bit address in location Y.
	LXX R1, R2 (RP)	4.65	Load R1 from RP indexed by R2.
	LK R1, K (RP)	4.65	Load R1 from RP indexed by K.
	STXX R1, R2 (RP)	4.50	Store R1 into location RP indexed by R2.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
	STK R1, K (RP)	4.50	Store R1 into location RP indexed by K.
	LAXX R1, R2 (RP)	6.00	Load R1 from location RP indexed by R2 and update RP.
	LAK R1, K (RP)	6.00	Load R1 from location RP indexed by R2 and update RP.
	STAXX R1, R2 (RP)	5.85	Store R1 into location RP indexed by R2 and update RP.
	STAK R1, K (RP)	5.85	Store R1 into location RP indexed by K and update RP.
Arithmetic	AI R1, I	2.40	Add 16 bits of immediate data, I, to R1 CF set upon carryout of bit 15.
	AN R1, N	1.20	Add 4 bits of immediate data, N, to R1 CF set upon carryout of bit 15.
	AR R1, R2	1.20	Add R2 to R1 and store the result in R1CF set upon carry-out of bit 15.
	SI R1, I	2.40	Subtract 16 bits of immediate data, I, from R1 CF set upon carryout of bit 15.
	SN R1, N	1.20	Subtract 4 bits of immediate data, N, from R1 CF set upon carryout of bit 15.
	SR R1, R2	1.35	Subtract R2 from R1 and store the result in R1 CF set upon carryout of bit 15.
	S1S Y	5.85	Subtract 1 from contents of location Y.
	S1SX R1 (RP)	5.85	Subtract 1 from the contents of location RP indexed by K.
	S1SK K (RP)	5.85	Subtract 1 from the contents of location RP indexed by K.
	A1S Y	5.40	Add 1 to the contents of memory at location Y CF set upon carryout of bit 15.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Arithmetic (Contd)			
	A1SX R1 (RP)	6.75	Add 1 to contents of location RP indexed by R1.
	A1SK K (RP)	6.75	Add 1 to contents of location RP indexed by K.
	CAR R1, R2	2.40	Compare arithmetic contents of R1 and R2 set CF if (R1) < (R2).
	AAR R1, R2	1.95/3.00	Add R1 to R2 and place result in R1; arithmetic overflow sets CF.
	SAR R1, R2	3.45/2.40	Subtract R2 from R1 and place result in R1; arithmetic overflow sets CF.
Logical			
	COM R1[,R2]	1.20	Complement R1/[R2] and store in R1.
	NI R1, I	2.40	AND 16 bits of immediate data, I, to R1 and store in R1.
	NR R1, R2	1.20	AND R2 to R1 and store in R1.
	OI R1, I	2.40	Inclusive OR 16 bits of immediate data, I, to R1 and store in R1.
	OR R1, R2	1.20	Inclusive OR R2 to R1 and store in R1.
	XI R1, I	2.40	Exclusive OR 16 bits of immediate data, I, to R1 and store in R1.
	XR R1, R2	1.20	Exclusive OR R2 to R1 and store in R1.
Bit	FLZ R1, R2	1.95/1.20	Find low zero in R1 and record its position in R2.
	ICF R1, N	1.20	Insert CF in bit N of R1.
	IRM R1, R2, M	2.40	Insert R2 into R1 under immediate mask M.
	SBN R1, N	1.20	Set bit B in R1.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Bit (Contd)	SBR R1, R2	1.20	Set bit in R1 determined by the low four bits of R2.
	SBS N (RA), B	4.35	Set bit B in memory word at location determined by adding N to RA.
	SCF	1.20	Set the condition-flop.
	ZBN R1, B	1.20	Zero bit B in R1.
	ZBR R1, R2	1.20	Zero bit in R1 determined by the low four bits of R2.
	ZBS N (RA), B	4.35	Zero bit B in memory word at location determined by adding N to RA.
	ZCF	1.20	Zero the CF.
Rotate	RL R1, R2	2.70	Rotate R1 left an amount determined by the low four bits of R2.
	RLN R1, N	1.20	Rotate R1 left by N bit positions.
	RR R1, R2	1.40	Rotate R1 right an amount determined by the low four bits of R2.
	RRN R1, N	1.20	Rotate R1 right by N bit positions.
Shift	SRLN R1, N	4.20	Shift R1 right N places with zero fill.
	SRL R1, R2	4.65	Shift R1 right an amount determined by low four bits of R2.
	SLLN R1, N	6.00	Shift R1 left N places with zero fill.
	SLL R1, R2	6.00	Shift R1 left an amount determined by low four bits of R2.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Test			
	CI R1, I	2.40	Compare R1 to 16 bits of immediate data, I.
	CIRM R1, I, N, M	2.40	Compare eight bits of R1 rotated by N with immediate data and mask.
	CR R1, R2	1.20	Compare R2 to R1.
	CRM R1, R2, M	2.40	Compare R2 to R1 under 16-bit immediate mask.
	TBN R1, B	1.20	Test Bit B in R1.
	TBR R1, R2	1.20	Test bits in R1 determined by low four bits of R2.
	TBS N (RA), B	3.30	Test bit B in memory word at location determined by adding N to RA.
	TRPH R1	1.20	Test general register parity high.
	TRPL R1	1.20	Test general register parity low.
	TSRPH RS	1.20	Test special register parity high.
	TSRPI RS	1.20	Test special register parity low.
	TZ R1	1.20	Test R1 for all zeros.
Branch			
	B Y	1.80	Branch to location Y.
	BC Y	1.95/1.50	Branch on condition to location Y.
	BCL Y	2.70/1.80	Branch long on condition to location Y.
	BL Y	2.70	Branch long to location Y.
	BNC Y	1.50/1.95	Branch on not condition to location Y.
	BNCL Y	1.80/2.70	Branch long on not condition to location Y.
	BPAX R1	1.65	Branch to location PA indexed by R1.
	BR N (RA)	2.70	Branch to location RA indexed by N.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Branch (Contd)			
	BRX R1 (RA)	2.55	Branch to location RA indexed by R1.
	BX R1, Y	2.55/2.70	Branch on index not zero to location Y.
	CASR R1, R2	2.85/5.70	Arithmetic compare if R1 > R2 execute next instruction, if R1 = R2 skip one word, if R1 < R2 skip two words.
Miscellaneous			
	NOP	1.20	No operation.
	PIE	4.20	Program interrupt end.
	SOP	1.20	Set OP code.
	ZOP	1.20	Zero OP-code fill bit.
Memory to Memory			
	MBLK RP1, RP2	7.95 + 2.7N*	Move block of (R0) words starting at location RP1 to block starting at RP2, R0 must be preloaded.
Communications (ref 9)			
	SCy R1, K (RP)	18.30/35.70	Store communications character from R1 according to parameter block at EA = K + (RP).
	ICy R1, K (RP)	16.20/33.60	Load communication character into R1 according to parameter block at EA.
	CRC R1, K (RP)	16.05	Update cyclic redundancy word at effective address using character in low eight bits of R1.
	XLAT R1, K (RP)	10.20	Translate character in R1 according to translation table and place translated character in R1 unless bit 15 of table entry is zero in which case branch.

*N = Number of words transferred.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Communications (ref 9) (Contd)	LB R1, RP	5.40	Load byte from buffer where byte index and base address of byte string are found in RP.
	LBA R1, RP	5.55	Load byte from buffer and increment index where byte index and base address are found in RP.
	STB R1, RP	6.45	Store byte in buffer byte where byte index and base address are found in RP.
	STBA R1, RP	6.0	Store byte in buffer and increment index where byte index and base address are found in RP.
Subroutine Privileged State	BSA Y	6.15	Branch to location Y and save address.
	BSAI X	5.70	Branch and save address indirect.
	BTSA	4.95	Branch to saved address.
	BTSAG	31.20	Get registers 2 through 15 and branch to saved address.
	BTSAGN N	31.35	Get registers, load return code, and branch to saved address.
	BTSAN N	5.10	Load return code and branch to saved address.
	GA	26.55	Get registers 2 through 15 from words 2 through 15 of hold-get area.
	GN R1, N	2.55	Get R1 from word N of hold-get area.
	HA	28.55	Hold registers 2 through 15 in words 2 through 15 of hold-get area.
	HALT	28.55	Halt the central control.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
Subroutine Privileged State (Contd)			
	HN R1, N	2.70	Hold R1 in word N of hold-get area.
Special Register Privileged State			
	LSR RS, R1	1.20	Load RS with the contents of R1.
	PACK RS	1.20	Packs general registers 2 and 3 into 20-bit special register RS.
	UNPK RS	1.20	Unpacks 20-bit special register RS to general registers 2 and 3.
Maintenance Privileged State			
	MI	1.20	Microinterpret.
	MIS	2.40	Single cycle microinterpret.
	MSTF N (RA)	4.65	Maintenance store function using Register 0 at location RA indexed by N.
	MSTFX R1 (RA)	4.50	Maintenance store function using Register 0 at location RA indexed by R1.
	TCC1	1.20	Test central control 1.
I/O Privileged State			
	SIO	1.65	Send I/O message over channel and subchannel defined in R9.
	SMIO	1.65	Send maintenance I/O message over channel and subchannel defined in R9.
	TCH	1.65	Test the channel defined in R9 for the idle state.
	TIO	1.65	Test for I/O message in channel defined in R9.

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	FUNCTION
I/O Privileged State (Contd)			
	TMIO	1.65	Test for maintenance I/O message in channel defined in R9.
	XIO	1.65	Idle the channel defined in R9.

CLASS	INSTRUCTION	EST EXEC TIME USEC	CURRENT EXEC TIME (NO PM TEST)	FUNCTION
I/O Privileged State				
	RD	3.30	2.85	Read data and parity from device specified in R9 into R11 (parity test).
	RDP	3.30	2.85	Read data and generate parity (no test).
	WD	3.00	2.55	Write data in R10 with parity to device specified in R9.
	SSA	4.80	4.35	Sense status/test address load R11 with status from device specified by R9, R9 (0-5) = R11 (0-5).
	SSPA	4.80	4.35	Sense status/test address and generate parity R9 (0-5) = R11 (0-5).
	SS	3.75	3.30	Sense status equality of R9 (0-5) = R11 (0-5) not assumed.
	SSP	3.75	3.30	Sense status generate parity equality of R9 (0-5) = R11 (0-5), not assumed.
	SDC	3.15	2.70	Send device command from R12 to device specified in R9.
	SCB	3.30	3.00	See CF on busy device specified in R9.
	IID	3.30	2.85	Identify interrupting device on subchannel specified by R9 (bits 15 through 6).

TABLE B (Contd)

COMBINED BASIC AND EXTENDED 3A CC INSTRUCTION SET

CLASS	INSTRUCTION	EST EXEC TIME USEC	CURRENT EXEC TIME (NO PM TEST)	FUNCTION
I/O Privileged State (Contd)				
	IP	2.10	1.65	Initialize periphery (all devices on channel R9 [bits 15 through 10]).
	CMCi	1.20	1.20	Maintenance commands (ref 3).
DMA Privileged State				
	WDMA REG	5.55/8.70		Write DMA register.
	RDMA REG	3.60/6.75		Read DMA register.
	MDMA	2.70/5.85		Read/write memory.
	IDMA	1.20		Initialize DMA.
	IDMA	1.20		Lock DMA.
	SDMA	1.20		Step DMA.
	TDMA	4.95/8.10		Test DMA decoder.

Note 1: On all I/O and DMA instructions, an ERROR condition causes the control-flop to be reset to zero. Maintenance, special register, PCH, DMA, and SCH instructions in the basic set, and all subroutine instructions from the basic set will only be executable in the privileged state (ie, after a supervisor call [SVC]).

Note 2: The execution time of some DMA instructions varies because it may take up to 3.15 μsec for the DMA to respond to the CC request.

TABLE C

DEFINITION OF ADDRESSABLE SPECIAL REGISTERS

THE FOLLOWING TABLE DEFINES THOSE REGISTERS WHICH ARE ACCESSED WITH THE SPECIAL REGISTER INSTRUCTIONS:

REG	TO FIELD FUNCTION		FROM FIELD FUNCTION	
0	GB ==> MCMTR	22-Bit Path	TI ==> GB	18-Bit Path P_L=0; P_H=0
1	GB ==> SAR	22-Bit Path	SAR ==> GB	22-Bit Path
2	GB ==> PA	22-Bit Path	PA ==> GB	22-Bit Path
3	GB ==> MCHB	22-Bit Path	MCHB ==> GB	22-Bit Path
4	Unassigned		MMSR ==> GB	14-Bit Path P_L=0; P_H=1
5	GB ==> AK	22-Bit Path	AK ==> GB	22-Bit Path
6	GB ==> AI	22-Bit Path	AI ==> GB	22-Bit Path
7	GB ==> DK	18-Bit Path	DK ==> GB	18-Bit Path
8	GB ==> DI	18-Bit Path	DI ==> GB	18-Bit Path
9	GB ==> DB	22-Bit Path	DB ==> GB	22-Bit Path
10	GB ==> ER	22-Bit Path	ER ==> GB	22-Bit Path
11	GB ==> DB If Display Bit in SS Is 1	22-Bit Path	Unassigned	
12	GB ==> IM	18-Bit Path	IM ==> GB	18-Bit Path
13	GB ==> SS_S	A 1 Sets SS	IS ==> GB	18-Bit Path
14	GB ==> MS	18-Bit Path	MS ==> GB	18-Bit Path
15	GB ==> SS_R	A 1 Resets SS	SS ==> GB	22-Bit Path P_L=CC; P_H=CC

TABLE D

INTERRUPT SET REGISTER BIT ASSIGNMENTS

IS BIT AND LEVEL	FUNCTION	SOFTWARE INTERRUPT
0	INTP A	
1	Utility Interrupt (External) INTP B	INTe
2	INTP C	
3	Address or Data Match	
4	INTP D	INTk
5	Error Register (Internal)	
6	INTP E	INTJ
7	Other CC (External)	
8	Hardware Interrupt (Timing Counter 5)	
9	Hardware Interrupt (Timing Counter 10)	
10	TTY and Tape-Even (External)	INTa/INTb
11	TTY and Tape-Odd (External)	INTc/INTd
12	INTP F	INTh
13	Manual Panel Execute	
14	INTP G	INTg
15	INTP H	INTf

TABLE E

MAINTENANCE STATE REGISTER FUNCTIONS

BIT	FUNCTION	MNEMONIC
0	Override Lock Off Line	OVLOF
1 2 3	Clock Test Conditions	CLK 1 CLK 2 CLK 3
4	Ground My Store Go Lead	STRGO
5		
6	Utility Freeze	FREEZ
7	Ground-My Store Complete Lead, My Store Busy Lead, and Inhibit Other Store Complete	MCMPT
8	Ground-Other Store Complete Lead, Other Store Busy Lead, and Inhibit My Store Complete	OCMPT
9	ENABLE IB(X, Y) to MIR When Stopped and Partially Inhibit From Check	XYGATE
10	Enable RAR — MAR Match Independent of RU Flip-Flop Plus Divert Store GO Flip-Flops into Store Write Protect Error Bits	MCADCMP
11	Hold MAR Parity	DISMARP
12	Disable to Field Decoder	DISTO
13	Code Merger Test 1	COM 1
14	Code Merger Test 2	COM 2
15	Jam Store Control Lead 3 = 1	STRIS

TABLE F

SYSTEM STATUS REGISTER

BIT	FUNCTION	MNEMONIC
0	ADDRESS MATCH ENABLE: Enables address matching between store address register and address input register.	AME
1	BLOCK HARDWARE CHECK: Disables error register output which would cause processors to switch on-line status.	BHC
2	BLOCK INTERRUPT: Inhibits recognition of any interrupts.	BIN
3	BLOCK TIMER CHECK: Inhibits inputs and outputs of program timer.	BTC
4	DATA MATCH ENABLE: Enables data matching between store data register and data input register.	DME
5	HALT: Indicates that the 3A CC is not executing an instruction.	HLT
6	INITIALIZATION SANITY CHECK 1: Initialization sanity check verifies hardware initialization routine. When a failure occurs, the ISC 1 bit is set and the off-line 3A CC is switched on-line.	ISC 1
7	INITIALIZATION SANITY CHECK 2: Further check of the initialization routine. ISC 1 ISC 2 0 0 Transfers control to address 0(40) in main memory. 1 0 Switches to off-line 3A CC. 1 1 Reloads memory from tape.	ISC 2
8	LOCK OFF LINE: Disables the I/O channel so that the off-line 3A CC cannot interfere with the on-line machine. LOF can also be manually set from system status panel.	LOF
9	LOCK ON LINE: Prevents on-line 3A CC from being switched off-line. LON can be manually set from system status panel.	LON
10	MANUAL: Places 3A CC in manual mode and lights "manual" lamp on 3A CC panel. This bit is set by operations of "manual" switch on 3A CC panel.	MAN
11	MICROINTERPRET MODE: Enables gating of contents of store instruction register to microstore instruction register in order to allow execution of microinstructions from main store.	MINT
12	CENTRAL CONTROL: Indicates whether the 3A CC is on-line (CC = 1) or off-line (CC = 0). Other functions are enabled to prevent interference from the off-line 3A CC.	CC
13	REJECT: Lights "reject" panel lamp to indicate that a requested panel action has been rejected.	REJ
14	STOP: Sets a maintenance routine address into the microstore address register which causes all zeros to be read out of microstore.	STP
15	DISABLE: Reads state of disable flip-flop in I/O enable and disable logic.	DISA
16	PRIVILEGE: Used for certain applications of 3A CC that require a privilege mode for instruction execution. Not used in No. 3 ESS.	PRI
17	DISPLAY: Enables gating of the contents of program register to display buffer.	DISP
18	BLOCK BUS PARITY CHECK: Inhibits program gating bus parity check circuits.	BPC

TABLE F (Contd)

SYSTEM STATUS REGISTER

BIT	FUNCTION	MNEMONIC
19	INITIATE PROGRAM LOAD TRACK: Used by the microcode in the sequence that initiates a program reload from tape. The state of this bit determines which of two tracks on tape will be read.	IPLTRK
20, 21	CENTRAL CONTROL 0, CENTRAL CONTROL 1 Hardwired to determine which 3A CC is CC0 and which is CC1. For CC0, $P_L = 1$ and $P_H = 0$. For CC1, $P_L = 0$ and $P_H = 1$.	(P_L and P_H)

TABLE G

ERROR REGISTER

BIT	FUNCTION	SYMBOL
0	TO Decoder Error	TODER
1	FROM Decoder Error	FRMDER
2	IB X, Y Field Error	IBER
3	BUS Parity Error	BUSER
4	DML Match Error	DMLER
5	MAR Parity Error	MARER
6	Clock Error	CLKER
7	My Store Error A	MSTRER
8	MAR — RAR Match Error	MADER
9	Function Register Parity Error	FRER
10	Store Read Parity Error (Error C)	SRPE
11	My Store Write Protect Error (Error B)	MSTWRP
12	My Store Fast Time-Out Error	MFSTER
13	Branch Allowed Error	BAER
14	Other Store Write Protect Error	OWRTER
15	Other Store Error	OSTRER
16	Other Store Fast Time-Out Error	OFSTM
17	I/O Multiple Channel Select Error	IOMLTER
18	PT Reset Received by On-Line CC Error	PTRER
19	Switch Received by On-Line CC Error	SWER
P_L	I/O Channel Error	IOER
P_H	I/O Bad Parity Received Error	IOPARER

TABLE H

MCS REGISTER AND BIT SIGNIFICANCE

CF — Conditional flip-flop bit is used to pass status information to main program, ie, to notify program of results of a logical comparison of an operand set.

DS — Data manipulation logic status bit stores results of certain DML operations, ie, adder overflow, all ones in a find low zero test.

TR1 — Test register 1 bit is a general purpose status used by microprogram control to indicate some state occurred, ie, which RA has memory address, was read or write requested, etc.

TR2 — Same as TR1

DR — Data ready bit is used to indicate the last previously initiated main memory operator is complete.

RU — RAR update bit controls the updating junction of the return address register used in conjunction with microprogram subroutines. When set, RAR contents are saved, not updated as new MAR addresses are developed, to provide a return address after a microsubroutine is completed.

BIT	MNEMONIC	COPY*
0	CF	0
1	CF	1
2	DS	0
3	DS	1
4	TR 1	0
5	TR 1	1
6	TR 2	0
7	TR 2	1
8	DR	0
9	DR	1
10	RU	0
11	RU	1
12	IFF	0
13	IFF	1
14	FIL	0
15	FIL	1
16	MARP	0
17	MARP	1
18		
19		
PL	PL	01
PH	PH	01

* Identifies Duplicate MCS Bit Assignment.

IFF — I-bit applies to applications of 3A CC where main memory words exceed 16 bits in length. When activated, extensions of SIR and SDR are utilized to buffer the enlarged memory words. I-bit also determines from which portion of the SIR the next OP code is to be obtained. This bit is not utilized in the 3A CC application within a No. 3 ESS.

OPF — Op Code FIL bit applies to application of the 3A CC where number of operational codes is expanded up to 256 codes. When clear microstore area 256 to 384 is referenced by a new op code. When set, microstore area 2048 to 2176 is referenced by a new op code. This bit is not utilized in the 3A CC application with in a No. 3 ESS.

MARP — Microstore address register parity bit fulfils a internal check function instead of a control function as the other MCS bits. It reflects the status of the MAR parity.

TABLE I

LAYOUT OF A HOLD GET AREA BLOCK (16 WORDS)

P_H	P_L	OP	Words 0 and 1 are reserved for return address. Bits (19—16) of RA
P_H	P_L		Bits (15—0) of return address
P_H	P_L		Word 2 reserved for Register 2 when held by Hold All instruction.
P_H	P_L		Word 3 reserved for Register 3 when held by Hold All instruction.
P_H	P_L		Word 4 reserved for Register 4 when held by Hold All instruction.
P_H	P_L		Word 5 reserved for Register 5 when held by Hold All instruction.
P_H	P_L		Word 6 reserved for Register 6 when held by Hold All instruction.
P_H	P_L		Word 7 reserved for Register 7 when held by Hold All instruction.
P_H	P_L		Word 8 reserved for Register 8 when held by Hold All instruction.
P_H	P_L		Word 9 reserved for Register 9 when held by Hold All instruction.
P_H	P_L		Word 10 reserved for Register 10 when held by Hold All instruction.
P_H	P_L		Word 11 reserved for Register 11 when held by Hold All instruction.
P_H	P_L		Word 12 reserved for Register 12 when held by Hold All instruction.
P_H	P_L		Word 13 reserved for Register 13 when held by Hold All instruction.
P_H	P_L		Word 14 reserved for Register 14 when held by Hold All instruction.
P_H	P_L		Word 15 reserved for Register 15 when held by Hold All instruction.
		15	0

1. Words 0 and 1 are held by BSA and TSA instructions and interrupts.

TABLE J

DML BOOLEAN FUNCTIONS

FR BIT				FUNCTION PERFORMED
7	6	5	4	f(AR, BR)
0	0	0	0	1
0	0	0	1	$\overline{AR} + \overline{BR}$
0	0	1	0	$\overline{AR} + \overline{BR}$
0	0	1	1	$\overline{AR}$
0	1	0	0	$AR + \overline{BR}$
0	1	0	1	$\overline{BR}$
0	1	1	0	$\overline{AR} \bullet \overline{BR} + AR \bullet BR$
0	1	1	1	$\overline{AR} \bullet \overline{BR}$
1	0	0	0	$AR + BR$
1	0	0	1	$\overline{AR} \bullet BR + AR \bullet \overline{BR}$
1	0	1	0	BR
1	0	1	1	$\overline{AR} \bullet BR$
1	1	0	0	AR
1	1	0	1	$AR \bullet \overline{BR}$
1	1	1	0	$AR \bullet BR$
1	1	1	1	0

TABLE K

MM BIT FUNCTIONS

MM 2	MM 1	RW	SC3-SC0	FUNCTION
0	0	0	0011	Normal Write
0	0	1	1100	Normal Read
0	1	0	0110	Write the Write Protect Register
0	1	1	1001	Read the Write Protect Register
1	1	0	0011	Normal Read or Write with the PBC Locked.
1	1	1	1100	

TABLE L

MMS REGISTER FORMAT

15																	0	
P_H	P_L	BLC	CW	BD SR	BD SR	ISO	ISO	UPD	UPD	IDL	IDL	RW	RW	MM	MM	MM	MM	
														2	2	1	1	
1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	Copy 0 or 1

Note: CW and BEC are gated to bits 14, 16 and 15, 17, respectively, of the gating bus as necessary to maintain good parity on the bus.

TABLE M

I/O MAIN CHANNEL 00 TYPICAL SUBCHANNEL ASSIGNMENTS

SUBCHANNEL	3/6 CODE	ASSIGNMENT
00	000111	System Status Panel
01	001011	(Spare)
*02	001101	TDC 0
*03	001110	TDC 1
04	010011	(Spare)
05	010101	TTYC 0 (MTCE)
06	010110	TTYC 1 (MTCE)
*07	011001	MAS0 (Associated MAS)
08	011010	MAS0 (Other MAS)
09	011100	(Spare)
10	100011	(Spare)
11	100101	(Spare)
12	100110	(Spare)
13	101001	(Spare)
14	101010	(Spare)
15	101100	(Spare)
16	110001	(Spare)
17	110010	(Spare)
18	110100	(Spare)
19	111000	(Spare)

Note: Subchannels 0-9 are on subchannel board 1 and subchannels 10-19 are on subchannel board 2.

*I/O Main Channel 0, Subchannels 2, 3, and 7 are hard coded in the microprogram. The remaining subchannels are available for usage as required.

TABLE N

I/O MISCELLANEOUS DECODER SIGNALS

MNEMONIC	FUNCTION
MDOIDCH	Loads the SCS field in the 10S from R-9, places sequencer in idle state, initializes IOC and enables gating.
MD1CHTN	Loads SCS field in IOS from R9, initializes SCTL field for transmit normal sequence.
MD2CHTN	Loads SCS field in IOCS from R9, initializes SCTL field for transmit maintenance sequence.
MD3CHC	Loads SCS field and SCTL field from R9, sets IOS bits 5, 4 to 0, 1.
MD4STH	Starts main channel sequencer and inhibits gating.
MD5EIOS	Enables gating from the IOS to the R11 gating bus. Maintenance only.
MD6RAIO	Gates R10 to IOD.
IODXR11	GATES IOD TO R11 via R11 gating bus.
TCH	Test I/O channel. Loads 3A CC status flip-flops with following flags: Idle — Sets DS Message Complete Normal — Sets TR1 Message Complete Maintenance — Sets TR2

TABLE O

IOC START CODES

CODE	FLAG	CONDITION
011	Normal	Normal Message Present
101	Maintenance	Maintenance Message Present
111 001	Normal	Illegal Start Code Present

TABLE P

CHANNEL NUMBER ASSIGNMENTS

CHANNEL NUMBER	3/6 CODE	UNIT ASSIGNMENT
0	000111	SCH_0
1	001011	SCH_1
2	001101	SCH_2
3	001110	PCH_0
4	010011	PCH_1
5	010101	DMA (PCH)
6	010110	DMA (UNIT)
7	011001	NOT ASSIGNED
8	011010	
9	011100	
10	100011	
11	100101	
12	100110	
13	101001	
14	101010	
15	101100	
16	110001	
17	110010	
18	110100	
19	111000	

TABLE Q

MPCH STATUS WORD FORMAT

BIT	SIGNIFICANCE
15	R11 P_L Bit
14	MFOL (Maintenance Flag Low Byte)
13	IFOL (Information Flag Low Byte)
12	SCAC 4)
11	SCAC 3) Subchannel Response
10	SCAC 2)
9	SCAC 1)
8	SCAC 0)
7	R11 P_H Bit
6	MFOH (Maintenance Flag High Byte)
5	IFOH (Information Flag High Byte)
4	CFO (Command Flip-Flop)
3	AFO (Address Flip-Flop)
2	PCHENA (PCH Controlled by DMA or 3A CC)
1	ENA (Channel Unit in Active State)
0	IO ER (Sum of IO Error Check)

TABLE R

MAINTENANCE CHANNEL COMMANDS

MNEMONIC	ACTION IN RECEIVING	CONDITIONAL ON	4-OUT-OF-8 CODE (HEX) (SEE NOTE 1)
CLER	CLEAR ERROR REGISTER		65
CLMSR	CLEAR MAINTENANCE STATES REGISTER		35
CLPT	CLEAR PROGRAM TIMER	CC=OFF-LINE	C5
CLTTO	CLEAR TIMER TIMEOUT REGISTER		2D
DISA	IO DISABLE A		A5
DISB	IO DISABLE B		E1
INITCLK	INITIALIZE CLOCK (BST)+	CC=OFF-LINE	47
LDMAR	CLEAR STP;MAR-MCHTR;SET FREEZE	CC=OFF-LINE AND EITHER STOP OR FREEZE	4D
LDMCHB	MCHB-MCHTR	STOP OR FREEZE	99
LDMIRH	MIRH-MCHTR	STOP OR FREEZE	8D
LDMIRL	MIRL-MCHTR;EXECUTE ONE CYCLE	CC=OFF-LINE	1D
MSTART	CLEAR FREEZE	CC=OFF-LINE	C9
MSTOP	SET STOP FLIP-FLIP		D1
*RTNER	MCHTR-ER;TRANSMIT (BST)		88
*RTNMB	MCHTR-MB;TRANSMIT (BST)		A3
*RTNMCHB	MCHTR-MCHB;TRANSMIT		B1
*RTNMMH	MCHTR-MMH;TRANSMIT		55
*RTNMML	MCHTR-MML;TRANSMIT		95
*RTNSS	MCHTR-SS;TRANSMIT (BST)		93
SPCLK	STOP THE CLOCK (BST)	CC=OFF-LINE	17
STCLK	START THE CLOCK (BST)		C3
SWITCH	HARDWARE INITIALIZATION (BST)	CC=OFF-LINE	0F
TOGCLK	TOGGLE CLOCK INPUT (BST)	CLK STOPPED	27

* These commands require a zero data field format in the transmitted message.

+ BST (bit stream timing) signifies clock-independent commands.

<u>Command Descriptions</u>

CLER — Clears the other CCs error register (ER).

CLMSR — Clears the other CC maintenance state register (MSR).

CLPT — Causes the other CC program timer (PT) to be cleared. This command is performed independent of any normal CC control or data paths. If this command is attempted in an on-line processor, the command is not executed and an error signal is generated.

CLTTO — Clears the timer time-out register (TTO) which clears the back up timing counter (BTC) after it has been set for a fixed period of time. In order to keep the other CC program timer (PT) blocked, the TTO must be cleared periodically.

DISA — Causes the first disable signal to be sent to the other CC disable I/O circuit.

DISB — Causes the second disable signal to be sent to the other CC disable I/O circuit.

INTCLK — Initializes the clock to phase 3. The running or stopped status of the clock is not affected by this command.

TABLE R (Contd)

MAINTENANCE CHANNEL COMMANDS

LDMAR — Resets the stop flip-flop, loads the other CC micro-address register (MAR) with the low 12 bits of the data field (MCHTR) and sets the freeze flip-flop, freezing the MAR contents. The controlled CC must be stopped prior to this command execution and be off-line.

LDMCHB — Causes the 20-bit data field and 2 parity bits of the MCHTR to be loaded into the other CC maintenance channel buffer register (MCHB).

LDMIRH — Causes the data field to be gated to the other CC MIR for one microcycle. This command is executed only if the other CC is stopped or frozen.

LDMIRL — Causes the data field to be gated to the microinstruction register (MIR) low 16 bits and the resultant to and from control fields to be active for a single cycle. This command is executed only if the other is stopped or frozen.

MSTART — Reset the freeze flip-flop allowing the other CC to start processing microinstructions at the address frozen in the MAR. This command is performed only if the CC flip-flop is cleared (off-line).

MSTOP — Sets the stop flip-flop which jams the MAR to a maintenance address resulting in all zeros being read out of the micromemory. This command is performed only if the CC flip-flop is cleared (off-line).

RTNER — Returns error register (ER) from the other CC. The ER is gated to the MCHTR and transmitted back, independent of any normal CC control or data paths.

RTNMB — Returns miscellaneous bits (MB) from the other CC. The MB is gated to the MCHTR and transmitted back independent of any normal CC control or data paths.

RTNMCHB — Causes the contents of the MCHB in the other CC maintenance channel to be gated to its MCHTR and transmitted back.

RTNMMH — Causes the high 16 bits of the micromemory word currently addressed by the microaddress register (MAR) to be gated to the MCMTR and transmitted back.

RTNMML — Causes the low 16 bits of the micromemory word currently addressed by the microaddress register to be gated to the MCHTR and transmitted back.

RTNSS — Returns the system status register (SS) from the other CC. The SS is gated to the MCHTR and transmitted back independent of any normal CC control or data paths.

SPCLK — Stops the clock with phase 3 active. CC must be off-line.

STCLK — Starts the clock at the phase 3 to phase 0 transition.

SWITCH — Performs a hardware initialization of the other CC in order to switch it on-line. This command is performed only if the CC flip-flop is cleared (off-line). If the CC flip-flop is set (on-line),then an interrupt is generated in the on-line CC.

TOGCLK — Causes the other CC clock to be advanced one-half phase. This command is used when the other CC clock has been stopped to manually step it through all of its states.

Notes:

1. MCH channel codes are normally represented as 4-out-of-8 codes, the extra 1 being the start bit, but this bit need not be decoded as it is common to all codes.
2. These MCH outputs cause interrupts when decoded in an on-line CC and are inhibited in the off-line CC.
3. BST=bit-stream timed.
4. These commands are only executed if CC is off-line.
5. These commands are only executed if CC is stopped or frozen.

TABLE S

MCH STATUS REGISTER

BITS		STATUS
RTN	ERROR	
1	0	Normal Completion
0	0	No Reply Received
0	1	Simultaneous Transmission
1	1	Normal Completion With a Double Output Error Detected in the Slave Controller.

TABLE T

CONTROL PANEL SWITCH REGISTERS

SWITCH REGISTER 1 (SWR1)					
DISPLAY BUFFER BIT	TERM	NETNAME		(PIN NO.)	FUNCTION
00	MPK0	MPK0	(10-23) FA1027	(216)	MAN PTY Key
01	STEOJ0	STEOJ0		(004)	STEP Key
02	HALTK0	HALTK0		(304)	HALT Key
03	BAEXK0	BAEXK0		(315)	BASIC or EXTENDED Key
04	DISRK0	DISRK0		(312)	DISPLAY REG Key
05	LOADK0	LOADK0		(213)	LOAD Key
06	RESCK0	RESB0		(109)	RESET CKT (S) FF OP
07	TMKOA	TMKOA		(210)	TST MODE REVERSAL Key
08	COAKT0	COAKT0		(110)	DATA MATCH Key
09	CADRK0	CADRK0		(011)	ADRS MATCH Key
10	MANK0	MANK0	(10-22) FA1028	(105)	MANUAL KEY
11					
12	DISMK0	DISMK0		(111)	DISPLAY MEM Key
13	STOK0	STOK0		(212)	STORE IN MEM Key
14	IADRK0	IADRK0		(112)	INCREMENT ADRS Key
15	HILOK0	HILOK0		(205)	HIGH-LOW MEM Key
16					
17					
18					
19					
P_L					
P_H					
	RS1K0	RS1K0	(10-23) FA1027	(003)	REG SELECT Key
	RS2K0	RS2K0		(303)	REG SELECT Key
	RS4K0	RS4K0		(104)	REG SELECT Key
	RS8K0	RS8K0		(012)	REG SELECT Key
	SGK0	SGK0		(112)	GENERAL or SPECIAL
	PWRKBB	PWKO		(314)	PWR Key (BB CONT) ON-OFF
			(10-22) FA1028		
	IK00	IK000	(10-23) FA1027	(100)	DATA INPUT Key
	IK10	IK010		(103)	DATA INPUT Key
	IK20	IK020		(204)	DATA INPUT Key
	IK30	IK030		(311)	DATA INPUT Key
	IK40	IK040		(212)	DATA INPUT Key

TABLE T (Cont)

CONTROL PANEL SWITCH REGISTERS

SWITCH REGISTER 3 (SWR3)					
DISPLAY BUFFER BIT	TERM	NETNAME		(PIN NO.)	FUNCTION
	IK50	IK050	(10-23) FA1027	(013)	DATA INPUT Key
	IK60	IK060		(113)	DATA INPUT Key
	IK70	IK070		(309)	DATA INPUT Key
	IK80	IK080		(010)	DATA INPUT Key
	IK90	IK090		(310)	DATA INPUT Key
	IK100	IK100	(10-22) FA1028	(011)	DATA INPUT Key
	IK110	IK110		(211)	DATA INPUT Key
	IK120	IK120		(311)	DATA INPUT Key
	IK130	IK130		(012)	DATA INPUT Key
	IK140	IK140		(016)	DATA INPUT Key
	IK150	IK150		(216)	DATA INPUT Key
	IK160	IK160		(116)	DATA INPUT Key
	IK170	IK170		(316)	DATA INPUT Key
	IK180	IK180		(017)	DATA INPUT Key
	IK190	IK190		(217)	DATA INPUT Key
	IKPL0	IKPL0		(117)	PANEL P_L INPUT DATA Key
	IKPH0	IKPH0		(317)	PANEL P_H INPUT DATA Key