SPACE DIVISION
NORTH AMERICAN ROCKWELL CORPORATION

MANPOWER DEVELOPMENT

[redacted]

APOLLO GUIDANCE & NAVAGATION

SYSTEM FAMILIARIZATION

PART II

COURSE NUMBER 733-GS

SPACE DIVISION
NORTH AMERICAN ROCKWELL CORPORATION

MANPOWER DEVELOPMENT

APOLLO GUIDANCE & NAVAGATION

SYSTEM FAMILIARIZATION

PART II

COURSE NUMBER 733-GS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (contd.)

SECTION IX

THE COMPUTER SUBSYSTEM

INTRODUCTION

This section of the study guide briefly describes the CSS equipment to a functional block level. A brief description of the CSS organization and functional operation is also presented.

9.1 CSS PURPOSE

The computer subsystem is the control and computational center of the PGNCS. It performs the following functions:

a. Solves the guidance and navigation problems for all mission phases including Saturn Guidance take over capability during the boost phase.

b. Provides control information to the PGNCS, as well as other spacecraft systems.

c. Displays pertinent information to the astronaut and the ground when requested.

d. Provides a means by which the astronaut or ground control can directly communicate with the PGNCS.

e. Provides direct on-off control for the reaction control jets, and service propulsion engines.

f. Monitors its own operation and other PGNCS operations.

9.2 CSS EQUIPMENT

The CSS consists of the Command Module Computer (CMC) and two identical display and keyboards (DSKY's) (see figure 9-1). The CMC is located in the lower center of the lower display and control panel below the power and servo assembly. One DSKY is located on the lower display and control panel and is called the navigation panel DSKY. The other DSKY is located on the main display and control panel and is called the main panel DSKY.

9.2.1 CMC PHYSICAL DESCRIPTION. The CMC measures approximately 24 x 12.5 x 6 inches and weighs approximately 60 pounds. It consumes 10 watts of power during standby operation and 100 watts of power during normal operation. The CMC is mounted on a cold-plate base which is cooled by a water glycol solution flowing through it.

The CMC consists of a logic tray assembly and a memory tray assembly which are mounted back to back (figures 9-2 and 9-3) . A tray assembly consists of a tray with the modules and connectors mounted on it. The modules contain the CMC's logic circuitry. The connectors provide the CMC with a hardware interface.

9.2.2 CMC USE. The CMC is a core memory, digital computer with two types of memory: fixed and erasable. The fixed memory permanently stores navigation tables, trajectory parameters, programs and constants. The erasable memory stores intermediate information.

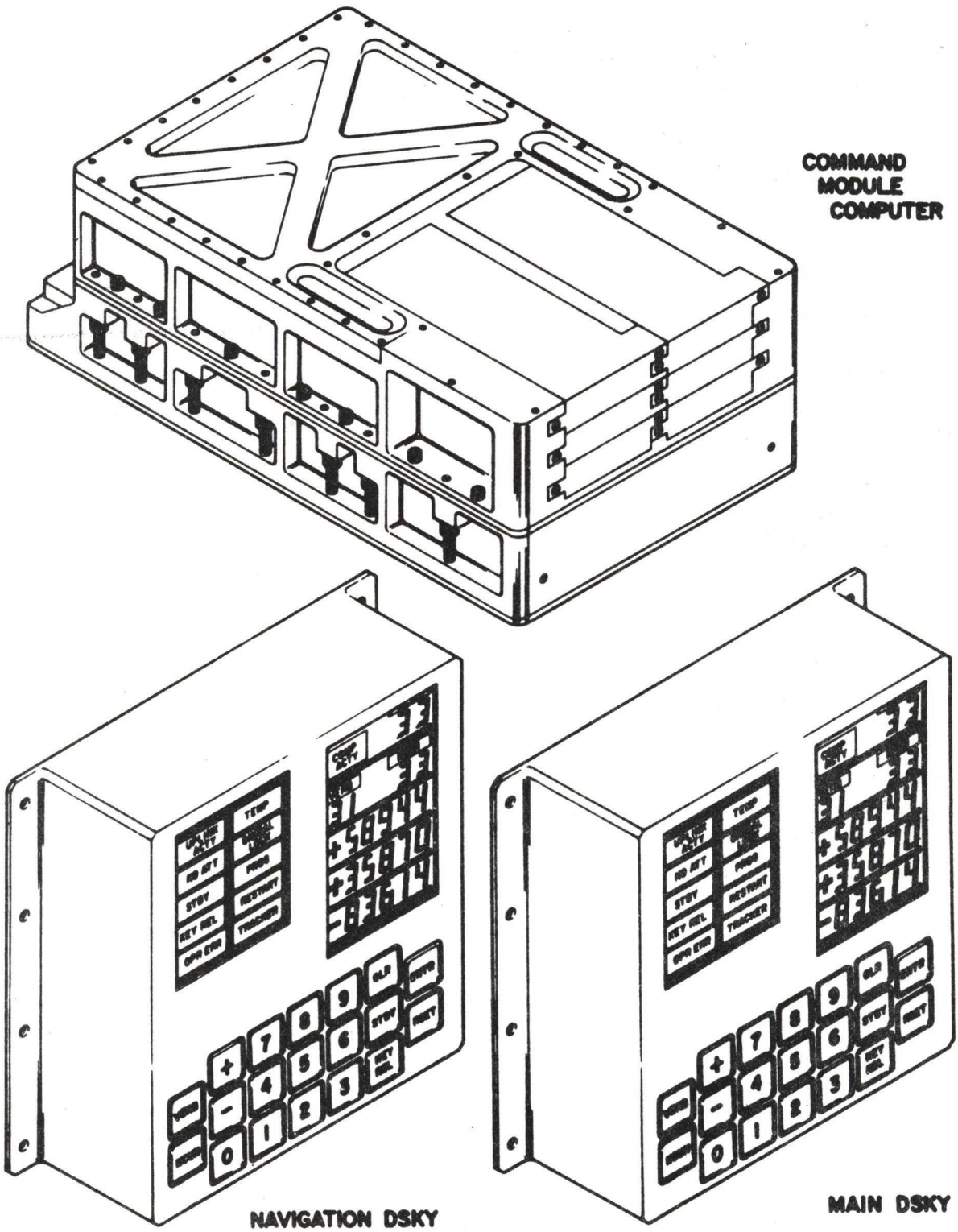


Figure 10-1. CSS Components

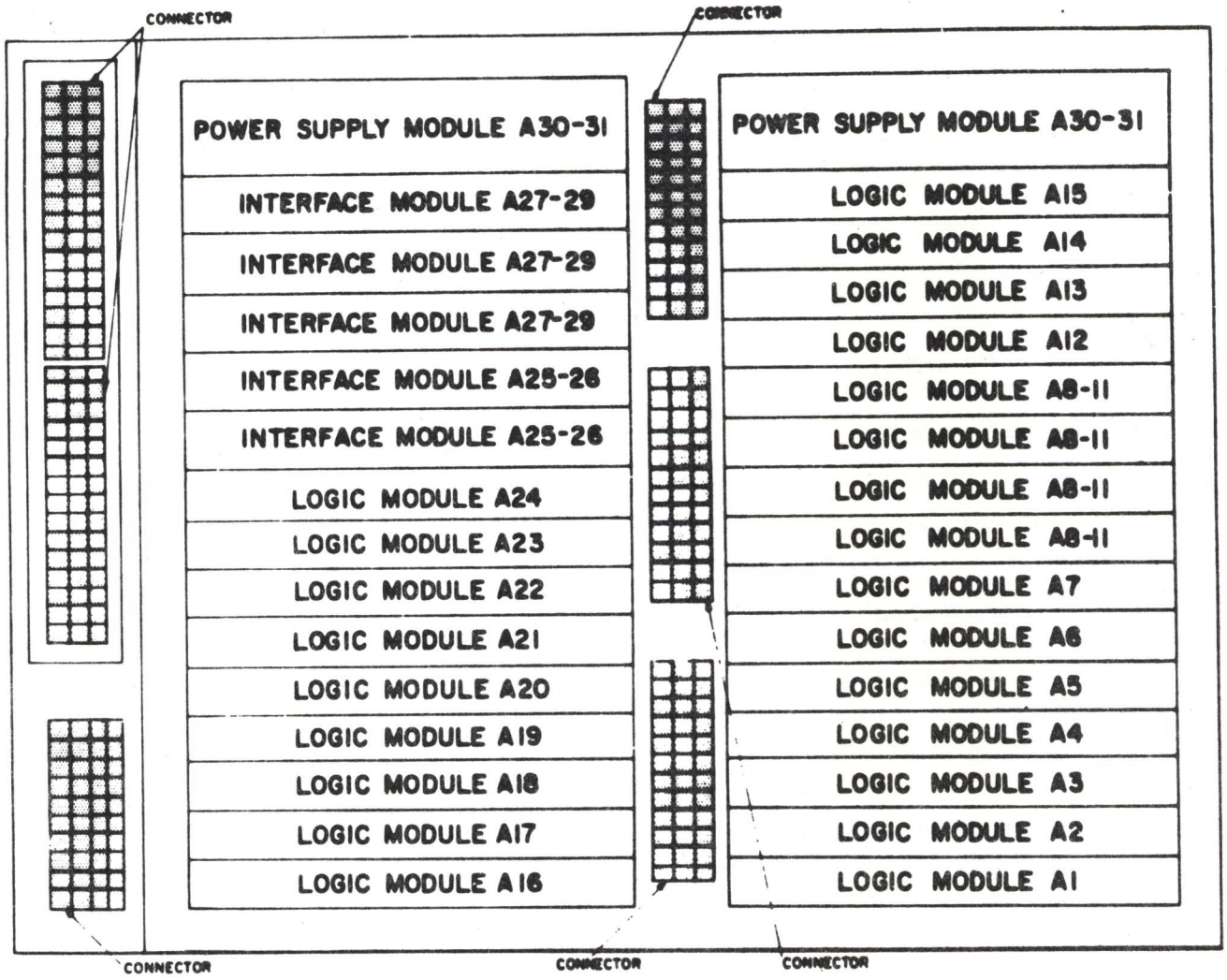


Figure 9-2. Logic Tray A

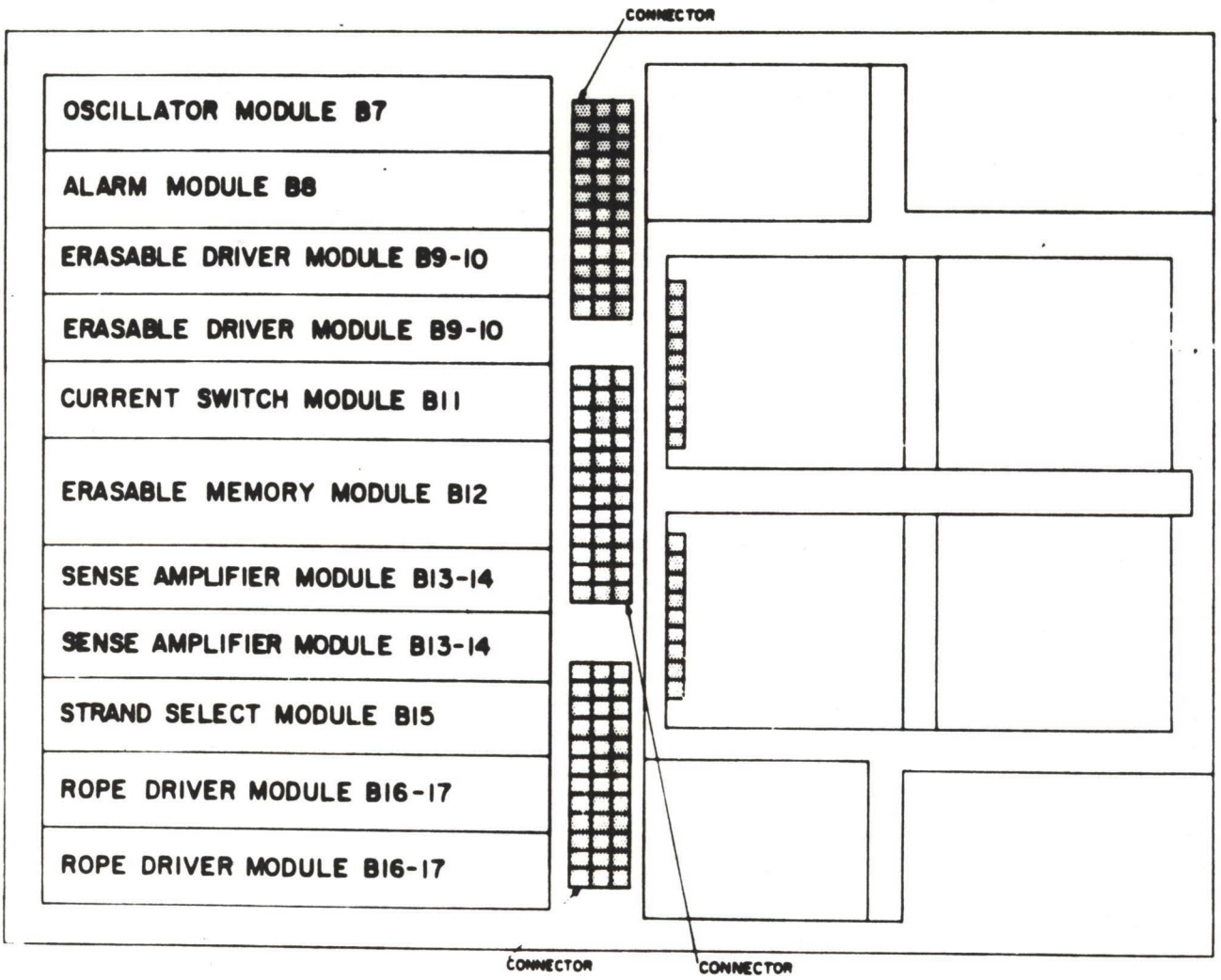


Figure 9-3. Memory Tray B

The CMC processes data and issues discrete control signals, both for the PGNCS and the other spacecraft systems. It is a control computer with many of the features of a general purpose computer. As a control computer, the CMC aligns the stable platform of the inertial measurement unit (IMU) in the inertial subsystem, positions the optical unit in the optical subsystem and issues control commands to the spacecraft. As a general purpose computer, the CMC solves guidance problems required for the spacecraft mission. In addition, the CMC monitors the operation of the PGNCS and and other spacecraft systems.

The CMC stores data pertinent to the flight profile that the spacecraft must assume in order to complete its mission. This data, consisting of position, velocity and trajectory information, is used by the CMC to solve the various flight equations. The results of various equations can be used to determine the required magnitude and direction of thrust required. Corrections to be made are established by the CMC. The spacecraft engines are turned on at the correct time, and steering signals are controlled by the CMC to reorient the spacecraft to a new trajectory, if required. The inertial subsystem senses acceleration and supplies velocity changes to the CMC for calculating the total velocity. Drive signals are supplied from the CMC to coupling data unit (CDU) and stabilization gyros in the inertial subsystem to align the gimbal angles in the IMU Error signals are also supplied to the CDU to provide steering capabilities for the spacecraft. CDU position signals are fed to the CMC to indicate changes in gimbal angles, which are used by the CMC to keep cognizant of the gimbal positions. The CMC receives mode indications and angular information from the optical subsystem during optical sightings. This information is used by the CMC to calculate present position and orientation and is used to refine trajectory information. Optical subsystem components can also be positioned by drive signals supplied from the CMC.

9.2.3 DSKY PHYSICAL DESCRIPTION. The DSKY measures approximately 8 x 8 x 7 inches and weighs about 17.5 pounds. Mounted on the back of the DSKY are six interchangeable driver modules, a 91 pin connector, a filler valve, and a power supply. The driver modules are used in the switching of the data displayed on the DSKY. The connector provides the interface between the DSKY and the CMC. The filler valve is used to pressurize the DSKY to one atmosphere. The power supply module supplies the voltage to light the DSKY indicators.

9.2.4 DSKY USE. The navigation DSKY located on the lower D & C panel and the DSKY located on the main D & C panel provides a two-way communications link between the astronaut and the CMC (see figure 9-1). Through this communications link, the following functions can be performed:

a. Loading of data into the CMC.

b. Display of data from the CMC and data loaded into the CMC.

c. Monitoring of data in the CMC.

d. Display of the CMC modes of operation.

e. System control by the initiation of subsystem and system testing and control of the system's major modes of operation.

f. Requests by the CMC to the system operator to perform actions.

The DSKY's consists of a keyboard, display panel, condition indicators, a power supply and a relay package (see figure 10-1). The keyboard provides the astronaut with the capability of inserting data into the CMC and initiating CMC operations. Through the keyboard, the astronaut can also control the ISS moding and some OSS moding. The DSKY display panel provides a visual indication of data being loaded into the CMC, the CMC activity and CMC program. The display panel also provides the CMC with a means of displaying or requesting data. The condition indicators display PGNCS status.

9.3 CSS ORGANIZATION

Figure 9-4 is a gross block diagram of the CSS. This diagram should be referenced while reading paragraphs 9.3.1 and 9.3.2.

9.3.1 CMC ORGANIZATION. The CMC is functionally divided into seven blocks:

1. Timer.
2. Sequence generator.
3. Central processor.
4. Memory.
5. Priority control.
6. Input - output.
7. Power.

9.3.1.1 Timer. The timer generates all the necessary synchronization pulses to insure a logical data flow from one area to another within the CMC. It also generates timing waveforms which are used by (1) the CMC's alarm circuitry and (2) other areas of the spacecraft for control and synchronization purposes.

9.3.1.2 Sequence Generator. The sequence generator directs the execution of machine instructions. It does this by generating control pulses which logically sequence data throughout the CMC. The control pulses are formed by combining the order code of an instruction word with synchronization pulses from the timer.

9.3.1.3 Central Processor. The central processor performs all arithmetic operations required of the CMC, buffers all information coming from and going to memory, checks for correct parity on all words coming from memory and generates a parity bit for all words written into memory.

9.3.1.4 Memory. Memory provides the storage for the CMC and is divided into two sections: erasable memory and fixed memory. Erasable memory can be written into or read from; its readout is destructive. Fixed memory cannot be written into and its readout is nondestructive.

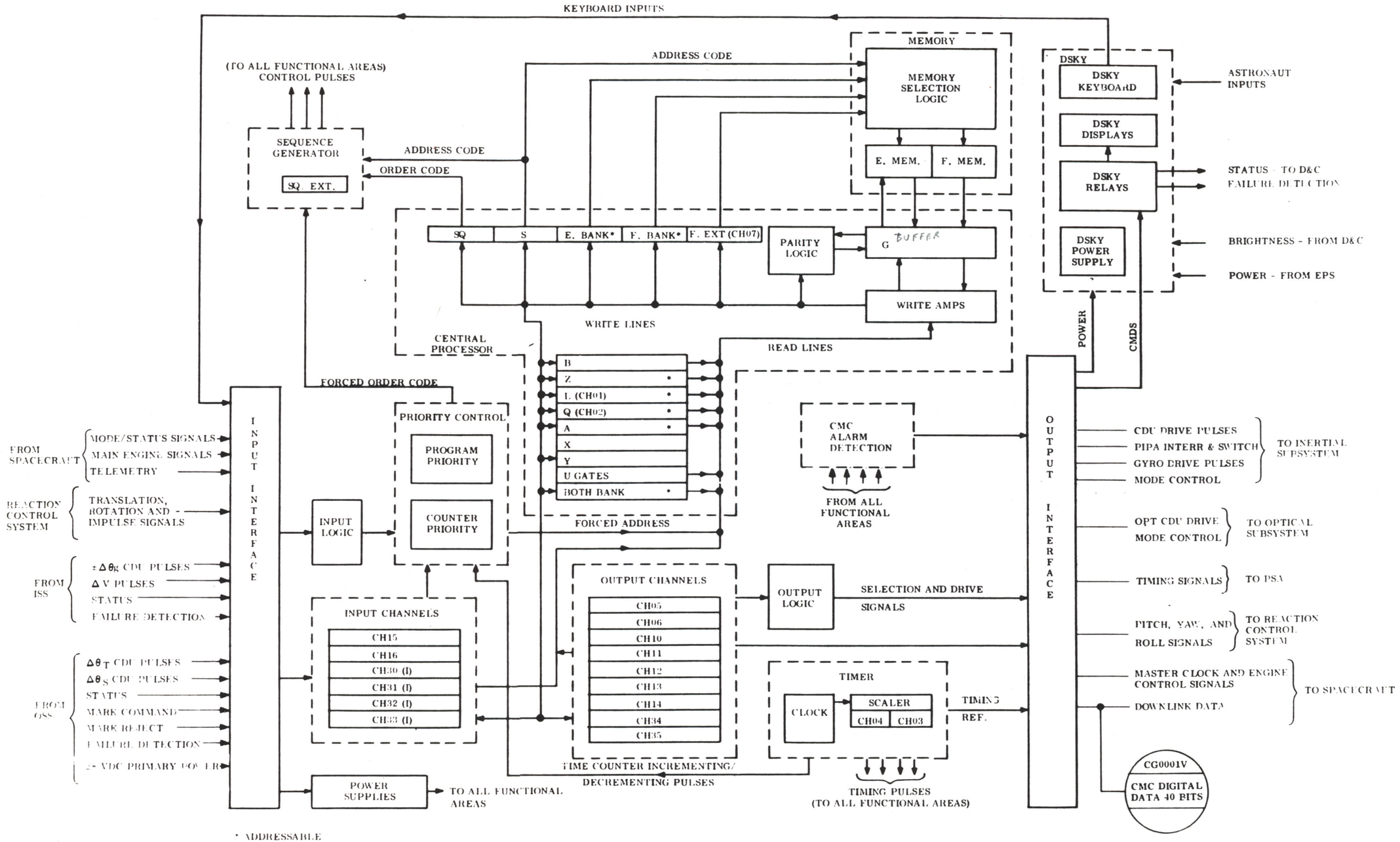


Figure 9-4. CSS Functional Flow

9.3.1.5 Priority Control. Priority control establishes a processing priority of operations which must be performed by the CMC. These operations are a result of conditions which occur both internally and externally to the CMC. Priority control consists of counter priority control and interrupt priority control. Counter priority control initiates actions which update counters in erasable memory. Interrupt priority control transfers control of the CMC to one of several interrupt subroutines stored in fixed memory.

9.3.1.6 Input - Output. The input - output section routes and conditions signals between the CMC and other areas of the spacecraft.

9.3.1.7 Power. This section provides voltage levels necessary for the proper operation of the CMC.

9.3.2 DSKY ORGANIZATION. The DSKY is functionally divided into three blocks:

1. Keyboard.
2. Display indicators.
3. Condition indicators.

9.3.2.1 Keyboard. The keyboard consists of ten numerical keys (pushbuttons), two sign keys, and seven instruction keys. The keys are used to enter data into the computer, place the computer in standby mode or remove it from standby, or release the DSKY so the computer can display data to the astronaut.

9.3.2.2 Display Indicators. The DSKY contains 21 digit display indicators and three sign display indicators. The program, noun, and verb displays each contain two digit positions. Each of the three data displays contains a sign display and five digit display indicators.

9.3.2.3 Condition Indicators. The DSKY contains 10 condition indicators which indicate computer mode of operation and some computer problems and PGNCS malfunctions.

9.4 COMPUTER SUBSYSTEM OPERATIONS

The CSS interfaces with the ISS, OSS, astronaut and certain spacecraft systems. These interfaces are described in the paragraphs below.

9.4.1 CSS/ISS OPERATIONS. The CSS receives changes in IMU gimbal angles and changes in velocity from the ISS. These changes are accumulated in the CMC. The IMU gimbal positions are used by the CMC to monitor platform orientation. The velocity changes are used by the CMC to calculate the velocity and, by integration, the position of the spacecraft. The CMC also commands ISS moding and monitors certain subsystem components for failures. The components monitored are:

a. CDU inertial channels
b. Accelerometer loops
c. Stabilization loops
d. Power supplies

Inertial subsystem moding, with the exception of IMU cage, is controlled by the computer by the generation of commands to the inertial CDU. The discrete commands issued are CDU Zero, Coarse Align, Error Counter Enable, And SIVB takeover. The generation of the discretes is done automatically through a computer program or by astronaut command through operation of either DSKY. During the coarse align mode of operation the computer provides digital inputs to the Coupling Data Unit. The CDU converts the digital command to an analog signal which is used to reposition the gimbals. During Fine Align the computer provides torquing pulses to the Apollo II IRIG torque ducosyns for fine alignment of the stable platform. The computer also supplies frequency references to the CDU and to the inertial subsystem power supplies.

9.4.2 CSS/OSS OPERATIONS. The optical subsystem is used in conjunction with the computer to determine spacecraft position, velocity, attitude and alignment angles for the inertial subsystem. The primary interface between the optical units and the CSS is via the CDU. Upon completion of a navigation sighting, the sextant trunnion and shaft angles are coupled, via the CDU, to the computer. The computer program uses these angles and time to determine spacecraft position, velocity and attitude. In the event that an IMU alignment is to follow the optical sighting, the program determines the angular corrections necessary to align the IMU to the inertial reference frame.

Under certain combination of OSS control switches on the G&N Indicator Control Panel it is possible for the computer to control the moding and the positioning of the optics. The astronaut must complete the navigation sighting after the computer positions the optics by tracking and marking on the celestial objects.

The computer monitors the optical subsystem modes of operation to determine what computer operation shall be performed. The mode Zero in the OSS, when manually selected, causes the optics angle counters in the CMC to be zeroed. The computer, in addition to monitoring or controlling the optical subsystem operation, provides timing pulses as references to the OSS power supplies.

9.4.3 CSS/SPACECRAFT. The functions of the PGNCS is to provide guidance, navigation and control for the Apollo spacecraft. Because of this increase in scope, over previous Apollo G&N Systems, the interface with the spacecraft has expanded and now includes the following items:

9.4.3.1 CSS/Saturn Instrumentation Unit. The CSS receives indications of Saturn lift-off and guidance release from the Instrumentation Unit. The receipt of the guidance release discrete causes the CMC to discontinue the gyro-compassing routine and allows the stable member to become inertially referenced. The lift-off signal causes the CMC to begin computing velocity, position and attitude of the spacecraft. The computer compares these data with a reference trajectory, the difference between the two are utilized to determine the need for PGNCS takeover of Saturn guidance. In the event that the crew selects the PGNCS to control Saturn guidance, the computer calculates steering errors, which are converted from digital to analog by the CDU, issues start injection sequence, and Saturn engine on/off commands all of which are coupled to the Saturn IU.

9.4.3.2 CSS/Central Timing Equipment (CTE). The Command Module Computer provides a 1.024 MC frequency reference to the CTE. This enables all command module timing to be referenced to a single time standard.

9.4.3.3 CSS/Communications and Instrumentation System (C and IS). The CSS receives uplink telemetry data and downlink telemetry synchronization pulses from the C and IS. The uplink telemetry data is in the form of five bit codes which are identical to the key codes provided by the DSKY's. The operation of the CSS then can be controlled from the ground station via the uplink telemetry. There are two methods of inserting the uplink telemetry data into the CMC One method is to insert all the uplink data required and thereby exert complete control of the CSS from the ground station. The other method is to insert all the uplink data except the final command for the CMC to process the data. The astronaut then has the option of using the uplink data or not. This data is displayed to the astronaut on the DSKY display panels for his inspection. The downlink telemetry sync pulses are used to regulate the gating of downlink telemetry data to the C and IS. The C and IS receives downlink telemetry data from the CSS. This data can be transmitted at 10 words per second or at 50 words per second (1.6 K bits per second or 51.2 K bits per second, respectively).

The ground station uses part of this downlink data to duplicate the displays and DSKY inputs that have occurred in the spacecraft. The ground station can send commands and data to the CMC in a format similar to the DSKY inputs.

9.4.3.4 CSS/Mission Sequencer. The CSS receives an indication of command module-service module separation and of SIVB separation from the mission sequencer.

9.4.3.5 CSS/Spacecraft Controls. The CSS interface with the spacecraft controls is primarily on-off switching of DC voltages. The following switches on the Main D and C panel have direct effects on the operation of the PGNCS.

a. S/C control of Saturn - Initiates a computer program for boost guidance.

b. ΔV CG Switch - Allows the computer to compensate for the change in the spacecraft center of gravity when the LM is attached. This information is necessary for midcourse correction maneuvers.

c. SC Control - source switch - Selects computer control of the propulsion system.

d. SPS Thrust - Indicates to the CMC that the SPS engine start checklist has been completed and the SPS is ready for thrusting.

e. SC Control - CMC switch - A three position switch - Hold, Free and Auto indicating to the CMC what type of attitude control it should exercise.

The CMC also has inputs from the rotation and translation controls for pilot initiated attitude and positional maneuvers. For attitude control during optical sightings, the navigator utilizes the minimum impulse controller which also has direct inputs to the CMC. For spacecraft maneuvers, whether automatically or manually commanded, the computer issues on/off discretes to the 16 RSC jets on the service module or the 12 RCS jets on the command module. During PGNCS controlled thrusting maneuvers using the SPS, the computer calculates steering signals and issues SPS engine on/off commands.

9.4.4 CSS/ASTRONAUT. The CSS interface with the astronaut is through the DSKY's. The operator of the DSKY can communicate with the CMC by the depression of a sequence of keys on the DSKY keyboard. Each depression of a key inserts a five bit code into the CMC. The CMC responds by returning a code to the DSKY, which controls the display on a particular display panel, or by initiating a CMC operation. The CMC is also capable of initiating a display of information or a request for some action to the operator under its own initiative.

The basic communication language used in the interchange of information is a pair of words known as the VERB and NOUN. Each of these words is represented by a two digit octal number. The VERB specifies that an action of some sort is to be performed while the NOUN, used in conjunction with the VERB, specifies on what the action is to be performed.

9.5 CMC INFORMATION FLOW

The main functions of the CMC are implemented through the execution of programs stored in memory. Programs are written in basic instructions. The order code defines data flow within the CMC. The relevant address selects data that is to be used for computations. The order code of each instruction is entered into the sequence generator which controls data flow and produces a different sequence of control pulses for each instruction. Each instruction is followed by another instruction. In order to specify the sequence in which consecutive instructions are to be executed, the instructions are normally stored in successive memory locations. By adding +1 to the address of an instruction being executed, the address of the next instruction is obtained. Execution of an instruction is complete when the order code of the next instruction is transferred to the sequence generator and its relevant address is in the central processor.

The central processor consists primarily of flip-flop registers. It performs arithmetic operations and data manipulations on information accepted from memory, the input registers and priority control. Arithmetic operations are performed using the binary ones complement numbering system. All operations within the central processor are performed under control of pulses generated by the sequence generator.

The CMC has provisions for handling certain priority programs. Before a priority program can be executed, the current program must be interrupted; however, certain information about the current program must be preserved. This information includes the program counter contents and any intermediate results contained in the central processor. The program counter is a register in the central processor which contains the address of the next instruction to be performed. The priority control produces an interrupt request signal which is sent to the sequence generator. This signal causes the execution of an instruction that transfers the current contents of the program counter and any intermediate results to memory. In addition, the control pulses transfer the priority program address to the central processor and then to memory. As a result, the first instruction word of the priority program is entered into the central processor from memory and the execution of the priority program is begun. The last instruction of each priority program restores the CMC to normal operation if no other interrupt request is present. This is done by transferring the previous program counter and intermediate results from their storage locations in memory back to the central processor.

Certain data pertaining to the flight of the spacecraft is used to solve the guidance and navigation problems required for the Apollo Mission. This data which includes real time, acceleration and IMU gimbal angles (via the CDU) is stored in the erasable memory counters which are updated as soon as new data becomes available. An incrementing process which changes the contents of the counters is implemented by priority control between the execution of sub-instructions. Data inputs to counter priority control are called incremental pulses. Each incremental pulse produces a counter address and a priority request signal which is sent to the sequence generator where it functions as an order code. The control pulses produced by the sequence generator transfer the counter address to memory. In addition, the control pulses enter the contents of the addressed counter into the central processor. The contents of the counter are then incremented or decremented.

Real time plays a major role in solving guidance and navigation problems. Real time is maintained within the CMC by the main time counter. The main time counter consists of two erasable memory counters. Incremental pulses are produced in the timer and sent to priority control for incrementing the main time counter every 10 milliseconds.

Continuous drive pulses originate in the CMC's timer. Rate signals, which are bursts of these drive pulses, are used to drive the inertial CDU channels, the gyros and the optical CDU channels. Rate signals are also used for controlling the attitude of the spacecraft. The number of pulses in each burst and occurrence of each burst are controlled by the program. The destination of the various rate signals, as well as the type of rate signals, is also selected by the program. The continuous drive signals are also sent to other areas of the spacecraft where they are used for synchronization and control purposes.

The uplink word from the spacecraft telemetry system is supplied to the CMC as incremental pulse inputs to priority control. As these words are received, priority control produces the address of the uplink counter in memory and requests the sequence generator to execute the instructions which perform the serial-to-parallel conversion of the uplink word. When the serial-to-parallel conversion is complete, the parallel word is transferred to a storage location in memory by the UPRUPT subroutine.

With the DSKY keyboard, the astronaut can load information into the CMC, receive and display information contained in the CMC and initiate any program stored in memory. A keycode is assigned to each keyboard pushbutton. When a keyboard pushbutton on either DSKY is depressed, the keycode is produced and sent to the CMC where it produces the address of a priority program (KEYRUPT) stored in memory and a priority request signal which is sent to the sequence generator. The priority request signal functions as an order code and initiates an instruction for interrupting the program in progress and executing the priority program stored in memory. A function of this program is to decode and process the keycode. A number of keycodes are required to specify an address or a data word. The program initiated by a keycode converts the information from the DSKY keyboard to a coded display format which is transferred to the display portion of the DSKY. The display informs the astronaut that the keycode was received, decoded and properly processed by the CMC.

Word formats, addition schemes, logic implementation, and hardware registers are described in preparation for a detailed description of computer organization.

CMC characteristics are summarized in table 9-1.

9.6 WORD FORMATS

All words are 16 bits long in the computer. There are two basic types of words: data words and instruction words. The format of the words depend on where the words are located. Figure 10-5 shows the word formats in memory (storage) and in the central processor registers.

In memory, data words contain a parity bit, fourteen magnitude bits and a sign bit. A binary "1" in the sign bit indicates a negative number, a binary "0" in the sign bit indicates a positive number. When located in the central processor, bits 1 through 14 are the magnitude bits, bit 15 is the "uncorrected sign" bit, and bit 16 is the sign as before. The "uncorrected sign" bit is used to enable an overflow detection without destroying the sign bit when two numbers are manipulated.

Parity bits are only included in words which are stored in memory.

An instruction word in memory contains a 12 bit address code and a 3 bit order code. Bits 10 through 12 are sometimes used to extend the order code when the work is transferred to the SQ register. The address code normally calls out the location of a word in memory or the central processor. The order code represents an operation which is to be performed on the data whose location is defined by the address code.

9-7 ADDITION SCHEMES

The CMC performs angular data addition and non-angular data addition. The processes involved with these manipulations differ slightly.

9.7.1 NON-ANGULAR DATA ADDITION. The CMC uses the binary ONE's complement numbering system. In this system, a negative number is the complement of the corresponding positive number. Assume that the CMC uses five bit data words. Figure 10-6 shows how +7 through -7 would be represented by words in memory and the central processor.

Table 9-1. CMC Characteristics

Characteristics	Description
Computer type	Automatic, electronic, digital, general purpose and control
Internal transfer	Parallel
Memory	Random access
Erasable	Coincident current core, capacity = 2048 words
Fixed	Core-rope, capacity = 36,864 words
Word length	Sixteen bits
Number system	Binary ONE's complement
Circuitry type	Flat pack NOR micrologic
Machine instructions	56 total
Regular	42 total
Involuntary	9 total
Peripheral	5 total
Interrupt options	10 total
Memory cycle time (MCT)	12 μ seconds
Add time	24 μ seconds
Multiply time	48 μ seconds
Number of counters	29 total
Downlink telemetry	Asynchronous to computer timing.
Basic clock oscillator	2.048 mc
CMC power supplies	One +4 volt switching regulator
	One +14 volt switching regulator
Logic	Positive
Parity	Odd

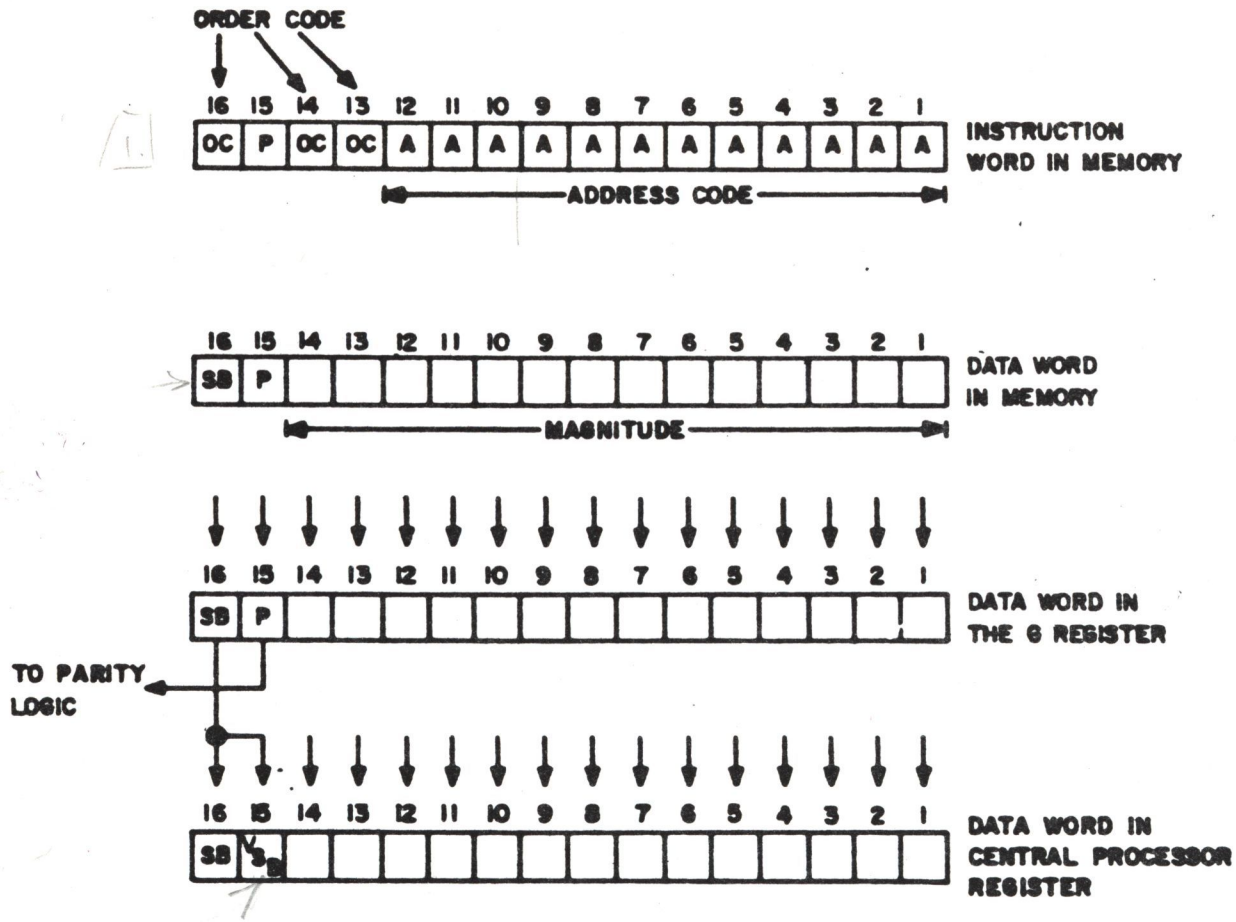


Figure 9-5. Word Formats in Memory and Central Processor

SB	M	M	M	P	
0	1	1	1	0	+7
0	1	1	0	1	+6
0	1	0	1	1	+5
0	1	0	0	0	+4
0	0	1	1	1	+3
0	0	1	0	0	+2
0	0	0	1	0	+1
0	0	0	0	1	+0
1	1	1	1	1	-0
1	1	1	0	0	-1
1	1	0	1	0	-2
1	1	0	0	1	-3
1	0	1	1	0	-4
1	0	1	0	1	-5
1	0	0	1	1	-6
1	0	0	0	0	-7

MEMORY

SB	OUB	M	M	M	
0	0	1	1	1	+7
0	0	1	1	0	+6
0	0	1	0	1	+5
0	0	1	0	0	+4
0	0	0	1	1	+3
0	0	0	1	0	+2
0	0	0	0	1	+1
0	0	0	0	0	+0
1	1	1	1	1	-0
1	1	1	1	0	-1
1	1	1	0	1	-2
1	1	1	0	0	-3
1	1	0	1	1	-4
1	1	0	1	0	-5
1	1	0	0	1	-6
1	1	0	0	0	-7

CENTRAL PROCESSOR

Figure 9-6. Number Representation

The CMC can only add. In order to subtract, the CMC adds the complement of a number. Multiplications are performed by successive additions and shiftings; divisions are performed by successive additions of complements and shiftings.

All adding is performed in the central processor. The CMC's consists of the X, Y and U registers. The X and Y registers are the adder's input registers, i.e., the registers to which the augend and addend are applied. The U register is the adder's output register, i.e., the register at which the sum is found. An overflow occurs during addition when the overflow/underflow bit (OUB) of the U register is a logic 1 and the sign bit (SB) is a logic 0; an underflow occurs when the overflow/underflow bit is a logic 0 and the sign bit is a logic 1.

Figure 9-7 shows the various addition schemes employed by the CMC for manipulating non-angular data. Note in the first case of Example C that the CMC performs (+5) + (-3) and not (+5) - (+3). Example A shows a case of overflow. Example B illustrates end-around-carry and underflow. End-around-carry is a characteristic of the binary ONE's complement numbering system: it is the result of a carry being generated during the addition of the last two binary bits. When this happens, +1 is added to the LSD of the subsum and the sum is then formed. Underflow results when a sum is negative and has a magnitude larger than that expressable with the assigned magnitude bits. End-around-carry and overflow/underflow conditions are mechanized by the CMC. When overflow/underflow occurs, a special counter is incremented or decremented to prevent loss of data.

Summarizing: If the SB and OUB of the sum are identical, then the sign of the sum is determined by the SB, and the magnitude of the sum is determined by the assigned magnitude bits, i.e., the OUB is not taken into consideration; if the SB and OUB of a sum are not identical, then the sign of the sum is determined by the SB, and the magnitude of the sum is determined by the assigned magnitude bits and the OUB. In this case, the OUB becomes the MSD.

One important fact should be noted: the configuration of the magnitude 0 (zero) is not unique but has signs associated with it, i.e., +0 = 00000 and -0 = 11111 in the central processor registers.

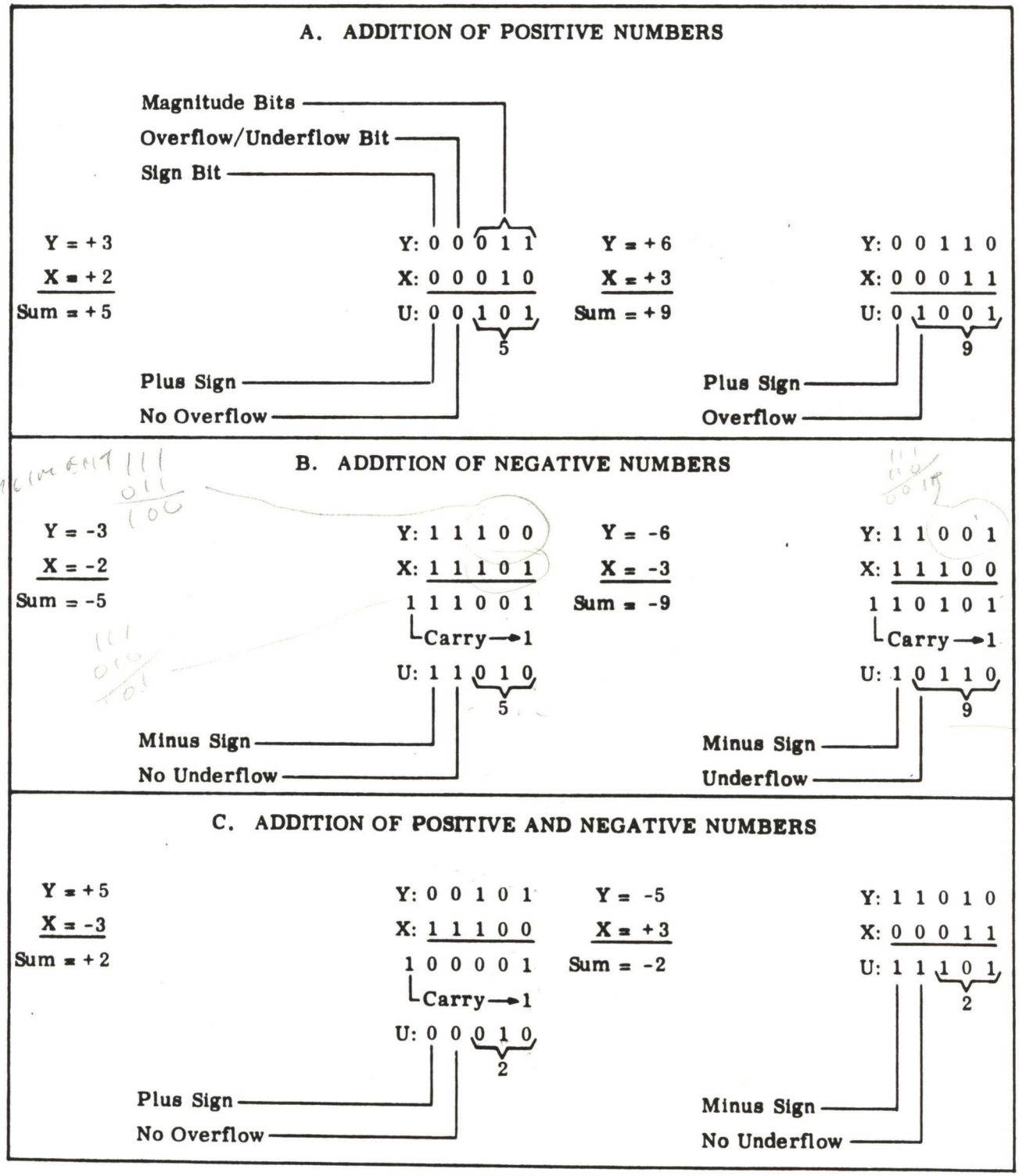


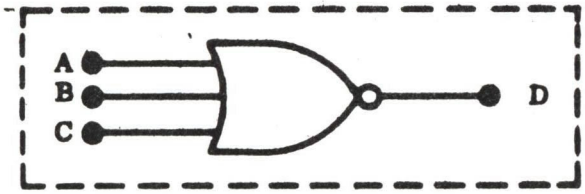
Figure 9-7. Addition Schemes for Non-Angular Data

9.8 LOGIC IMPLEMENTATION

All logic sticks in the CMC are built up from "micrologic" elements. In the CMC, the prime micrologic element used is a three-input NOR gate. The circuitry for two of these elements is diffused onto a single wafer which is contained in a flat pack. Figure 10-10 shows the accepted symbol for a single NOR gate and the associated truth table. The truth table defines the output logic level (1 or 0) for all possible input combinations. The CMC uses positive logic throughout. A logic 1 is 4 volts, a logic 0 is 0 volts. When inputs A, B or C equal a logic one in the NOR circuit, the associated transistor is biased to an on condition . This results in a path for current flow to ground. The output then falls to nearly 0 volts (logic 0) as indicated by the truth table. If all three inputs equal 0, the output D becomes logic 1.

Although the logic circuits of the CMC are comprised entirely of CMC gates, it is convenient to use the standard AND and OR logic blocks to simplifiy the presentation of later material. Figures 10-10 and 10-11 show the OR, AND, FLIP-FLOP and EXCLUSIVE OR circuits. The NOR gates within the dotted lines show the exact mechanization of the circuits in the simplified systems. The reader should compare the truth tables included in the diagrams with the circuits. Note that a one input NOR gate is just an inverter (figure 10-10).

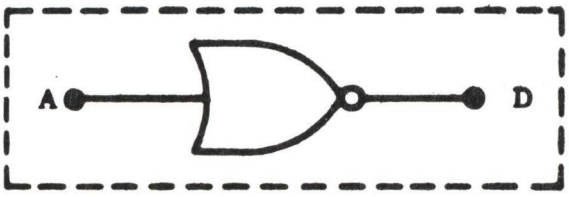
Figure 10-11 shows an EXCLUSIVE OR gate. Its distinguishing feature is that when both inputs are logic one's, the output is a logic zero. A standard OR gate has a logic one output when any or all inputs are logic one's.



"NOR" GATE, SYMBOL

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
D	1	0	0	0	0	0	0	0

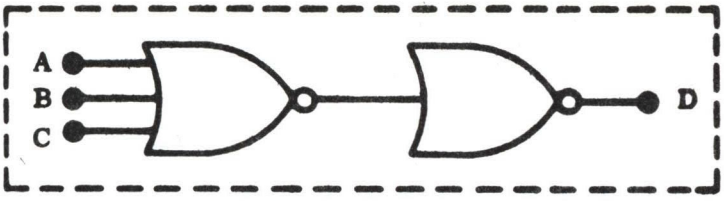
TRUTH TABLE



INVERTER, SYMBOL

A	0	1
D	1	0

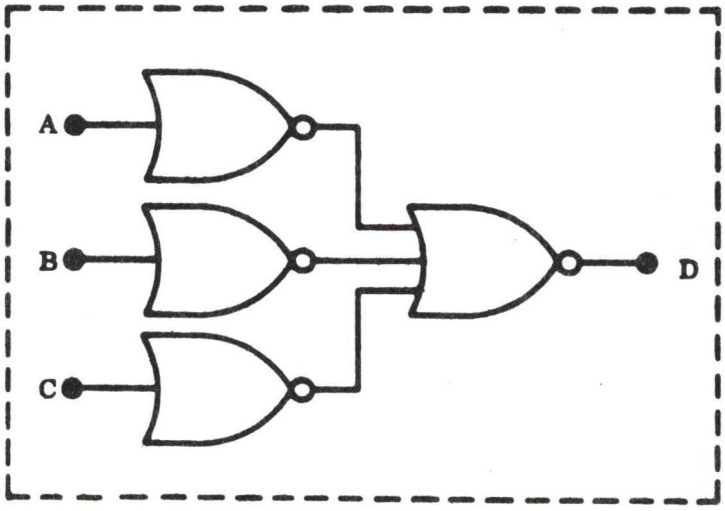
TRUTH TABLE



"OR" GATE, SYMBOL

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
D	0	1	1	1	1	1	1	1

TRUTH TABLE

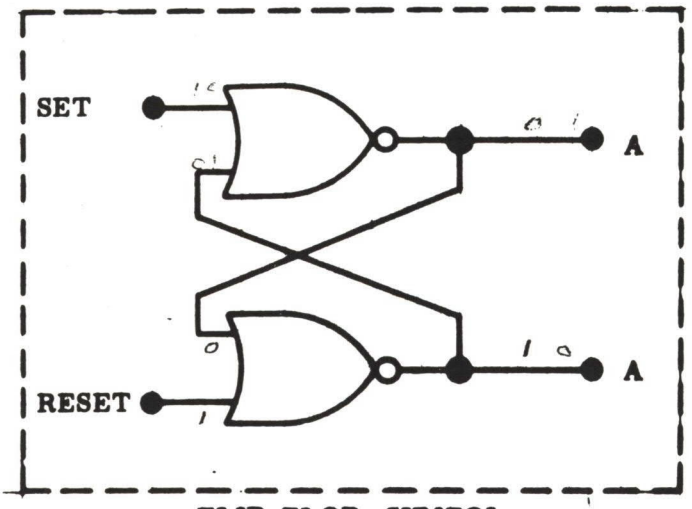


"AND" GATE, SYMBOL

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
D	0	0	0	0	0	0	0	1

TRUTH TABLE

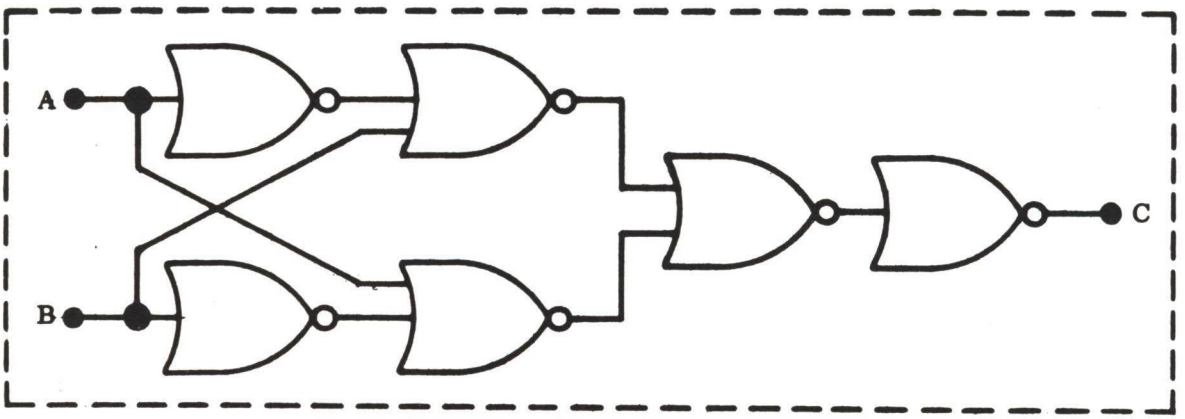
Figure 9-8. Typical Logic Functions



FLIP-FLOP, SYMBOL

	A	A
SET	1	0
RESET	0	1

TRUTH TABLE



"EXCLUSIVE OR", SYMBOL

A	0	0	1	1
B	0	1	0	1
C	0	1	1	0

TRUTH TABLE

Figure 9-9. Logic Functions

9.9 HARDWARE REGISTERS

A hardware register is a device which accepts and stores parallel information. Most of the CMC's registers are 16 bit registers (comprised of 16 storage elements).

The basic storage element for one bit, its read and write service gates, and the pulse required for write-in and read-out are shown in figure 10-12. The register service gates control write-in and read-out. Each bit position (storage element) is cleared coincident with write-in. Data from the write line is applied to the write service gate and is written into the storage element under control of a write control pulse from the sequence generator. Data is read out of a storage element under control of a read control pulse from the sequence generator.

Hardware register use and interface is described in the central processor description, paragraph 10.10.2.

9.10 CMC ORGANIZATION

The CMC can be functionally divided into the following areas:

a. Timer.

b. Central Processor.

c. Sequence Generator.

d. Memory.

e. Priority Control.

f. Input - Output.

g. Power.

Each area (see figure 9-11) will be discussed separately in subsequent paragraphs.

9.10.1 TIMER. The timer generates the timing signals required for operation and synchronization of the computer and is the primary source of timing signals for all spacecraft systems.

The timer is divided into five functional areas (see figure 9-12):

a. Oscillator.

b. Clock divider logic.

c. Scaler.

d. Time pulse generator.

e. Sync and timing logic.

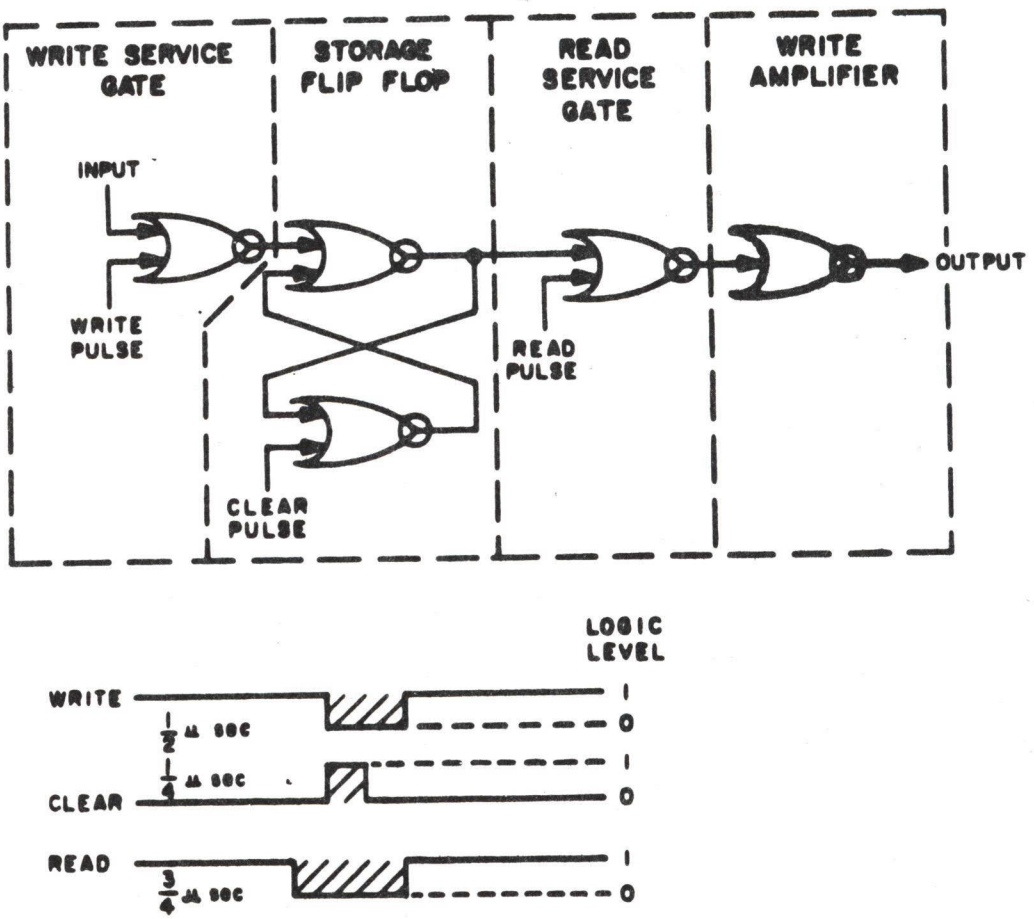


Figure 9-10. Basic Storage Element

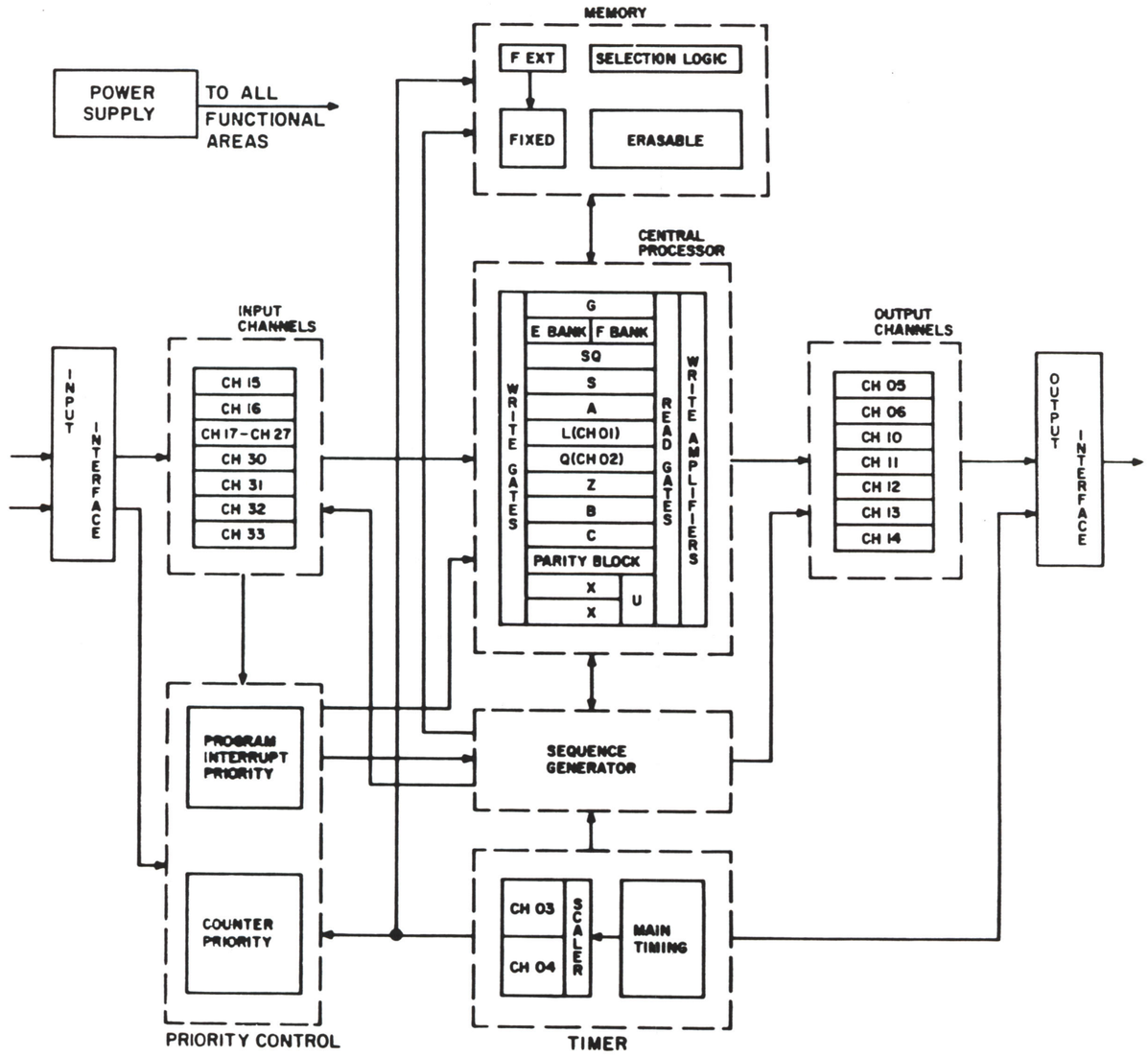


Figure 9-11. Computer Organization

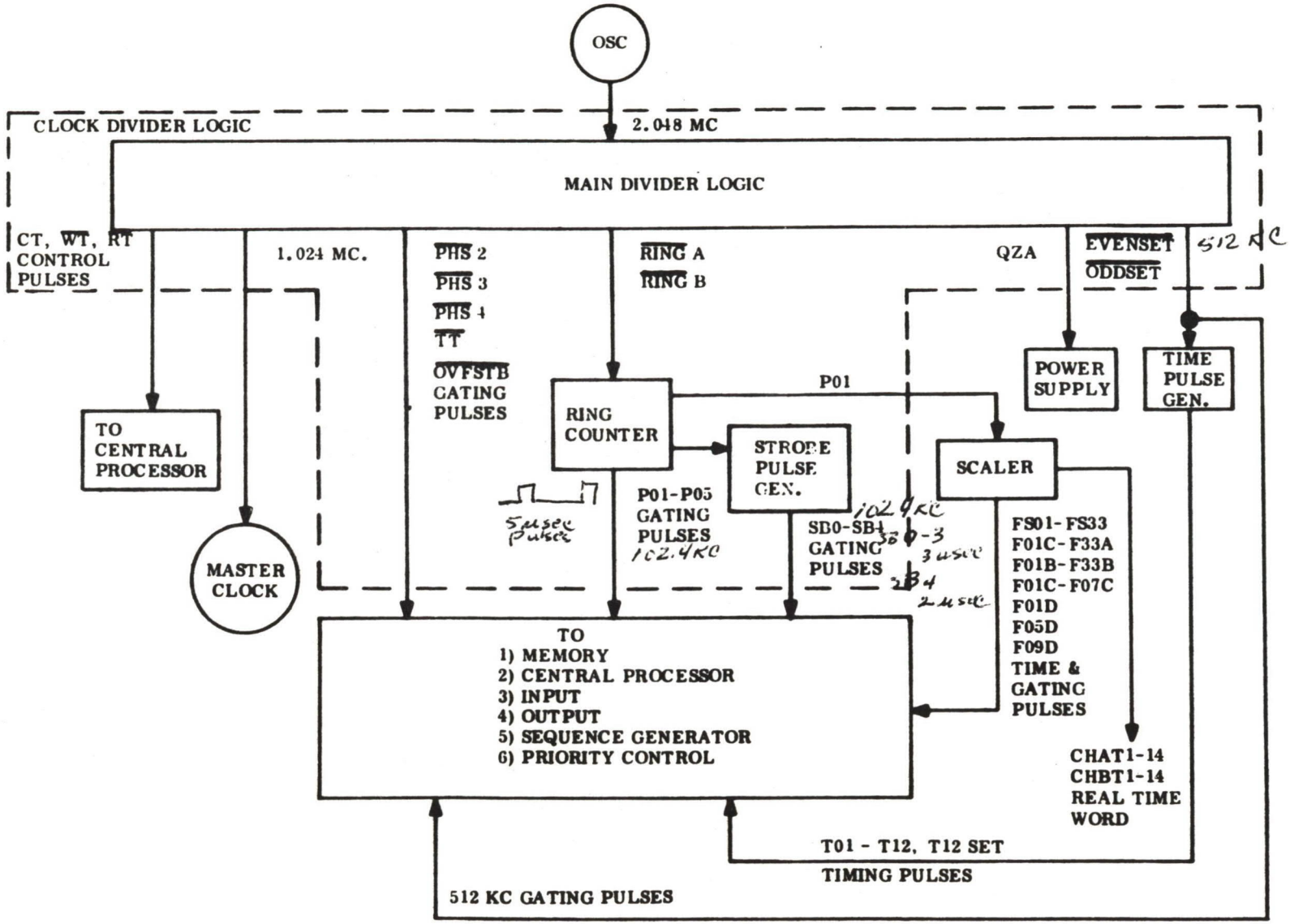


Figure 9-12. Timer, Functional Diagram

The master clock frequency is generated by an oscillator and is applied to the clock divider logic. The divider logic divides the master clock input into gating and timing pulses at the basic clock rate of the computer. Several outputs are available from the scaler, which further divides the divider logic output into output pulses and signals which are used for gating, to generate rate signal outputs and for the accumulation of time. Outputs from the divider logic also drive the time pulse generator which produces a recurring set of time pulses. This set of time pulses defines a specific interval (memory cycle time) in which access to memory and word flow take place within the computer.

The start-stop logic senses the status of the power supplies and specific alarm conditions in the computer, and generates a stop signal which is applied to the time pulse generator to inhibit word flow. Simultaneously, a fresh start signal is generated which is applied to all functional areas in the computer. The start-stop logic and subsequently word flow in the computer can also be controlled by inputs from the Computer Test Set (CTS) during pre-installation systems and subsystem tests.

9.10.1.1 Oscillator. The crystal controlled modified Pierce oscillator circuit generates a 2.048 mc square wave which is applied to the clock divider logic. Temperature compensated components in the oscillator circuit maintain high stability and assure an extremely accurate output frequency. The 2.048 mc signal is the prime source frequency for the timer, and, consequently, for the entire computer and space craft.

9.10.1.2 Clock Divider Logic. The clock divider logic is further sub-divided into the main divider logic section, ring counter and strobe pulse generator. The 2.048 mc input from the oscillator is applied to the main divider logic. This circuit divides the input frequency by two, and generates the following outputs: clear, write, and read control (CT, $\overline{\text{WT}}$, $\overline{\text{RT}}$) which are applied to the central processor to produce the signals necessary to clear, write into, and read out the flip-flop registers; 1.024 mc gating pulses ($\overline{\text{PHS2}}$, $\overline{\text{PHS3}}$, $\overline{\text{PHS4}}$, $\overline{\text{OVFSTB}}$, $\overline{\text{TT}}$) which are used throughout the CMC; the master clock signal (CLK) - a 1.024 mc output used to synchronize the other spacecraft systems; and signal QZA which is applied to the oscillator alarm circuit in the power supply to indicate oscillator activity. In addition, the main divider supplies signals ($\overline{\text{RING A}}$ and $\overline{\text{RING B}}$) to drive the ring counter, and signals ($\overline{\text{EVNSET}}$ and $\overline{\text{ODDSET}}$) to the time pulse generator. These latter outputs occur at a 512 kc rate as a result of further division of the 1.024 mc gating rate within the main divider.

The ring counter generates outputs (P01 - P05) at a 102.4 kc rate. The outputs are 5 microsecond pulses and are used for gating and for deriving other timing functions in the CMC. Ring counter outputs are also used to derive the strobe pulses (SB0, SB1, SB2, SB4) from the strobe pulse generator. These outputs also occur at a 102.4 kc rate and are 3 microseconds in width with the exception of SB4, which is a 2 microsecond pulse.

9.10.1.3 Scaler. The scaler consists of 33 identical divider stages. The stages are cascaded so that the frequency division is successive. The first stage is driven by signal $\overline{\text{P01}}$ from the ring counter, and generates outputs at a rate of one-half the input or 51.2 kc. This output and the remaining outputs through stage 17 (0.78125 pps) are used for timing and gating. The outputs appear as signal outputs from flip-flop circuits (FS01 etc.), and 10 microsecond pulse outputs (F01A etc.) at the same frequency as the associated stage. Stages 6 through 19 and 20 through 33 form a 28 bit real time word (CHAT01 - CHAT14, CHBT01 - CHBT14) which indicates time intervals up to 23.3 hours.

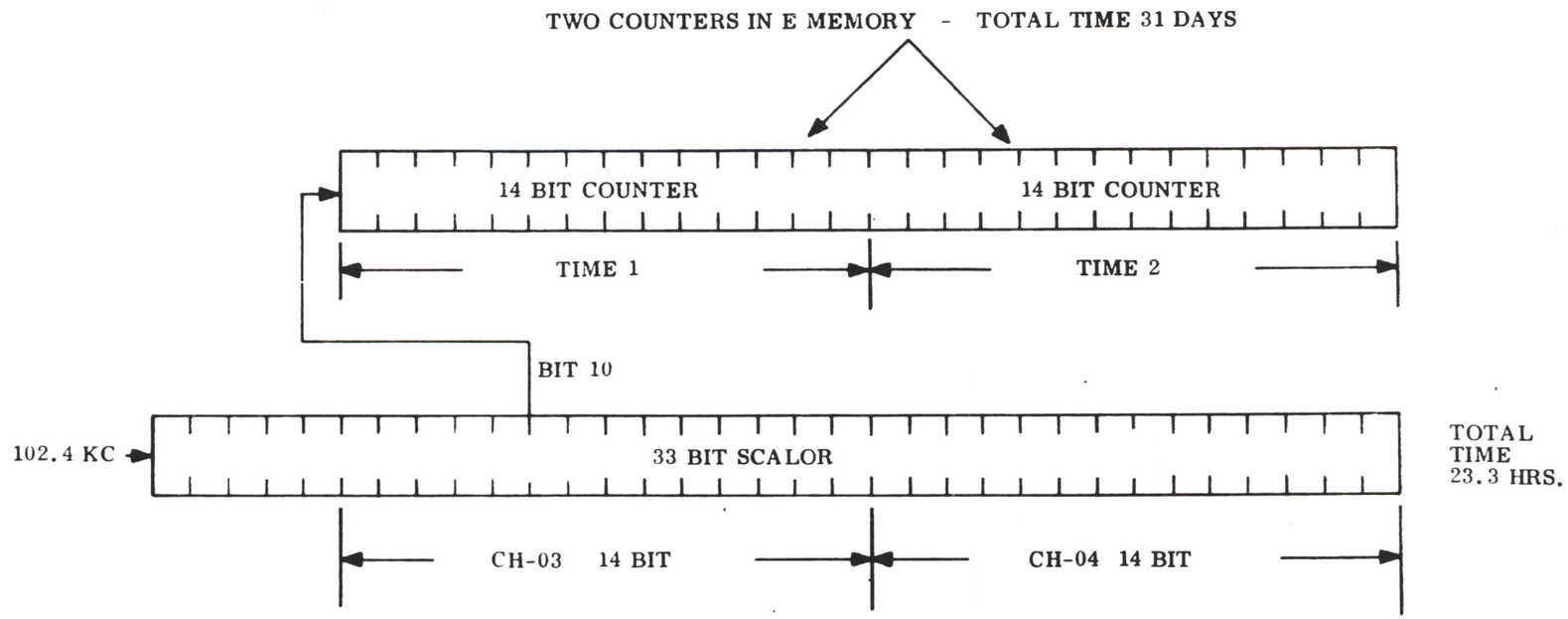


Figure 9-13. Scaler

9.10.1.4 Time Pulse Generator. The time pulse generator consists of 12 flip-flop circuits and generates timing pulses $\overline{T01}$ through $\overline{T12}$. (See figure 10-15.) This sequence of timing pulses defines one memory cycle time (MCT) within the CMC or a period of 11.97 microseconds, in which word flow takes place. The time pulse generator is driven by inputs ($\overline{EVNSET}$ and $\overline{ODDSET}$) from the main divider logic. Signal $\overline{ODDSET}$ can be inhibited by signal STOP from priority control. Signal STOP, an input from the Computer Test Set (CTS) during preinstallation system and subsystem tests, and subsequently inhibits the time pulses from being generated thus preventing word flow in the CMC. This feature allows individual memory cycle times to be observed during tests.

9.10.1.5 Sync and Timing Logic. The sync and timing logic (start-stop logic) consists of a gating complex which generates various outputs for use within the CMC, and synchronization signals for systems external to the CMC. The inputs to, and outputs from this section are rather extensive.

The ring counter, strobe pulse generator, and the scaler supply inputs to the sync and timing logic. These inputs are used to derive gating and strobe signals for the input and output channels, pulse outputs for the program time counters in memory, and synchronization signals for the computer and DSKY power supplies and for systems external to the computer.

During standby operation, the oscillator, clock divider logic, and the scaler are operative and generate the signals associated with these functional areas. However, the outputs that are significant during this mode of operation are the real time word from the scaler and the synchronization signals to the other spacecraft systems. The real time word continues to be accumulated during standby, and the external systems sync signals continue to be generated.

9.10.2 CENTRAL PROCESSOR. The central processor, figure 9-14, consists of the flip-flop registers, the write, clear, and read control logic, write amplifiers, memory buffer register, memory address register and decoder, and the parity logic. All data and arithmetic manipulations within the CMC take place in the central processor.

Primarily, the central processor performs operations indicated by the basic instructions of the program stored in memory. Communication within the central processor is accomplished through the write amplifiers. Data flows from memory to the flip-flop registers or vice-versa, between individual flip-flop registers, or into the central processor from external sources. In all instances, data is placed on the write lines and routed to a specific register or to another functional area under control of the write, clear, and read logic. This logic section accepts control pulses from the sequence generator and generates signals to read the content of a register onto the write lines, and write this content into another register of the central processor or to another functional area of the CMC. The particular memory location is specified by the content of the memory address register. The address is fed from the write lines into this register, the output of which is decoded by the address decoder logic. Data is subsequently transferred from memory to the memory buffer register. The decoded address outputs are also used as gating functions within the CMC.

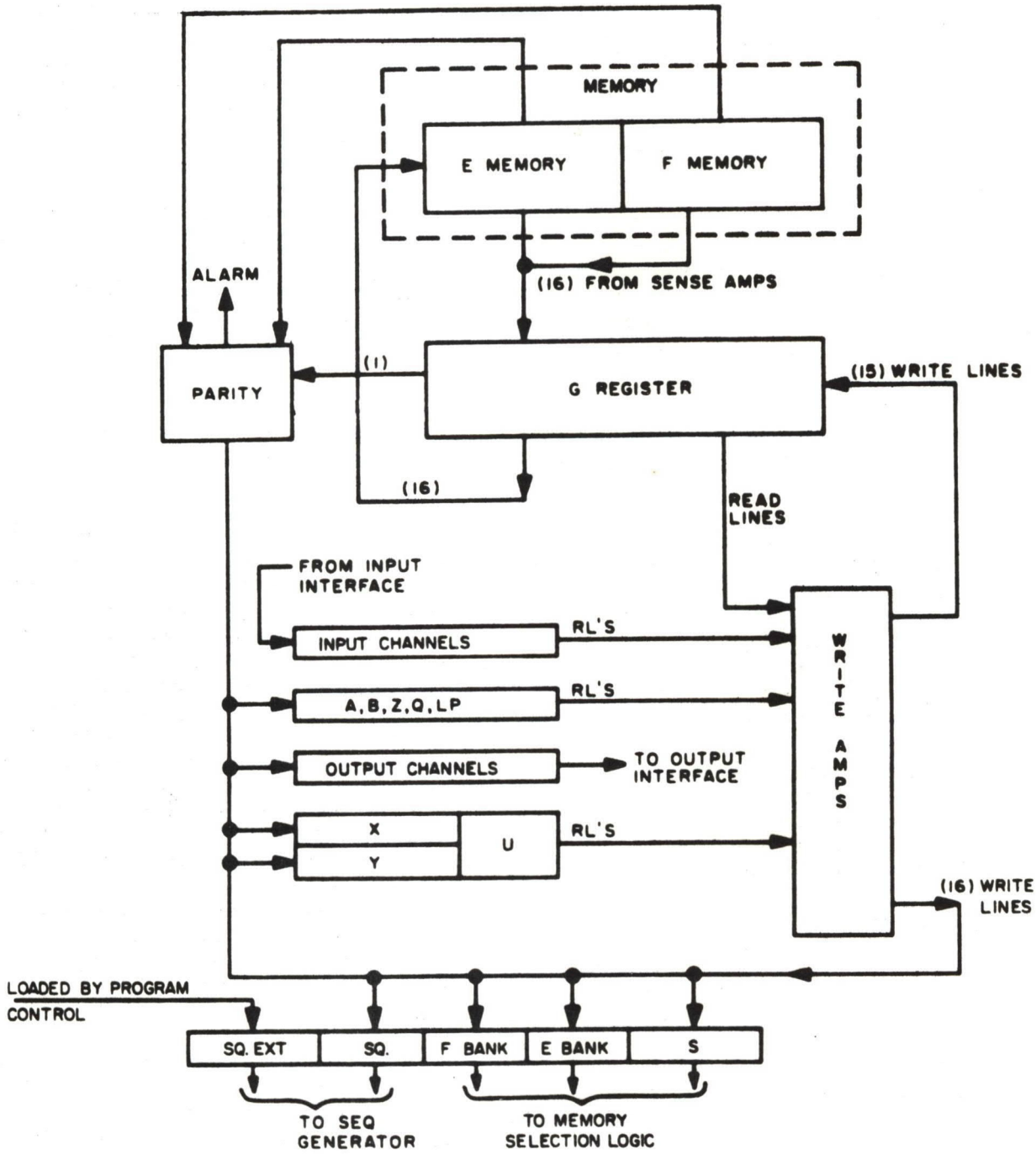


Figure 9-14. Central Processor Block Diagram

The memory buffer register buffers all information read out or written into memory. During readout, parity is checked by the parity logic and an alarm is generated in case of incorrect parity. During write-in, the parity logic generates a parity bit for information being written into memory. The flip-flop registers are used to accomplish the data manipulations and arithmetic operations. Each register is 16 bits or one computer word in length. Data flows into and out of each register as dictated by control pulses associated with each register. The control pulses are generated by the write, clear, and read control logic.

External inputs through the write amplifiers include the content of both the erasable and fixed memory bank registers, all interrupt addresses from priority control, control pulses which are associated with specific arithmetic operations, and the start address for an initial start condition. Information from the input and output channels is placed on the write lines and routed to specific destinations either within or external to the central processor. The CTS inputs allow a word to be placed on the write lines during system and subsystem tests.

9.10.2.1 Registers. Each register can be considered to contain 16 bits, unless otherwise specified. A description of each register in the central processor is given in figure 9-15.

9.10.2.2 Basic Data Flow. Figure 9-14 is a block diagram of the central processor including interface with other computer functional areas. When a word is gated out of memory, it is stored just in register G. Register G is a temporary storage area. If the data came from erasable memory, it is normally restored into erasable memory after it is read into some other register. A parity bit is removed from all words that come from memory. It is routed to the parity block where it is used to make the parity check. It is the interface between register G and the write amps that causes a duplication of the sign bit to form the overflow bit. This overflow bit is not written into the parity block; however, all other registers will receive this bit. There are 16 write amp outputs which can be gated into any of the central processor registers under control of the sequence generator. Words that are to be written into memory must have a parity bit generated for them in the parity block. This parity bit is sent directly to register G where it will join the appropriate word that has been gated into register G via the write amps. Words that are routed from the write amps to register G do not contain the overflow bit. This bit is replaced by the newly generated parity bit. The entire word is then written into erasable memory. In addition to temporarily storing all words to and from memory, register G will cycle or shift words to the right or left when programmed to do so. When shifting words to the left or right, the last bit is lost. However, in cycling, bit 1 becomes bit 16 or bit 16 becomes bit 1. The addresses mentioned will contain the cycled or shifted contents when used with most order codes.

9.10.2.3 Adder. The adder consists of registers X and Y and associated carry gates. The adder is the only true arithmetic manipulation circuitry in the CMC. Its function is to add two numbers together. Provisions are made to handle carries when necessary. The adder has the capability of adding + 1 to a number by forcing an end-around-carry.

Register Designation	Address (Octal)	Bits	Name and Purpose
A	0000	16	1. Called the accumulator 2. Usually used to store results of these manipulations: a. Addition b. Subtraction c. Multiplication d. Division e. "AND"ing f. "OR"ing g. "EXCLUSIVE OR"ing
L	0001	16	1. Called the lower order accumulator. 2. Used to store results of these manipulations: a. Multiplication b. Division 3. Sometimes used in channel instructions.
Q	0002	16	1. Called the return address register. 2. Used to store the contents of the Z register, when the computer transfers control to another operation. 3. Used to exchange information with addresses in the central processor or erasable memory.
Z	0005	16	1. Called the program counter. 2. Contains the address of the next instruction word in the program.
B	0006 ~~None~~	16	1. A buffer register. 2. Used to hold the order code of an instruction. 3. Used to hold address information. 4. Used as a temporary storage of data information. 5. Has the ability to read out its actual contents or the compliment of its contents.
S	None	12	1. Contains the address portion of an instruction.
SQ	None	7 (includes an ext. bit)	1. Referred to as the instruction register. 2. Contains the operand of an instruction word.

Figure 9-15. Register Designation and Function (Sheet 1 of 2)

Register Designation	Address (Octal)	Bits	Name and Purpose
X and Y	None	16 ea.	1. Are referred to as the adder. 2. Performs all arithmetic operations. 3. The output gates of the adder are sometimes called the U register.
G	None	16	1. A buffer between memory and the central processor. 2. All information going to or coming from memory passes through this buffer register. 3. Any parity bit received from memory is transferred to the parity block before going to the central processor.
E-BANK	0003	3	1. Used to complete an address in erasable memory.
F-BANK	0004	8 (includes 3 ext. bits)	1. Used to complete an address in fixed memory.

Figure 9-15. Register Designation and Function (Sheet 2 of 2)

9.10.2.4 Parity Block. The parity block provides a means of self-checking the computer memory and transfer circuits. The parity block must test the parity of words just arriving in the central processor from memory, and also generate a new parity bit for words being stored in erasable memory. Any time a word is read out of memory, its parity must be tested. Every time a word is written into memory, its parity must be generated. The parity block is active during all of the 34 regular instructions and some of the involuntary ones.

The essence of the parity test is as follows:

a. Logically (using NOR logic elements) determine whether the number of ONE's in a word from memory is odd or even (excluding the parity bit).

b. Compare the parity bit with the information found in (a) above. If even, the parity bit should be 1 if the total number of ONE's is to be odd. If odd, the parity bit should be 0 for the same reason.

c. Any other condition except those defined in (b) above is an alarm condition.

The essence of the parity bit generation is as follows:

a. Repeat step (a) above.

b. If there is an even number of ONE's (excluding parity bit), generate parity bit = 1. If there is an odd number of ONE's (excluding parity bit), generate parity = 0.

9.10.2.5 Registers S, SQ, F BANK and E BANK. The Sq register will receive bits 16 and 14 through 10 directly from the write amps under control of the sequence generator. The S register receives bits 1 through 12 from the write amps under control of the sequence generator. The F BANK register will receive bits 16 and 14 through 11 from the write amps under program control. The E BANK register receives bits 9, 10 and 11.

Registers A, L, Q, F BANK, E BANK, and Z are addressable; registers B, S, SQ, Y, X and G are not addressable. A register is addressable when it can be selected for write-in or read-out by entering the proper address into register S.

A word can be transferred from one central processor register to another either via the write amplifiers or directly. Direct transfer, however, takes place only among four of the registers. Read and write control pulses occur simultaneously and gate the read gates of the transmitting register and the write gates of the receiving register simultaneously. This also causes data to flow in parallel from one register to another via the write amps. The flow of information from one register to another via the write amps occurs on two different sets of lines: the read lines and the write lines. When one read pulse and two or three write control pulses occur simultaneously, the same information is entered into two or three registers. Direct read and write pulses transfer information directly from one register to only one other register.

9.10.3 SEQUENCE GENERATOR. The sequence generator (figure 9-16) contains the order code processor, command generator, and control pulse generator. The sequence generator executes the instructions stored in memory by producing control pulses which regulate the data flow of the computer. The manner in which the data flow is regulated among the various functional areas of the computer and between the elements of the central processor causes the data to be processed according to the specifications of each machine instruction.

The order code processor receives signals from the central processor, priority control, and peripheral equipment (test equipment). The order code signals are stored in the order code processor and converted to coded signals for the command generator. The command generator decodes these signals and produces instruction commands. The instruction commands are sent to the control pulse generator to produce a particular sequence of control pulses depending on the instruction being executed. At the completion of each instruction, new order code signals are sent to the order code processor to continue the execution of the program.

9.10.3.1 Order Code Processor. The order code processor (figure 9-17) consists of the register SQ control, register SQ and decoders, and stage counter and decoders. The register SQ control is regulated by special purpose control pulse NISQ from the control pulse generator. Control pulse NISQ produces clear and write signals for register SQ and initiates a read signal for register B. The clear, read, and write signals place the order code content of register B onto the write lines and into register SQ. The order code signals from the priority control and the peripheral equipment pertain to instructions start, interrupt, and transfer control to specified address. These order code signals cause the SQ control to produce the clear signals. If the order code signal is start or transfer control to specified address, no further action occurs because the order code for each of these instructions is binary 0 000 000. If the order code signal is interrupt, register SQ is set to 1 000 111. Other special purpose control pulses provide regulatory functions within the register SQ control during instruction interrupt and some address - dependent instructions.

Register SQ is a seven-bit register with only six of its bit positions (16 and 14 through 10) connected to the central processor write lines. The seventh (high-order) bit position is the extend bit. This high-order bit position is used for extending the order code field; it contains a ZERO for basic instructions and a ONE for extracode, channel, and interrupt instructions. Bit positions 16, 14, and 13 produce the SQ signals. At any time, only one of the eight possible SQ signals is present to indicate the octal number specified by these bit positions. Bit positions 12 and 11 contain the quarter code. These bits are decoded into one of four QC signals to indicate the octal number specified by these two bit positions. Bit position 10 is not used for basic and extracode instructions; however, it is used for the channel and interrupt instructions.

The stage counter is a three-stage Gray counter especially adapted for various counts other than the Gray code. Most instructions are several memory cycle times (MCT's) long. The stage counter controls the length of each instruction. Most instructions use the two low-order bits of the stage counter. The stage counter always starts an instruction with count 000. Then it may be advanced to 001, 010, or 011 by special purpose control pulses ST1 and ST2 from the control pulse generator. The Gray code count is used for the divide instruction. Control pulse DVST advances the counter through the states, 000, 001, 011, 111, 110, and 100. The control pulse ST2 sets the stage counter to 010 to complete the divide instruction. The content of the stage counter is decoded into the ST code signals. Some of the ST code signals reflect the standard binary count

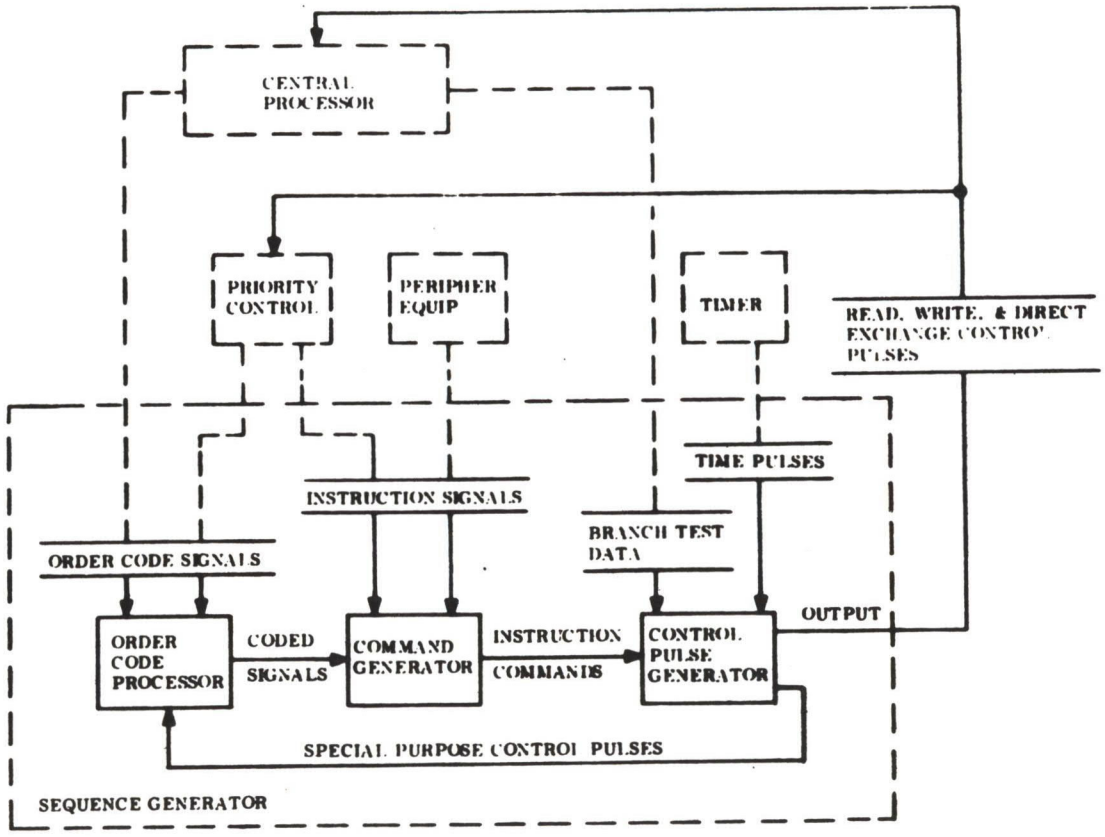


Figure 9-16. Sequence Generator Block Diagram

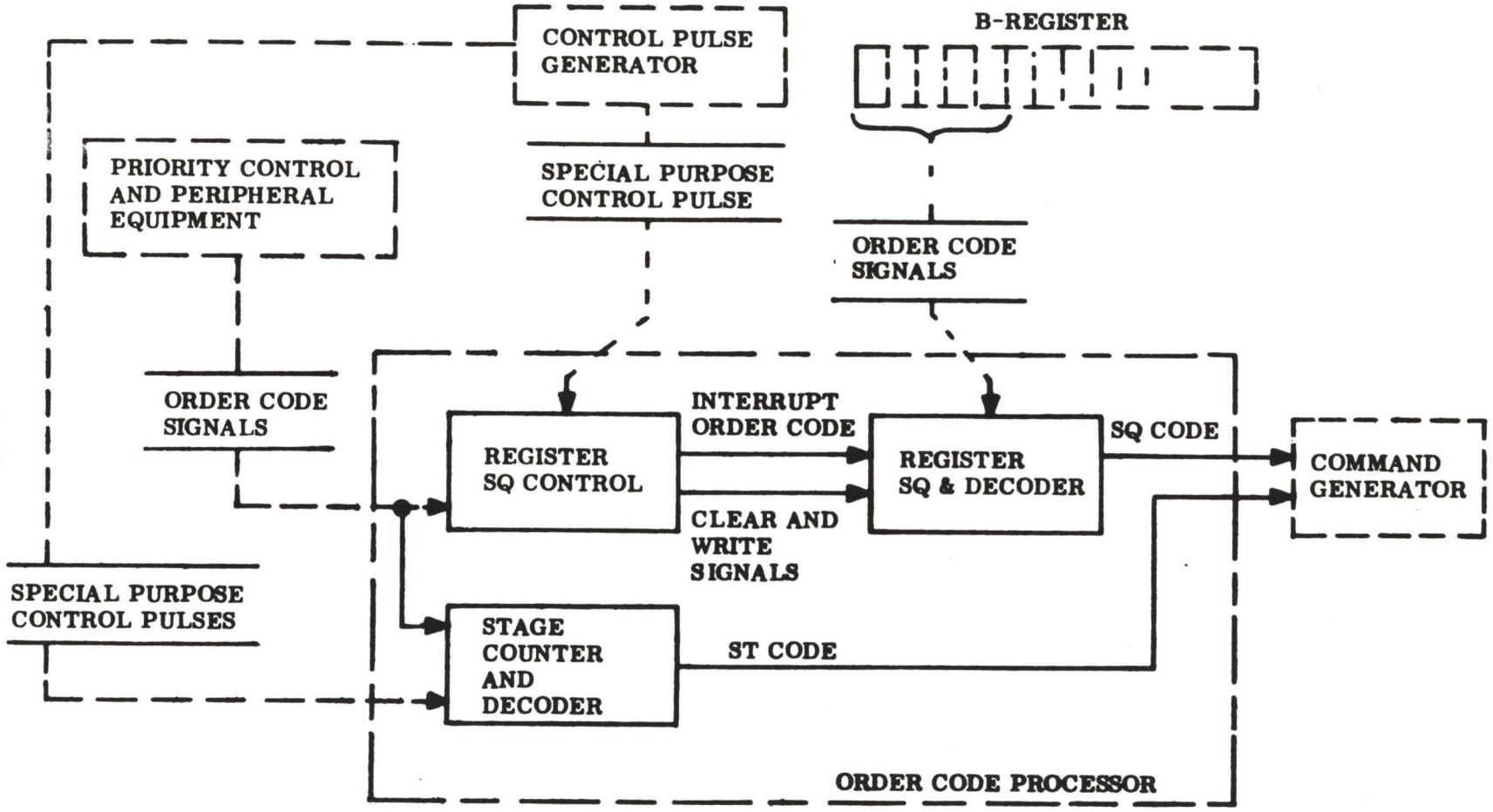


Figure 9-17. Order Code Processor, Block Diagram

from octal 0 through 3, and others reflect the Gray code count of octal 0, 1, 3, 7, 6, and 4. The order code signals from the priority control and the peripheral equipment set the stage counter to a particular state in a manner similar to that in which the SQ register is set. The interrupt order code signal sets the stage counter to 000, the start order code signal sets it to 001, and the transfer control to specified address signal sets it to 011. The outputs of the SQ and stage decoders are sent to the command generator where they are used to produce subinstruction and instruction commands.

9.10.3.2 Command Generator. The command generator (figure 9-18) contains the subinstruction decoder, instruction decoder, and the counter and peripheral instruction control. The subinstruction decoder receives the SQ and ST code signals from the order code processor. These signals represent the order codes of all machine instructions and are decoded into subinstruction and instruction commands. For example, channel instruction WOR has a binary order code 1 000 101 and stage code 000. The SQ code signals SQEXT, SQ0, QC2, and SQR10 are combined with ST code signal ST0 to produce subinstruction command WOR0.

The instruction decoder receives the coded signals from the order code processor in addition to certain subinstruction commands. It produces signals called instruction commands. An instruction command is used for two or more subinstructions as compared to a subinstruction command which is used only for one subinstruction. For example, instruction command IC1 generates a combination of control pulses shared by subinstruction NDX0 and NDXX0. Instruction command IC1 is produced by signals SQEXT, SQ5, and ST0 for subinstruction NDX0 or by signals SQ5, QC0, and ST0 for subinstruction NDXX0. Other instruction commands are produced from subinstruction commands. For example, IC8 is produced by ORing DXCH0 with LXCH0.

The counter and peripheral instruction control receives instruction signals from the priority control and the peripheral equipment. These signals are applied to separate circuits which control the individual counter and peripheral instructions. The instruction signals from the priority control pertain to counter locations and the instruction(s) associated with each location. For example, signal C31A is interpreted as counter 31 address. The content of this location can only be changed by instruction DINC whose subinstruction command is produced by the counter and peripheral instruction control. Another example is signal C42P, interpreted as counter 42 positive increment or signal C42M, counter 42 negative increment. The peripheral equipment supplies instruction signals such as MREAD and MLOAD for the fetch and store instructions, respectively. While the particular instruction is being executed, the counter and peripheral instruction control stores the input signals in the same way that order code signals are stored by register SQ. Since some of the peripheral instructions are several MCT's long, they use the ST code signals. The subinstruction and instruction command outputs of the command generator are used by the control pulse generator in conjunction with time pulses T01 through T12 to produce action pulses.

9.10.3.3 Control Pulse Generator. The control pulse generator (figure 9-19) contains the crosspoint generator, control pulse gates, and branch control. The crosspoint generator receives instruction and subinstruction commands from the command generator and branch commands from the branch control. The crosspoint generator produces an action pulse when a command signal and a time pulse are ANDed. This is called the crosspoint operation. For example, action pulse 5XP12 is produced from subinstruction command DAS0 and time pulse T05. Many instructions use identical action pulses. When this is the case, several command signals such as TC0, TCF0,

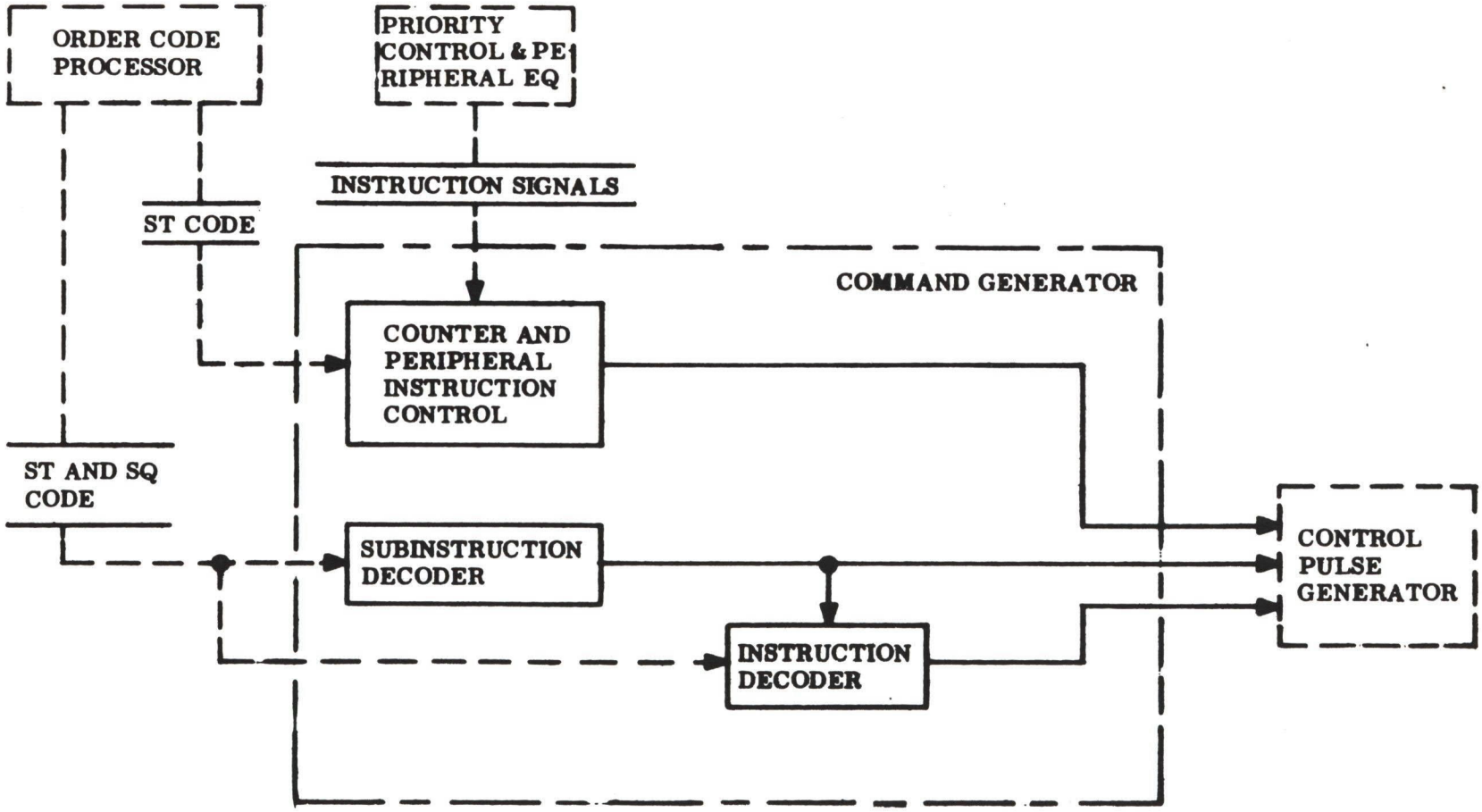


Figure 9-18. Command Generator, Block Diagram

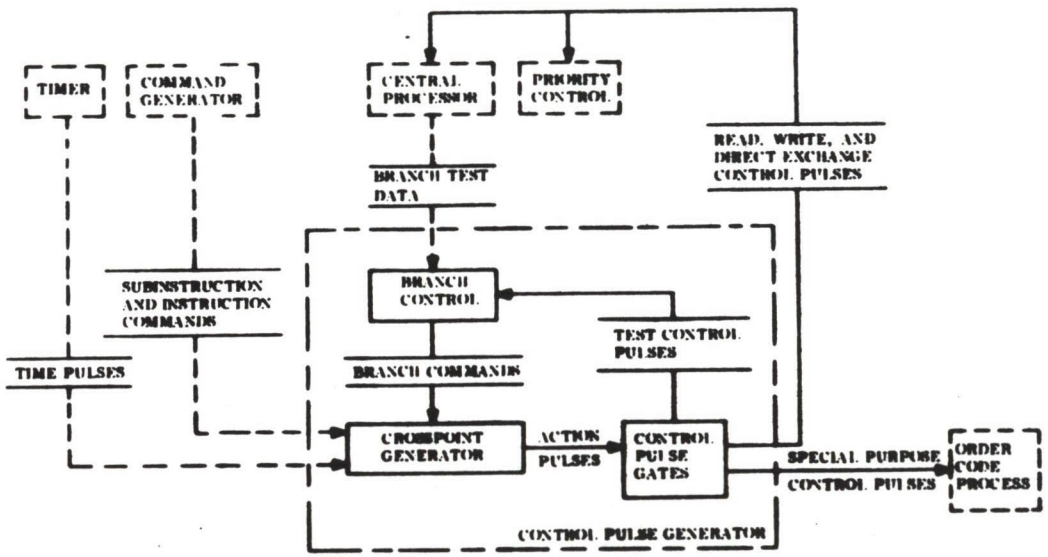


Figure 9-19. Control Pulse Generator, Block Diagram

or IC4 will produce the same action pulse during time period T01. The branch commands are used to change the action pulse that normally is produced at a given time. For example, when certain conditions exist, a branch command will produce action pulse 8XP6 in addition to another action pulse normally produced at time period T08. The action pulses are supplied to the control pulse gates which convert them to specific control pulses for use in instruction execution.

The control pulse gates perform the Boolean NOR function. There is one gate for each control pulse. These gates split the action pulses into as many control pulses as are required for a particular operation. For example, action pulse 3XP6 is converted to control pulses RZ and WQ. Some of the control pulses produced by the control pulse gates are used by the sequence generator. These include the special purpose control pulses which control the operation of the order code processor and the test control pulses which are applied to the branch control. The other control pulse groups, namely the read, write, and direct exchange control pulses are used in the central processor and the priority control.

The branch control is connected to the write lines of the central processor. Data which is placed onto the write lines by read control pulses is tested in the branch control. The branch control contains two stages. Branch 1 normally tests for sign and branch 2 test for full quantities such as plus or minus zero. Both branches test for positive and negative overflow and have the overflow bits written directly into the branch register. Positive overflow is 01 where branch 1 is the high order bit. Negative overflow is 10. The branch commands sent to the crosspoint generator affect the action pulses at given times. The branch control also contains the special instruction flip-flop which controls the execution of instructions RELINT, INHINT, and EXTEND.

9.10.3.4 Register SQ Control. The register SQ control is regulated by special purpose control pulse NISQ from the control pulse generator. Control pulse NISQ causes the register SQ control to produce clear signal CSQG, read signal RBSQ, and write signal WSQG. These signals place the order code (content of register B) onto the write lines and into register SQ at the beginning of each new instruction. The order code signals applied to the register SQ control from the priority control (GOJAM and RUPTOR) and peripheral equipment (MTCSAI) pertain to instructions start, interrupt, and transfer control to specified address, respectively. A distinct priority is associated with each of these three instructions. Instructions interrupt and transfer control to specified address can never be requested when the computer is forcing the execution of the start instruction, which has the highest priority. Certain peripheral instructions occupy the next level of priority, followed by the counter instructions and in turn the transfer control to specified address instruction, which has priority over the interrupt instruction; all six of these instruction categories have priority over basic instructions.

9.10.4 DESCRIPTION OF COMPUTER INSTRUCTIONS

The same instructions, which are directions given to perform specific operations, are the same for the CMC and LGC. Together with data addresses, they constitute the building blocks of a program. Programs are sequential lists of instruction words. There are two general categories of instructions, machine and interpretive. Several types of instructions used in the computer may be categorized as follows:

MACHINE (56)

REGULAR (42)

BASIC (15)

EXTRACODE (12)

CHANNEL (7)

SPECIAL (8)

INVOLUNTARY (9)

INTERRUPT (2)

COUNTER (7)

PERIPHERAL (5)

INTERPRETIVE

The machine instructions can be interpreted and executed directly by using the sequence generator to control the computer operation. The interpretive instructions are a programmer's convenience and must be interpreted under program control, converted to machine instructions and then executed as machine instructions. Table 9-2 lists the machine instructions alphabetically and gives a brief description of each. The reader will find it to his advantage to refer back to this table once he has gained a greater familiarity with the computer. The following symbols are used in table 9-2.

K represents any address in the central processor, erasable memory or fixed memory.

F represents an address in fixed memory only.

E represents an address in the central processor or erasable memory.

H represents any channel address.

C represents any counter address.

A represents the A register on the central processor.

L represents the L register in the central processor.

c(K) represents the contents of K, i.e., the data located in address K.

I, I + 1, I + 2 represents the addresses of successive instruction words stored in memory.

c (I), C (I + 1), C (I + 2) represents the contents of successive instruction words stored in memory.

9.10.4.1 Machine Instructions. The CMC has three classes of machine instructions: regular, involuntary, and peripheral. Regular instructions are programmed and are executed in whatever sequence they have been stored in memory. Involuntary instructions (with one exception) are not programmable and have priority over regular instructions: no regular instruction can be executed when the CMC forces the execution of an involuntary instruction. The peripheral instructions are used during ground testing when the CMC is connected to the CTS or PAC: the CMC cannot perform any program operation during a peripheral instruction.

9.10.4.1.1 Regular Instructions. There are four types of regular instructions: basic, channel, extracode, and special. The difference between the regular instructions is directly related to the way in which the CMC interprets an instruction word. Instruction words stored in memory are called "basic instructions words" and consist of a three bit order code and a twelve bit address code. The order code defines an operation and the address code defines a location.

The contents of the SQ register will determine what instruction the CMC will perform. The SQ register reflects that data transferred into it from memory. The SQ register consists of six bits and one EXT. bit (figure 9-20). A binary point is assumed to be located between bits thirteen and twelve. When an instruction word is transferred from memory to the SQ register, bits 15 through 10 of the word in memory are transferred to bits 16 and 14 through 10 of the SQ register (figure 9-21). In the following paragraph, however, only the transfer of bits 15, 14 and 13 from memory to bits 16, 14 and 13 of the SQ register will be considered.

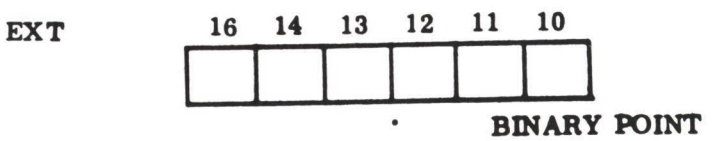


Figure 10-23. SQ Register

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 1 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
AD K	06.	Basic Instruction: add c(K) to c(A); stores result in A; takes next instruction from I + 1 where I is location of AD K.	2
ADS E	02.6	Basic Instruction: adds c(A) to c(E) and stores result in both A and E; takes next instruction from I + 1 where I is location of ADS E.	2
AUG E	12.4	Extra Code Instruction: adds +1 to c(E), if c(E) is positive and -1 if c(E) is negative; stores result in E; takes next instruction from I+1 where I is location of AUG E. NOTE: AUG, DIM and INCR are slightly modified counter increment sequences. Accordingly, if one of this group overflows when addressing a counter for which overflow (during involuntary incrementing) is supposed to cause an interrupt, the interrupt will occur. It should be noted that all three of these instructions unlike the increment sequences, always operate in one's complement, even when addressing CDU counters.	2
BZF F	11.2 11.4 11.6	Extra Code Instruction: takes next instruction from F if c(A) is ±0; otherwise takes next instruction from I + 1 where I is location of BZF F.	1 if c(A) is ±0; otherwise 2
BZMF F	16.2 16.4 16.6	Extra Code Instruction: takes next instruction from F if c(A) is +0 or negative; otherwise takes next instruction from I + 1 where I is location of BZMF F.	1 if c(A) is +0 or negative, otherwise 2
CA K	03.	Basic Instruction: clears c(A) and copies c(K) into A; takes next instruction from I + 1 where I is location of CA K.	2

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 2 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
CCS E	01.0	Basic Instruction: if c(E) is nonzero and positive, takes next instruction from I + 1 where I is location of CCS E, adds -1 to c(E) and stores result in A. If c(E) is +0, takes next instruction from I + 2 and sets c(A) to +0. If c(E) is nonzero and negative, takes next instruction from I + 3, adds -1 to the absolute value of the c(E) and stores result in A. If c(E) is -0, takes next instruction from I + 4 and sets c(A) to +0.	2
CS K	04.	Basic Instruction: clears c(A) and copies $\overline{c(K)}$ into A; takes next instruction from I + 1 where I is location of CS K.	2
CYL	.0022	Special Instruction: cycles quantity, which is entered into location 0022, one place to left.	
CYR	.0020	Special Instruction: cycles quantity, which is entered into location 0020, one place to right.	
DAS E	02.0	Basic Instruction: adds c(A, L) to c(E, E + 1); stores result in E and E + 1; sets c(L) to +0 and sets c(A) to net overflow if address E is not 0000_8. Net overflow is +1 for positive overflow, -1 for negative overflow, otherwise c(A) is set to +0. Takes next instruction from I + 1 where I is location of DAS E. Note: DAS A doubles the contents of the double precision accumulator - implied address code DDOUBL assembles as DAS A. Since the nardware must operate on the low order operands first, consider DAS as the operation code 20001 to which the address E is added to for the instruction.	3
DCA K	13.	Extra Code Instruction: copies c(K, K + 1) into A and L; takes next instruction from I + 1 where I is location of DCA K.	3

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 3 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
DCS K	14.	Extra Code Instruction: copies c(K, K + 1) into A and L; takes next instruction from I + 1 where I is location of DCS K.	3
DIM E	12.6	Extra Code Instruction: adds -1 if c(E) is nonzero and positive and +1 if c(E) is nonzero and negative; stores result in E; if c(E) is ±0, c(E) is not changed; takes next instruction from I + 1 where I is location of DIM E. See NOTE under AUG.	2
DINC C	None	Counter Instruction: adds +1 to c(C) if c(C) is negative; adds -1 to c(C) if c(C) is positive; provides no change if c(C) is ±0; stores result in C, delays program execution for 1 MCT.	1
DV E	11.0	Extra Code Instruction: divides c(A, L) by c(E): stores quotient in A; stores remainder in L; takes next instruction from I + 1 where I is location of DV E. NOTE: The signs of the double length dividend in A & L need not agree. The net sign of the dividend is the sign of c(A) unless c(A) is ±0, in which case it is the sign of c(L). The remainder bears the net dividend sign, and the quotient sign is determined strictly by the divisor and net dividend signs.	6
DXCH E	05.2	Basic Instruction: exchanges c(E, E + 1) with c(A, L); takes next instruction from I + 1 where I is location of DXCH E.	3
EXTEND	00.0006	Special Instruction: Take the next instruction from I + 1, where I is the EXTEND instruction and execute it as an extracode instruction. If I + 1 is INDEX (full operation code 15), the following instruction will also be executed as an extracode.	1
FETCH K	None	Peripheral Instruction: reads and displays c(K) as binary numbers on CTS or PAC where K is address supplied by CTS or PAC.	2

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 4 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
EDOP	.0023	Special Instruction: shifts quantity, which is entered into location 0023, seven places to left.	
COJ	00.	Interrupting Instruction: transfers control to instruction stored in location 4000_8 and proceeds from there.	2
INCR E	02.4	Basic Instruction: adds + 1 to c(E); stores result in E; takes next instruction from I + 1 where I is location of INCR E. See NOTE under AUG.	2
INHINT	00. 0004	Special Instruction: Inhibit program interrupts until a subsequent RELINT. Take the next instruction from I + 1 where I was INHINT. NOTE: The inhibition set by INHINT and removed by RELINT in entirely independent of the one set by an interrupt and removed by a RESUME.	1
INOT LD H	None	Peripheral Instruction: loads data supplied by CTS or PAC into location H where H is channel address also supplied by CTS or PAC.	1
INOTRD H	None	Peripheral Instruction: reads and displays c(H) as binary number on CTS or PAC where H is channel address supplied by CTS or PAC.	1
LXCH E	02.2	Basic Instruction: exchanges c(E) with c(L); takes next instruction from I + 1 where I is location of LXCH E.	2
MCDU C	None	Counter Instruction: adds -1 (two's complement) to c(C). NOTE: Incrementing in two's complement modulator notation transfers octal 40000 to 57777 and 00000 to 77777 and is otherwise like one's complement. PCDU and MCDU replace PINC and MINC for counters 0032 through 0036.	1

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 5 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
MINC C	None	Counter Instruction: adds -1 to c(C); delays program execution for 1 MCT. If negative overflow occurs, c(C) is set to -0.	1
MP K	17.	Extra Code Instruction: multiplies c(A) by c(K); stores result in A and L; c(A, L) agree in sign; takes next instruction from I + 1 where I is location of MP K. A zero result is positive unless c(A) = ±0 and c(K) is non-zero with the opposite sign.	3
MSK K	07.	Basic Instruction: AND's c(A) with c(K); stores result in A; takes next instruction from I + 1 where I is location of MSK K.	2
MSU E	12.0	Extra Code Instruction: forms signed one's complement difference between c(A) and c(E) where c(A) and c(E) are unsigned (modular or periodic) two's complement numbers; stores result in A; the method is to form the two's complement difference, to decrement it if it is negative, and to take the overflow-uncorrected sum as the result; takes next instruction from I + 1 where I is location of MSU E.	2
NDX K	05.0	Basic Instruction: adds c(K) to c(I + 1) where I is location of NDX E; takes sum of c(K) + c(I + 1) as next instruction. INDEX 0017 is an implied instruction to resume an interrupted program.	2
NDX K	15.	Extra Code Instruction: adds c(K) to c(I + 1) where I is location of NDX K; sets extra code switch: sum of c(K) + c(I + 1) becomes an Extra Code Instruction which is taken as next instruction. This INDEX will not act as a RESUME.	2
PCDU C	None	Counter Instruction: adds +1 (two's complement) to c(C); delays program execution for 1 MCT. See NOTE under MCDU.	1

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 6 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
PINC C	None	Counter Instruction: adds +1 to c(C); delays program execution for 1 MCT. If positive overflow occurs, the counter is set to +0 and an interrupt is set up if the counter is T3, T4, T5 or set up a PINC for T2 if the counter was T1.	1
QXCH E	12.2	Extra Code Instruction: exchanges c(E) with c(Q); takes next instruction from I + 1 where I is location of QXCH E.	2
RAND H	10.2	Channel Instruction: AND's c(H) with c(A); stores result in A; takes next instruction from I + 1 where I is location of RAND H.	2
READ H	10.0	Channel Instruction: copies c(H) into A; takes next instruction from I + 1 where I is location of READ H.	2
ROR H	10.4	Channel Instruction: Inclusive OR's c(H) with c(A); stores result in A; takes next instruction from I + 1 where I is location of ROR H.	2
RELINT	00.0003	Special Instructions: Removes program interrupt inhibits. Allows interrupts after this instruction subject to the restriction that an interrupt cannot occur while there is plus or minus overflow in A.	1
RESUME	05.0017	Special Instruction: takes next instruction from return address (location of which address is stored in location 0017). This allows the resumption of the interrupted program.	1
RUPT	10.7	Interrupting Instruction: takes next instruction from address supplied by Interrupt Priority Control; stores c(B) (instruction that was to be executed) in location 0017_8; stores c(Z) = I in location 0015_8 where I is assigned location of instruction stored in 0017. This instruction is for machine checkout only.	3

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 7 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
RXOR H	10.6	Channel Instruction: forms the exclusive OR of c(H) and c(A); stores result in A; takes next instruction from I + 1 where I is location of RXOR H.	2
SHANC C	None	Counter Instruction: doubles c(C) and adds +1; stores result in C; delays program execution for 1 MCT. This action amounts to shifting c(C) one digit to the right and adding +1. NOTE: SHANC and SHINC are used to convert incoming serial bit streams into words for parallel access.	1
SHINC C	None	Counter Instruction: doubles c(C); stores result in C; delays program execution for 1 MCT. This action amounts to shifting c(C) one digit to the right. See NOTE under SHANC.	1
SR	.0021	Special Instruction: shifts quantity, which is entered into location 0021, one place to right.	
STORE E	None	Peripheral Instruction: data supplied by CTS or PAC is stored in location E where E is address supplied by CTS or PAC; delays program execution for 2 MCT's.	2
SU E	16.0	Extra Code Instruction: subtracts c(E) from c(A); stores result in A; takes next instruction from I + 1 where I is location of SU K.	2
TC K	00.	Basic Instruction: takes next instruction from K; stores I + 1 in Q where I is location of TC K; if K is 0006 (EXTEND), sets extra code switch and takes next instruction from I + 1; if K is 0004_8 (INHINT), sets inhibit interrupt switch and takes next instruction from I + 1; if K is 0003_8 (RELINT), resets inhibit interrupt switch and takes next instruction from I + 1.	1

Table 9-2. Machine Instructions, Alphabetical Listing (Sheet 8 of 8)

Symbolic Instruction Word	Order Code	Description	Execution Time in MCT's
TCF F	01.2 01.4 01.6	Basic Instruction: takes next instruction from F. Does not change the contents of Q.	1
TCSAJ K		Peripheral Instruction: takes next instruction from K where K is address supplied by CTS or PAC.	2
TS E	05.4	Basic Instruction: if c(A) is not an overflow quantity, copies c(A) into E and takes next instruction from I + 1 where I is location of TS E; if c(A) is a positive overflow quantity, copies c(A) into E, sets c(A) to +1, and takes next instruction from I + 2; if c(A) is a negative overflow quantity, copies c(A) into E, sets c(A) to -1, and takes next instruction from I + 2.	2
WAND H	10.3	Channel Instruction: AND's c(H) with c(A); stores result in H and A; takes next instruction from I + 1 where I is location of WAND H.	2
WOR H	10.5	Channel Instruction: Inclusively OR's c(H) with c(A); stores result in H and A; takes next instruction from I + 1 where I is location of WOR H.	2
WRITE H	10.0	Channel Instruction: copies c(A) into H; takes next instruction from I + 1 where I is location of WRITE H.	2
XCH E	05.6	Basic Instruction: exchanges c(A) with c(E); takes next instruction from I + 1 where I is location of XCH E.	2

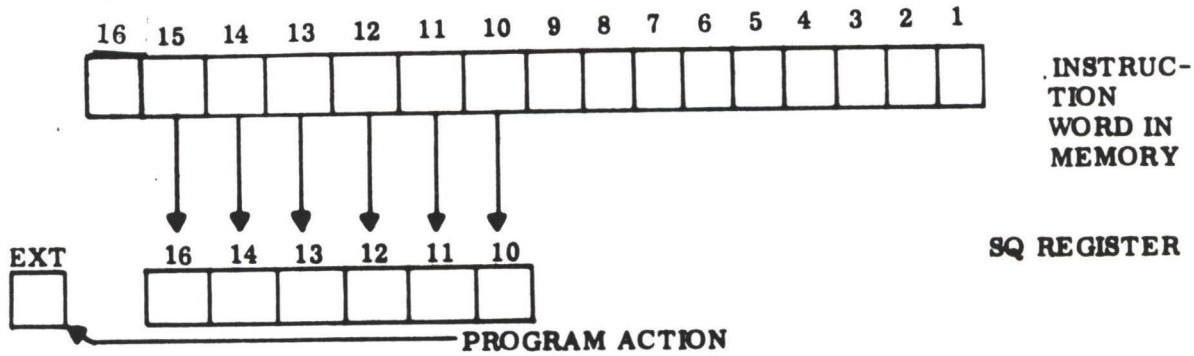


Figure 9-21. Memory to SQ Register Transfer

The three bit order code in the memory basic instruction words has a capability of uniquely defining eight operations: To increase the number of operations defined by the SQ register, bit EXT (extend) is made a 1 or 0 under program control, therefore, bits EXT, 16, 14 and 13 of the SQ register define sixteen operations.

Note the order codes in column 2 of table 9-2. These order codes are determined, in most cases, by the contents of the SQ register. Figure 9-21 shows how the order codes in table 9-2 are related to the actual contents of the SQ register. The instruction defined by figure 9-22 is TS E.

In table 9-2, the instructions can be categorized into three distinct groups by their listings in the order code column.

a. Those that list "None."

b. Those that list four digits to the right of the binary point.

c. Those that list two or three digits with the binary point written to the right of the second digit.

Group a contains the counter and peripheral instructions. There are no order codes associated with these instructions.

Group b contains the special instructions that are address dependent basic instructions. Their order codes are, in part, determined by bits 1 through 12. Those special instructions with no digits to the left of the decimal point can be combined with any basic instruction order code. Those with digits to the left of the decimal point are combined with that basic instruction whose order code appears to the right of the decimal point.

Group c contains the basic, extracode and channel instructions, i.e., all the regular instructions with the exception of the special instructions. Also included in this group are the two interrupt instructions: these are <u>not</u> regular instructions.

Note that the instructions in this group may or may not have a digit to the right of the decimal point. When there is a digit to the right of the decimal point, it is determined by bits 11 and 12 or bits 10, 11 and 12 of the SQ register. When bits 11 and 12 are necessary to extend the order code field, their configuration is called a "quarter code." When bits 10, 11 and 12 are necessary to extend the order code field, their configuration is called an "eighth code." Table 9-3 shows the configuration of the various quarter and eighth codes associated with this group. Note that there are two ways of defining a zero or an even digit to the right of the decimal point. Observe instructions CCS E and TCF F in table 9-3. These instructions are identical if only the digits to the left of the decimal point are considered. There, two instructions can be distinguished, however, if bits 11 and 12 of the SQ register are observed. Note that the content of bit 10 in register SQ is irrelevant because only four cases have to be distinguished and, consequently, a quarter code is sufficient to define the necessary operation. Now observe the instruction in table 9-3 which have digits 1 and 0 to the left of the decimal point in the order code column. There are eight of these instructions and to differentiate between them, bits 10, 11 and 12 of the SQ register are necessary because eight cases must be differentiated. If just bits 11 and 12 were used, only four cases could be distinguished.

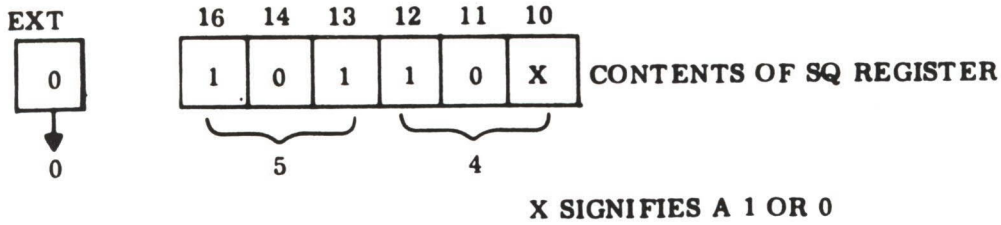


Figure 9-22. Order Code Determination

Basic instructions can be differentiated from extracode and channel instructions by the left hand digit of the order code. If bit EXT in the SQ register is a 0, then the left hand digit is a zero and the instruction is a basic instruction. If bit EXT is a 1, then the left hand digit is a one and the instruction is an extracode or channel instruction.

10.10.4.1.2 Involuntary Instructions. The involuntary instruction class contains two types of instructions - interrupt and counter. The interrupt instructions use the basic instruction word format just as the regular instructions do. However, the interrupt instructions are not entirely programmable. The contents of the order code field and the address field are supplied by computer logic rather than the program. The counter instructions have no instruction word format. Signals which function as a decoded order code specify the counter instruction to be executed and the computer logic supplies the address. The address for these instructions is limited to one of 29 counter locations in memory.

There are two interrupt instructions. One instruction initializes the computer when power is first applied and when certain program traps occur. The other interrupt instruction is executed at regular intervals to indicate time, receipt of new telemetry or keyboard data, or transmission of data by the computer. This interrupt instruction may be programmed to test the computer.

Table 9-3. Quarter and Eighth Codes

EIGHTH OR QUARTER CODES		SQ REGISTER BITS		
		12	11	10
EIGHTH	.0	0	0	0
QUARTER	.0	0	0	X
EIGHTH	.1	0	0	1
EIGHTH	.2	0	1	0
QUARTER	.2	0	1	X
EIGHTH	.3	0	1	1
EIGHTH	.4	1	0	0
QUARTER	.4	1	0	X
EIGHTH	.5	1	0	1
EIGHTH	.6	1	1	0
QUARTER	.6	1	1	X
EIGHTH	.7	1	1	1

X stands for a 1 or 0

There are several counter instructions. Two instructions will either increment or decrement by one the content of a counter location using the one's complement number system. Two other instructions perform the same function using the two's complement number system. Certain counter instructions control output rate signals and convert serial telemetry data to parallel computer data.

9.10.4.1.3 Peripheral Instructions. There are two types of peripheral instructions. One type deals with memory locations and the other type deals with channel locations. The peripheral instructions are not used when the computer is in the spacecraft. They are used when the computer is connected to peripheral equipment during subsystem and preinstallation system testing. The peripheral instructions are not programmable and are executed when all computer program operations have been forcibly stopped. These instructions are used to read and load any memory or channel location and to start the computer program at any specified address. The peripheral instructions and counter instructions are processed identically.

9.10.4.2 Interpretive Instructions. Interpretive instructions, a programmer's convenience and a means of saving memory storage area, must be interpreted under program control, converted to machine instructions and then executed as machine instructions. The coding into interpretive instructions of routines which contain double precision, triple precision, vector, and vector matrix operations results in a considerable saving in program storage area in fixed memory. This saving is achieved at the expense of computer operating speed; however, when operating in basic machine language the computer operates much faster than the equipment with which it interfaces. Since most of the PGNCS problems the computer is required to solve involve complex mathematical equations, the use of interpretive instructions for vector matrix algebra and complex differential calculus is a definite asset.

9.10.4.3 Instruction Data Flow. Examples of instruction data flow are illustrated in figures 9-23a, 9-23b, and 9-24a and 9-24b. Figures 9-23a and 9-23b indicate the detailed subinstruction flow diagrams necessary to produce an AD K instruction. Similar diagrams, figures 9-24a, and 9-24b are presented for the Resume instruction.

9.10.5 MEMORY. The CMC has erasable and fixed memories. The erasable memory can be written into and read out of; fixed memory can only be read out of. Erasable memory stores intermediate results of computations, auxiliary program information, and variable data supplied by external inputs from the PGNCS and other systems of the spacecraft. Fixed memory stores programs, constants, and tables. There is a total of 38,912, sixteen bit word storage locations in fixed and erasable memories. It should be noted that the majority of the memory capacity is in fixed memory (36,864 word locations). Both memories are magnetic core storage devices; however, the cores are used differently in each type of memory. It is assumed that the reader is familiar with the basic magnetic properties of a ferrite core as described by a square hysteresis curve. A core is a static storage device having two stable states. It can be magnetized in one of two directions by passing a sufficient current, I, through a wire which pierces the core. The direction of current determines the direction of magnetization. The core will retain its magnetization indefinitely until an opposing current switches the core in the opposite direction. Wires carrying current through the same core are algebraically additive. Sense wires which pierce a switched core will carry an induced pulse.

9.10.5.1 Erasable Memory Core Array. Erasable memory is arranged in a three-dimensional 32 x 64 x 16 core array for a total of 2048 16-bit storage locations. Each bit requires 1 core for its storage. The direction of magnetization determines whether a logic 1 or 0 is stored. The intersection of X planes and Y planes define 16-bit words. There are 64 X planes and 32 Y planes as indicated in figure 9-25.

Each core of erasable memory is threaded by four lines: an X selection line, a Y selection line, an inhibit line and a sense line. (See figure 10-29.) Each of the 64 X planes has a unique X selection line threading each core in that plane; each of the 32 Y planes has a unique Y selection line threading each core in that plane. Each of the 16-bit planes has a unique inhibit line threading each core in that plane and a unique sense line threading each core in that plane. Therefore, there is a total of 64 X selection lines, 32 Y selection lines, 16 inhibit lines and 16 sense lines in erasable memory. Readout of the erasable memory cores is destructive. If it is desired to retain the information in erasable memory, it must be written back into erasable memory. To read a core, currents = +1/2 I are simultaneously passed through the X and Y selection lines threading that core. Note the plus sign associated with the current. Current in the plus direction is a read current; current in the minus direc-

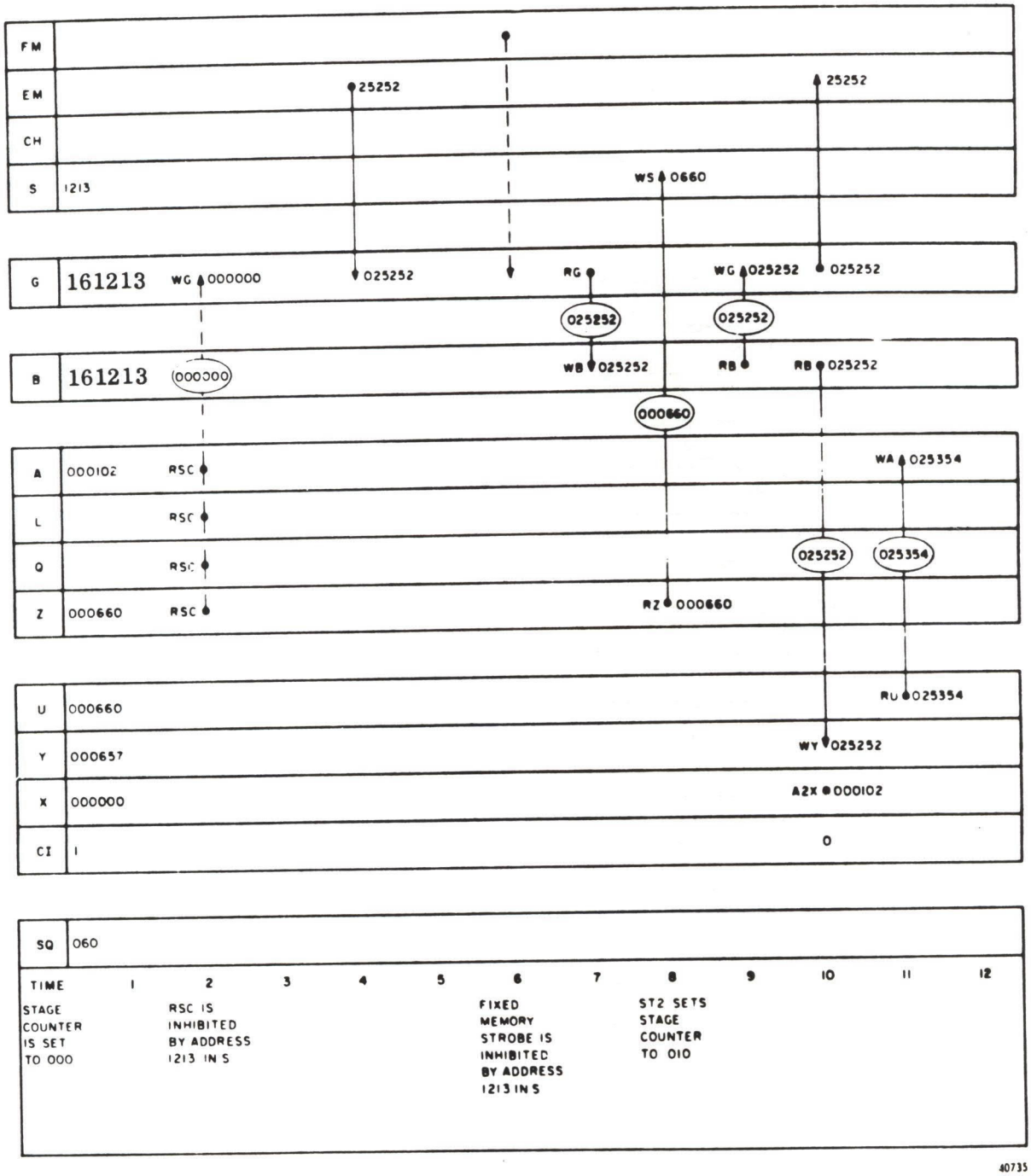


Figure 9-23b. **Subinstruction AD0, Data Transfer Diagram**

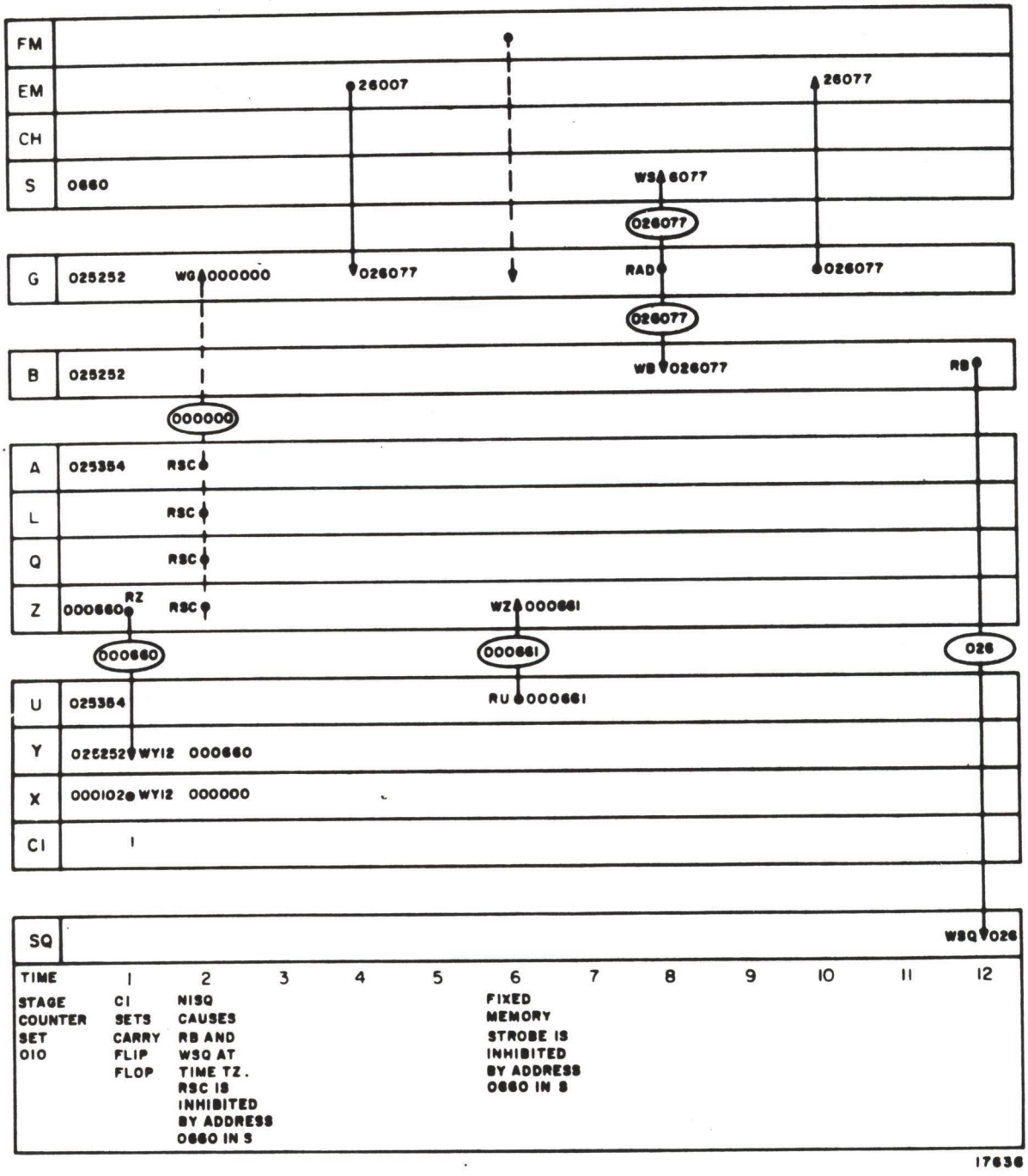


Figure 9-23a. Subinstruction STD2, Data Transfer Diagram

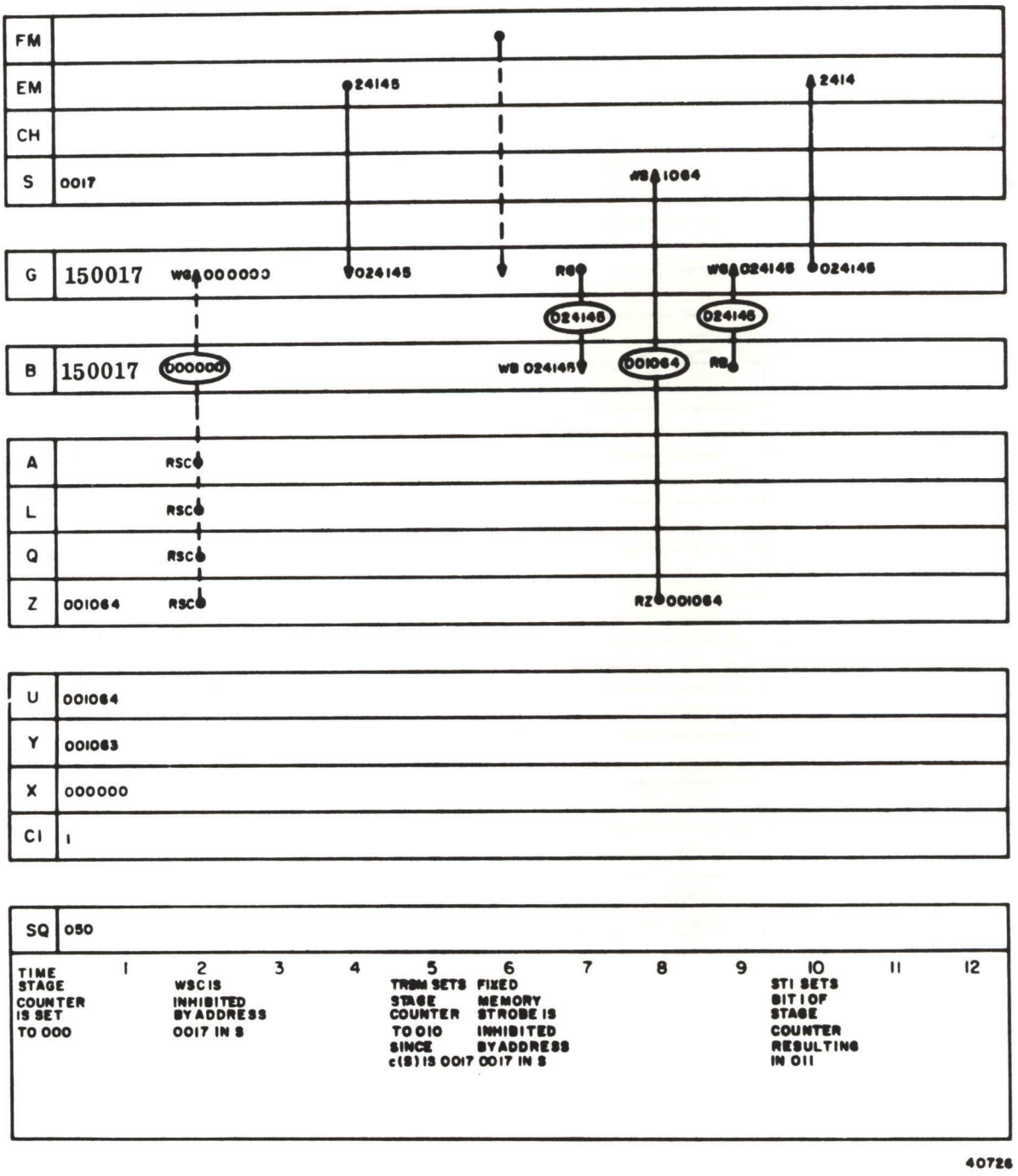


Figure 9-24a. Subinstruction NDX0, with Implied Address Code RESUME, Data Transfer Diagram

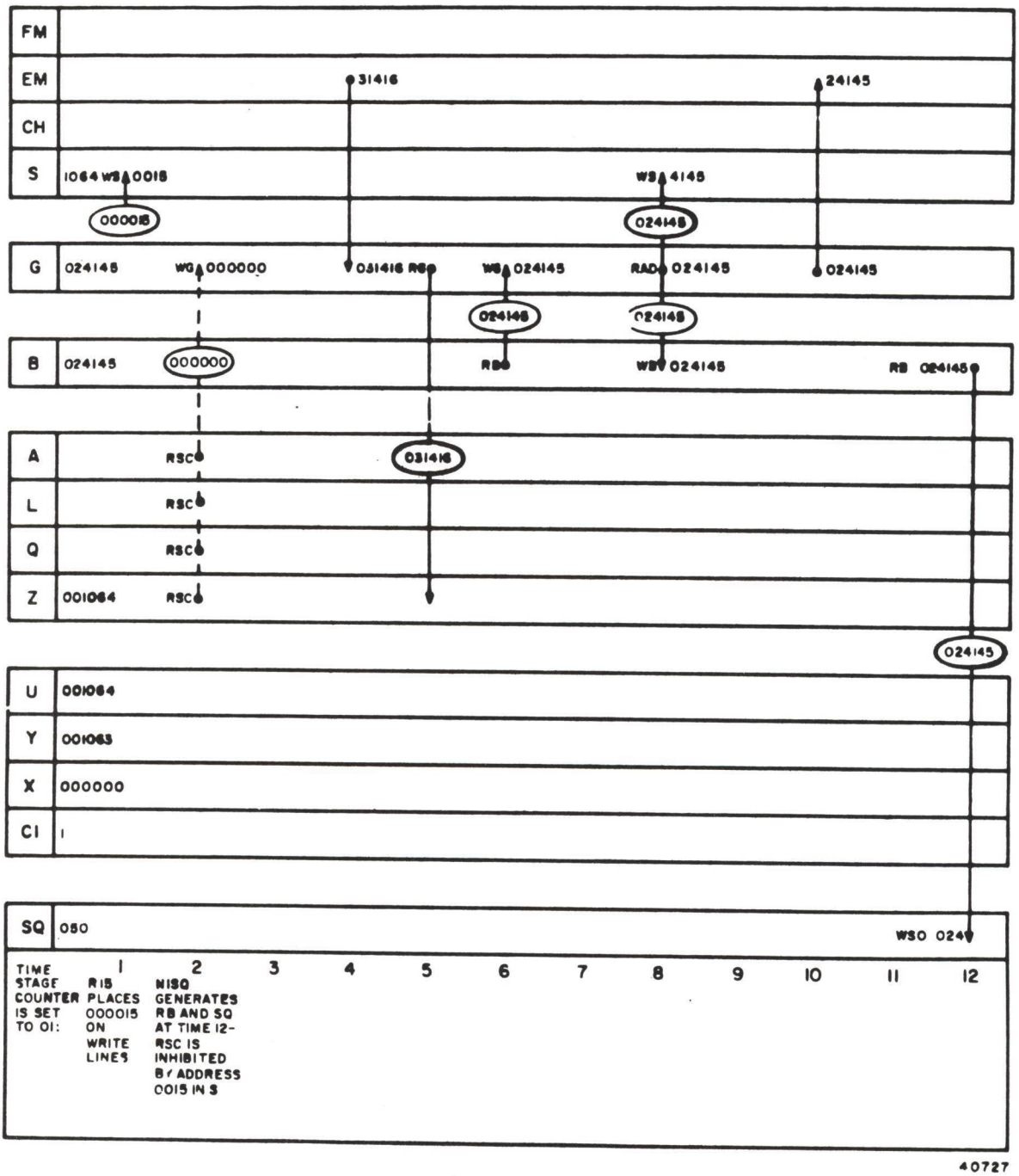


Figure 9-24b. Subinstruction RSM3, Data Transfer Diagram

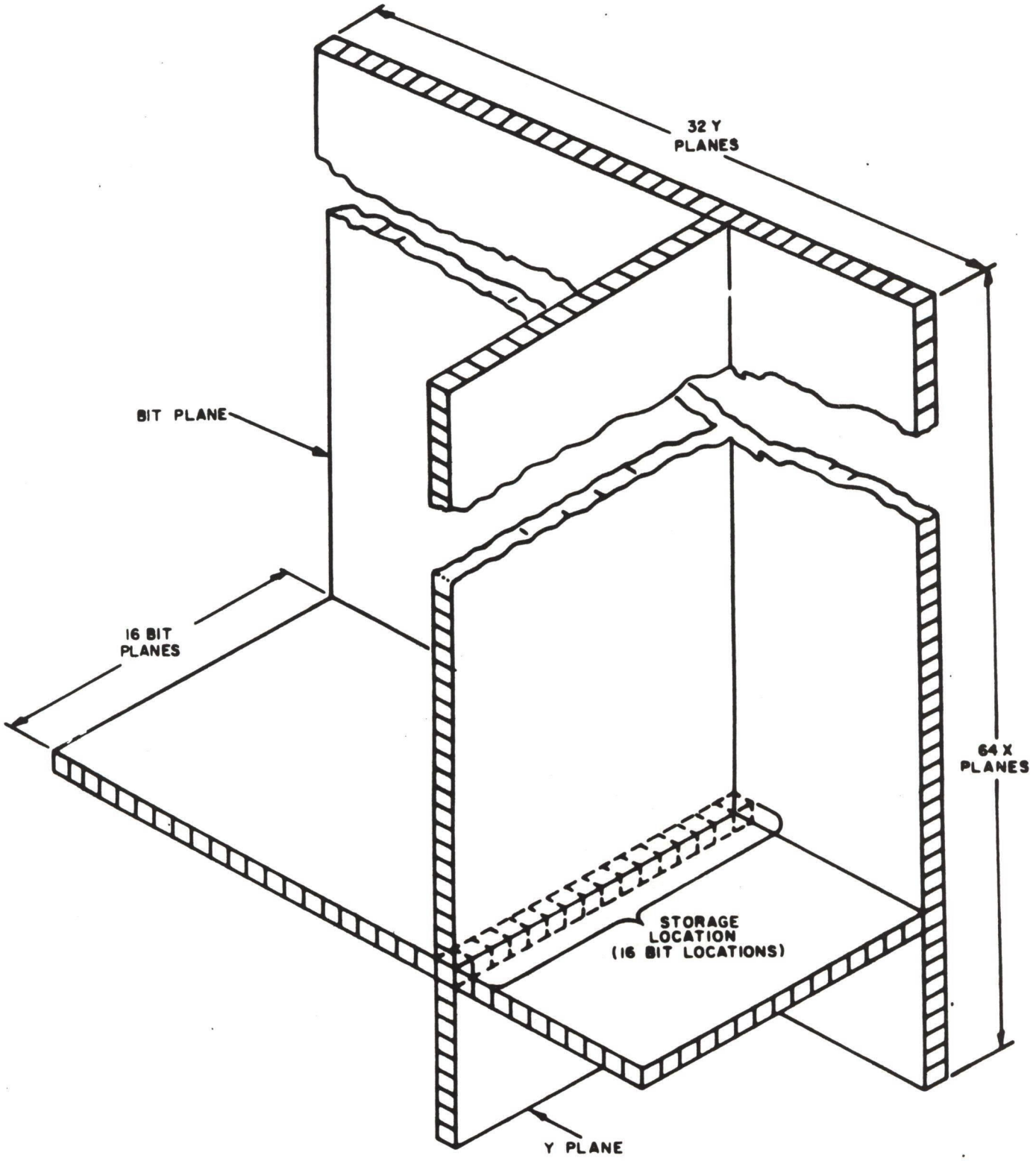
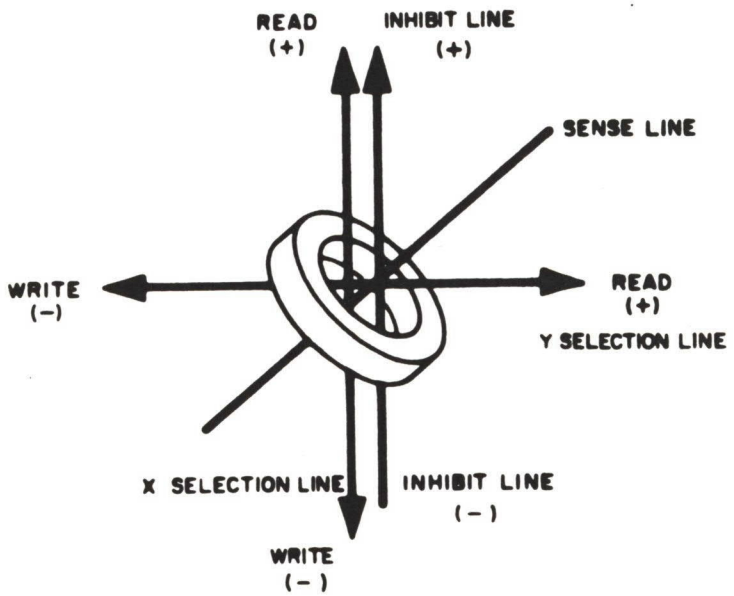


Figure 9-25. Erasable Memory Core Array



FUNCTION	X	Y	INHIBIT
WRITE ONE	$-\frac{1}{2} I_X$	$-\frac{1}{2} I_Y$	0
WRITE ZERO	$-\frac{1}{2} I_X$	$-\frac{1}{2} I_Y$	$+\frac{1}{2} I_I$
READ (CLEAR)	$+\frac{1}{2} I_X$	$+\frac{1}{2} I_Y$	0

Figure 9-26. Erasable Memory Core Threading

tion is a write current. Therefore, the total read current = +I. A read current = +I will try to switch the core to a zero state. If the core was in a one state prior to the generation of the read currents, the core will switch to a zero state and a pulse indicating a logic one will be induced on the sense line. If the core was in a zero state prior to the generation of the read currents, the core will remain in the zero state, no switching will occur, and no pulse will be induced on the sense line; this indicates that a zero was stored in the core. Note that the logic one stored in the core was destroyed after the core was read.

To write a 1 into the core, -1/2 I is passed through the X and Y selection wires. Thus, an attempt is made to switch the core to the 1 state. The two currents add and cause a switching to take place unless it is desired to write a 0 into the core. In this case, the inhibit line will carry +1/2 I opposing the X selection line current. Thus, the total current passing through the core is -1/2 I. This is not sufficient to switch the core, hence, a logic zero remains in the core. In order to write into a core, it must be cleared. This amounts to reading the core as described above.

The core array consists of 32,768 cores wired as described above. Figure 9-25 shows an X plane and Y plane intersecting to form a word. This intersection represents 16 cores, each core being a bit in the word. A single bit location is defined by the intersection of an X plane, a Y plane, and a bit plane. There are 16 bit planes, each containing 2048 bit locations, e.g., bit number five (5) of every word. Each core in a bit plane is threaded diagonally by a common sense line and vertically by a common inhibit line. (See figure 9-27.) Since a unique sense line threads all cores in a given bit plane, current will be induced into the line if the state of any core is changed. similarly, current through the inhibit line prevents every core in the bit plane from changing to the 1 state during the writing process. Each X plane and each Y plane is threaded by a common selection line. Choosing one of 32 Y and one of 64 X selection lines will choose 1 word of 2048 in erasable memory.

9.10.5.2 Addressing Erasable Memory. Erasable memory is divided into eight banks, each of which is capable of storing 256 (16 bit) words. These banks are named E-BANK 0 through E-BANK 7 (figure 9-28). E-BANK 0 contains 207 locations for general use; the other 49 locations have specific uses: 29 are used for counters, 12 have special functions, 8 are not used. The addresses of the eight unused locations are used to address registers in the central processor. All locations in E-BANK 1 through E-BANK 7 are utilized for general use.

E-BANK 0, 1 and 2 are referred to as unswitched erasable memory because all of their locations (as well as the addressable central processor registers) can be addressed by entering their addresses into register S without regard for what might be contained in register E BANK. E-BANK 3 through E-BANK 7 are referred to as switched erasable memory because all of their locations can be addressed only if the respective bank number is contained in register E BANK. Locations in unswitched erasable memory can also be addressed in the same manner as switched erasable memory if the proper bank number is contained in register E BANK and bits 9 and 10 of register S are logic 1's.

Figure 9-29 is a general block diagram of erasable memory and its associated circuitry. The first twelve bits of an instruction word constitute the address portion of the instruction. These bits are loaded into the S register and then applied to the address decoder. If either bit 11 or bit 12 or both is a logic 1, then fixed memory is being addressed and the erasable memory circuitry is not enabled, i.e., bit 11 and bit 12 of register S must be a logic 0 if erasable memory is to be addressed. If bits 9 and 10 of register S are logic 1's, then register E BANK is enabled and its three bits are also applied to the address decoder.

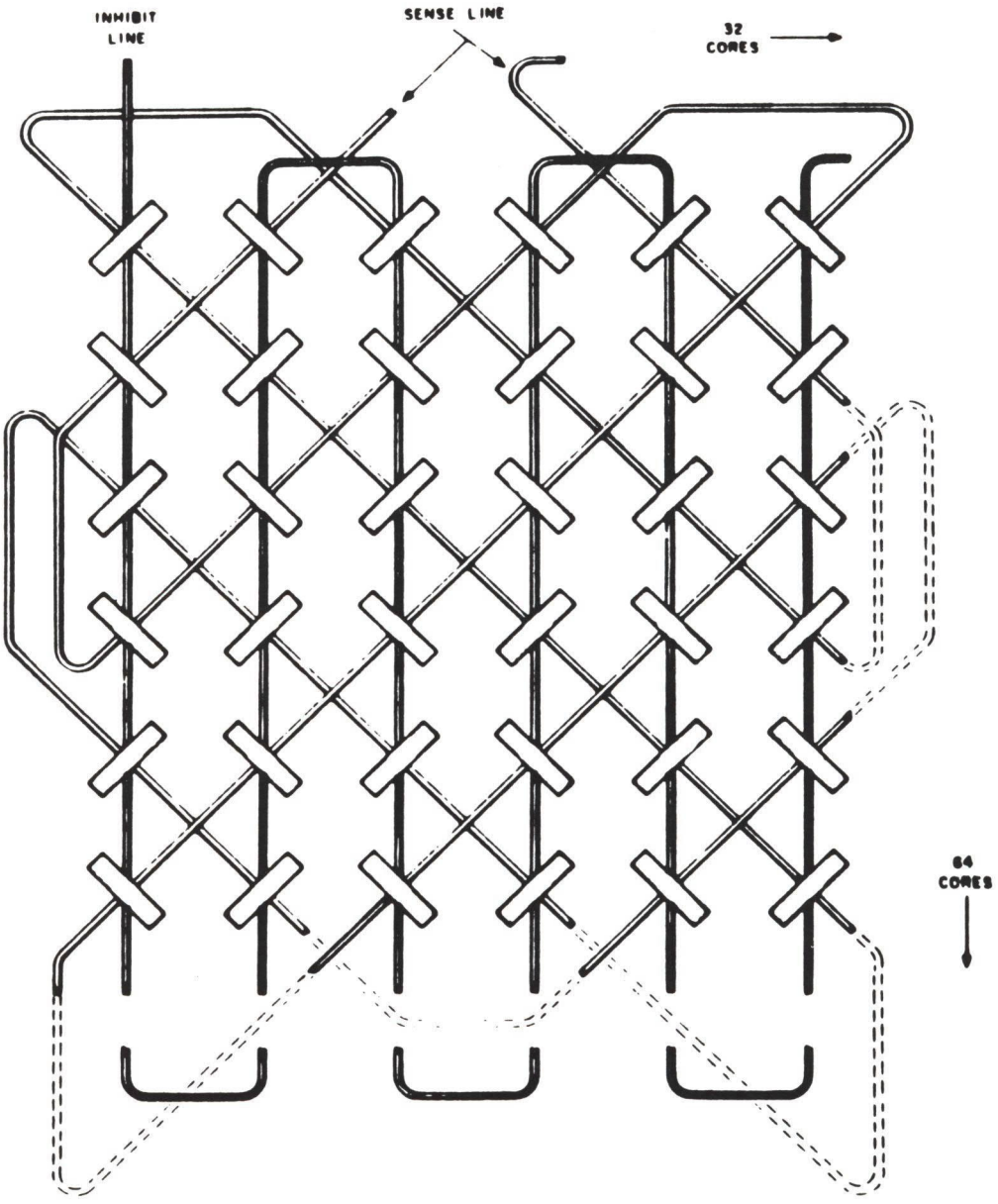


Figure 9-27. Bit Plane

	Register or Location Groups		C(E-Bank)			C(S)											
			11	10	9	12	11	10	9	8	7	6	5	4	3	2	1
Unswitched Erasable Memory	E-Bank 0	Central Processor	x	x	x	0	0	0	0	0	0	0	0	0	y	y	y
			0	0	0	0	0	1	1	0	0	0	0	0	y	y	y
		Special Locations	x	x	x	0	0	0	0	0	0	0	y	y	y	y	y
			0	0	0	0	0	1	1	0	0	0	y	y	y	y	y
		Counters	x	x	x	0	0	0	0	0	0	y	y	y	y	y	y
			0	0	0	0	0	1	1	0	0	y	y	y	y	y	y
		General Use	x	x	x	0	0	0	0	y	y	y	y	y	y	y	y
			0	0	0	0	0	1	1	y	y	y	y	y	y	y	y
	E-Bank 1		x	x	x	0	0	0	1	y	y	y	y	y	y	y	y
			0	0	1	0	0	1	1	y	y	y	y	y	y	y	y
	E-Bank 2		x	x	x	0	0	1	0	y	y	y	y	y	y	y	y
			0	1	0	0	0	1	1	y	y	y	y	y	y	y	y
Switched Erasable Memory	E-Bank 3		0	1	1	0	0	1	1	y	y	y	y	y	y	y	y
	E-Bank 4		1	0	0	0	0	1	1	y	y	y	y	y	y	y	y
	E-Bank 5		1	0	1	0	0	1	1	y	y	y	y	y	y	y	y
	E-Bank 6		1	1	0	0	0	1	1	y	y	y	y	y	y	y	y
	E-Bank 7		1	1	1	0	0	1	1	y	y	y	y	y	y	y	y

Y Means 0 or 1 as defined by address

X Means 0 or 1 which does not have an effect on addressing

Figure 9-28. Erasable Addressing

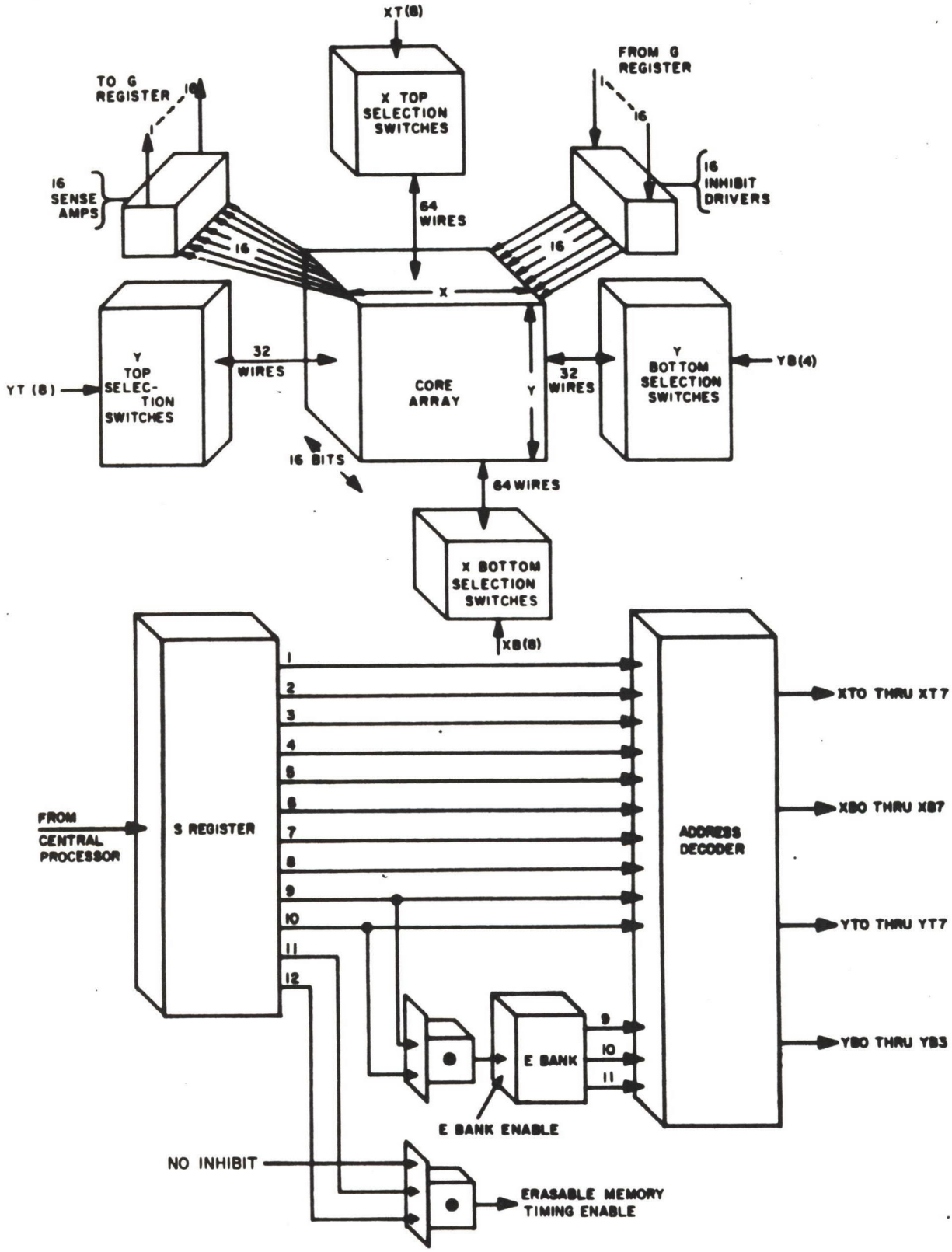


Figure 9-29. Erasable Memory Block Diagram

The address decoder generates signals XB0 through XB7, XT0 through XT7, YB0 through YB3 and YT0 through TY7. There are 64X selection and 32Y selection lines associated with erasable memory. A conbination of an X selection and a Y selection line determine a unique location (16 bit) in erasable memory. Figure 10-33 lists the outputs of the address decoder and the inputs required to generate a given output. Note that the address decoder can generate 64 separate combinations of an XT and an XB signal (8 x 8) and 32 separate combinations of a YT and YB signal (8 x 4). Note also that the XB's, XT's and YB's are a result of information located in the S register while the YT's are a result of information located in the E BANK or S registers.

Figures 9-31 and 9-32 are block diagrams of the circuitry involved in determining a particular X and Y selection line so that a unique location can be read from or written into erasable memory.

The following is an example of how a location in erasable memory is addressed. After the example, a detailed view of the selection switches involved will be presented.

Assume that the S register contains 001, 100, 000, 110 and that the E BANK register contains 10^{o}. Using figure 10-33 it is seen that signals XB6, XT0, YB0 and YT4 are logic 1's while the other listed signals are logic 0ls; note that bits 11 and 12 of the S register are logic 0's; therefore, erasable memory is addressed. Note also that bits 9 and 10 are logic 1's; therefore, register E BANK selects the YT signal. From figure 10-31 it can be seen that the location addressed is in E BANK 4. Assume that we wish to read the addressed location. In figure 9-32, only the switches that are selected by logic 1 outputs of the address decoder can pass current; therefore, the switches associated with outputs YB0 and YT4 can pass current while the others cannot. Consequently, a Y selection line has been selected. Also, in figure 9-32 , only the switches that are selected by the address decoder can pass current. Using the same process as above, it can be seen that an X selection line has been selected, therefore, a specific location (16 bit word) in erasable memory has been read out.

The core switches are not a part of the core array. They are, however, magnetic cores with storage properties. The cores act as small transformers as well as storage devices. The YB0 and YT4 pulses will be generated simultaneously. This causes the switch transistors to be momentarily biased on since their core windings are wired the same as the control pulses. Thus, current will flow in the read direction (from B+ to ground) through one of 32 Y selection wires. In addition, the cores are switched. The cores essentially remember the location which is being read so that the information can be rewritten back into the same location. The information would otherwise be destroyed. To write requires that current be passed in the opposite direction in the same selection wires. The RESET pulse will be routed to all core switches in the selection switch blocks. Only those switches just previously switches will be in a position to switch back to their original state. This will cause the transistors to be biased on, allowing conduction of current in the write direction from B+ to ground. Note that the diodes simply steer the current by acting as blocking diodes.

S Register *												
Signal	10	9	Signal	8	7	Signal	6	5	4	Signal	3	2	1
YT0	0	0	YB0	0	0	XT0	0	0	0	XB0	0	0	0
YT1	0	1	YB1	0	1	XT1	0	0	1	XB1	0	0	1
YT2	1	0	YB2	1	0	XT2	0	1	0	XB2	0	1	0
			YB3	1	1	XT3	0	1	1	XB3	0	1	1
						XT4	1	0	0	XB4	1	0	0
						XT5	1	0	1	XB5	1	0	1
						XT6	1	1	0	XB6	1	1	0
						XT7	1	1	1	XB7	1	1	1

E-Bank			
Signal	11	10	9
YT0	0	0	0
YT1	0	0	1
YT2	0	1	0
YT3Ψ	0	1	1
YT4Ψ	1	0	0
YT5Ψ	1	0	1
YT6Ψ	1	1	0
YT7Ψ	1	1	1

* Bits 11 and 12 are both assumed to be logic 0

Ψ Generated only if bits 9 and 10 of register S are both logic 1

Figure 9-30. Address Decoder

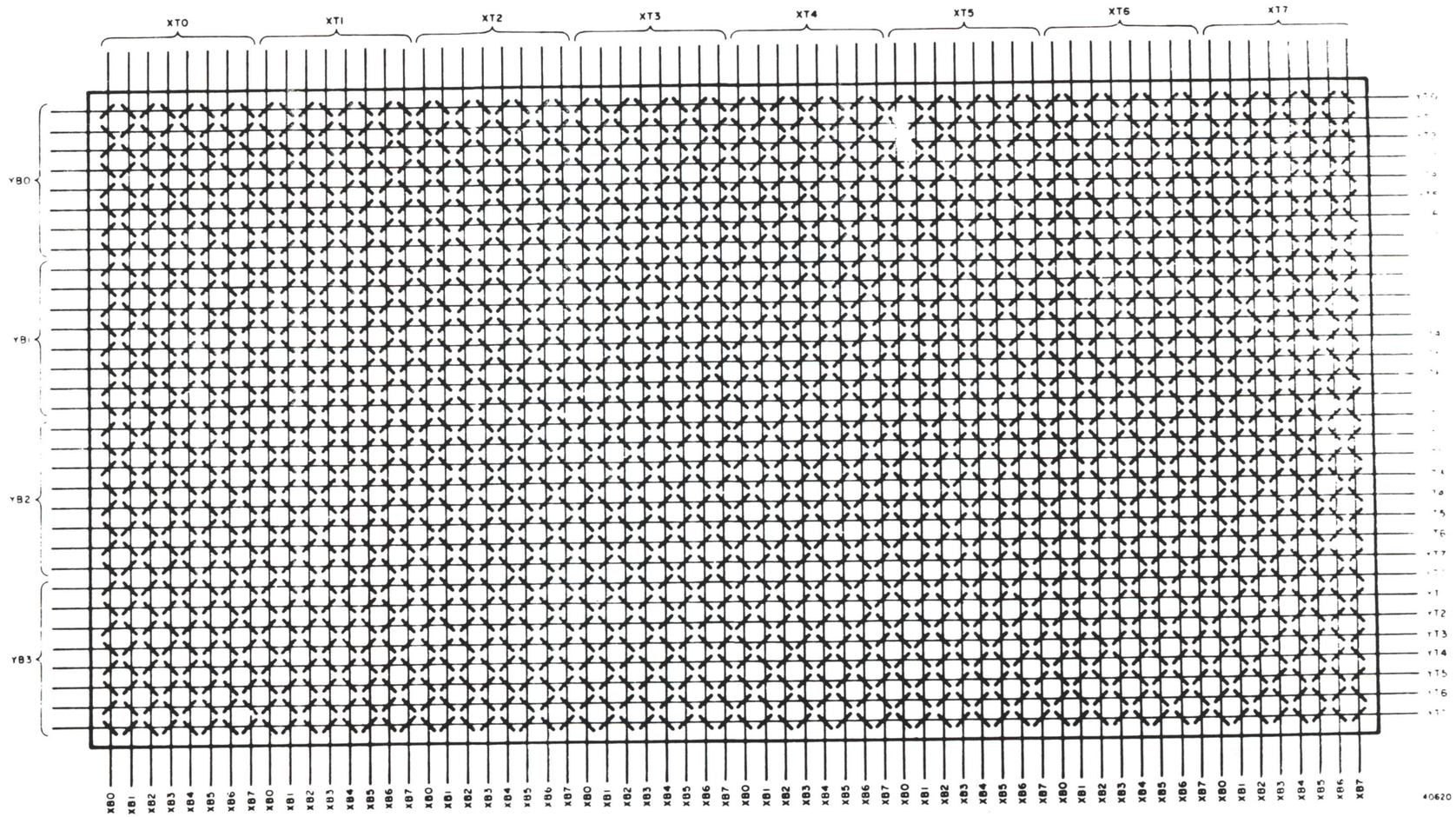
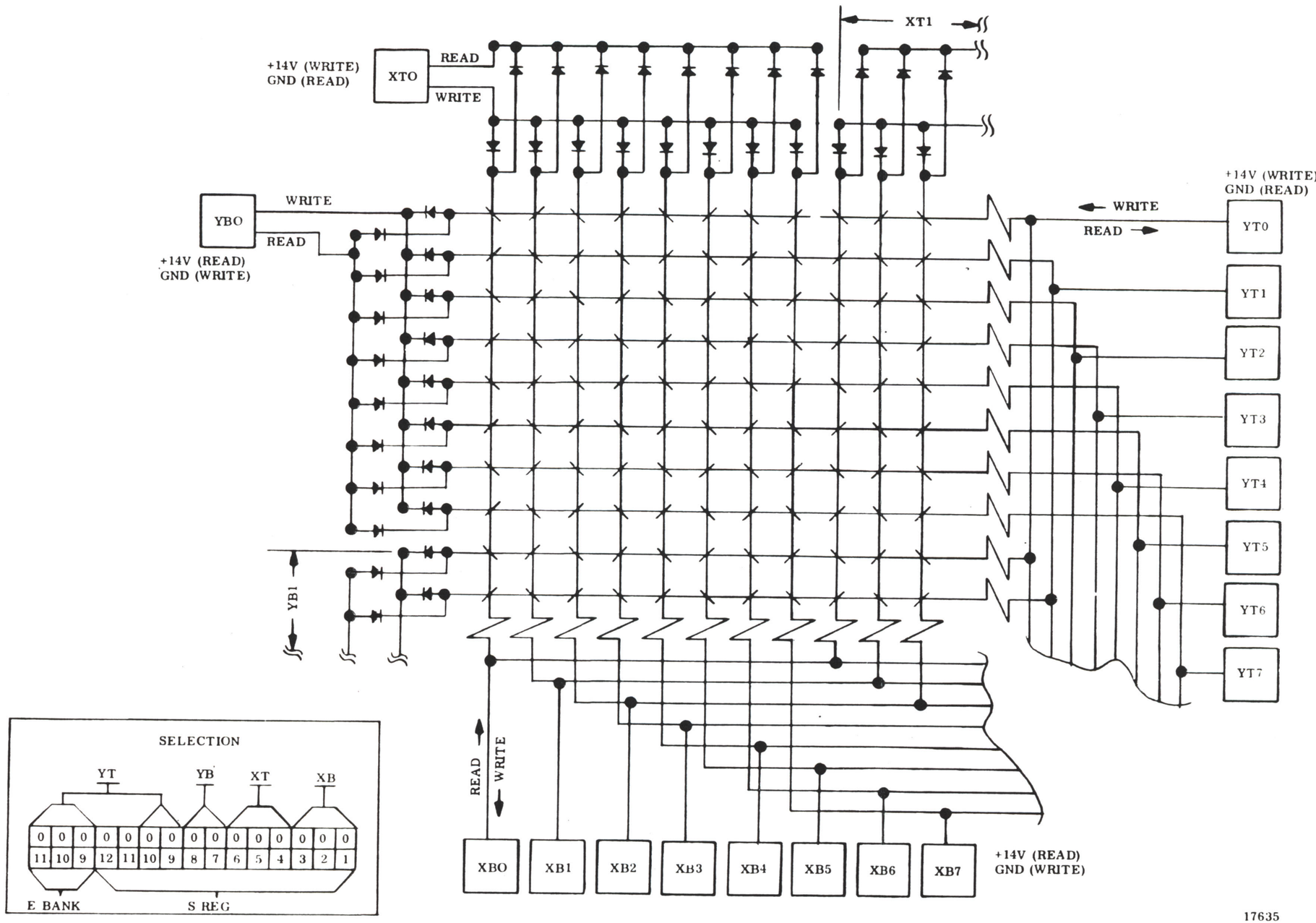


Figure 9-31. X and Y Coordinates



17635

Figure 9-32. Erasable Memory Selection

The X selection circuitry is similar to the Y selection circuitry discussed above. It is the selection of both an X and a Y wire that causes the selection of a 16 bit word. It should be mentioned that whether it is desired to read out or write into erasable memory, the selection procedure and operation of the core switches is the same. If it is desired to simply write a new word into a location, readout will occur, clearing the locations to all zeros. The information read out can be loaded into the G register where it will control the inhibit wires at the time the RESET pulse is issued. All bits in the selected word will switch to a 1 except those which have inhibiting current running through them. The sense amps receive and amplify pulses which are induced into the sense wires during readout. These amplifier pulses are applied to the G register where they are stored as logic ONE's. There are 16 sense amps, one for each bit in the selected word.

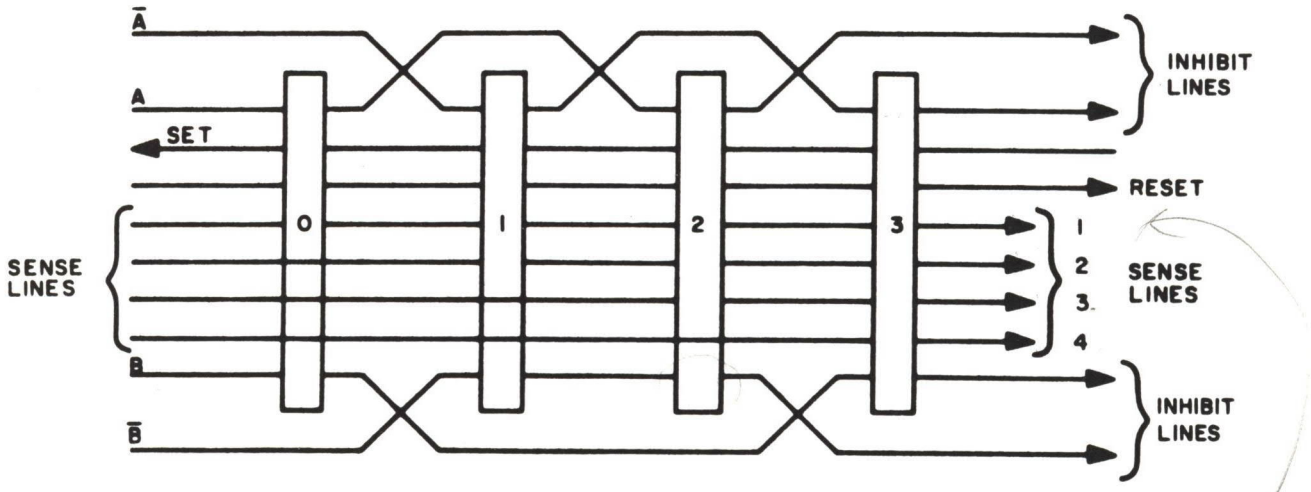
Erasable memory timing consists of several flip-flop circuits which produce the timing signals for erasable memory (see figure 10-36). These timing signals are produced in one memory cycle time (T01 through T12). At time $\overline{T03}$ a set signal (SETEK) is generated to enable the core selection switches to be addressed and read strobes (REX and REY) are generated to enable the selected data to be read out of memory. At time $\overline{T04}$, the sense strobe signal (SBE) is generated to enable the sense amplifiers to supply memory data to the G register. At time $\overline{T10}$ information is written into erasable memory using the reset, inhibit, and write strobes. The reset strobes (RSTKX and RSTKY) enable the reset drivers, thereby clearing the addressed memory location prior to writing in data. The inhibit strobe (ZID) enables the inhibit gates and the write strobes (WEX and WEY) enable data to be written into a particular memory address.

9-33

9.10.5.3 Fixed Memory Core Array. Figure ~~10-37~~ illustrates the use of the magnetic core in the rope core memory. This hypothetical model consists of 4 cores (edge views) labelled CORE 0 through CORE 3. Each core stores one 4-bit word, labelled WORD 0 through WORD 3. The bit configuration of each word is determined by whether the particular sense wires are routed through the hole or around the hole. The associated table in figure 10-37 shows the bit configuration that would be produced on the sense wires if the associated core were switched. Only one core could be switched at any one time. The problem becomes one of selecting the desired core to be switched. A two-bit address could be used to uniquely define any one of 4 cores.

The A and B inhibit lines (along with their complements) represent the bit configuration of the two-bit address. These wires are routed in and out of the 4 cores in order to select only one core.

Assume that it is desired to read WORD 2 from fixed memory. In figure 10-37, the direction of the arrows indicates the direction of current flow. Lines that are broken at the core pierce the core. Lines that are continuous at the core are routed around the core. Before reading a core, a current I is passed through the RESET line switching all cores into one state of magnetism. Currents are then passed through the SET line, $\overline{A}$ line and B line simultaneously in the direction shown. The current I in the SET line attempts to switch all cores back to the original state. However, the inhibit wires, $\overline{A}$ and B, carry current = 1/2 I in the opposite direction in order to oppose the SET current and cancel its effects. All cores except one will receive inhibiting current.



	INHIBIT LINES				SENSE LINES				
	A	Ā	B	B̄	4	3	2	1	
CORE 0		X		X	0	0	0	1	WORD 0
CORE 1	X		X		0	0	1	1	WORD 1
CORE 2		X	X		0	1	1	1	WORD 2
CORE 3	X			X	1	1	1	1	WORD 3

X INDICATES INHIBIT LINES NECESSARY TO ENABLE CORE SWITCHING

Figure 9-33. Sample Rope Core

Note that CORE 2 is not pierced by the $\overline{A}$ or B inhibit line. Therefore, CORE 2 changes state. This changing state produces a changing flux field which induces an electrical pulse in all sense lines which pierce the hole. Thus, SENSE lines 1, 2, and 3 will carry a momentary pulse while SENSE line 4 carries no pulse. The pulses represent logic ONE's while the lack of pulses represents a logic ZERO. The table included in figure 9-33 shows how the inhibit lines are wired to inhibit certain cores from switching. Note that if current is assumed to flow in A, it cannot flow in $\overline{A}$. The same idea applies to B and $\overline{B}$. Additional cores could be added to this model by increasing the number of inhibit lines.

9.10.5.4 Addressing Fixed Memory. Fixed memory consists of three ropes (R, S, T). Each rope is divided into two rope modules; B1 and B2, B3 and B4, B5 and B6, respectively. Each rope module is divided into six banks; therefore, each rope consists of 12 banks and, consequently, fixed memory consists of 36 banks. Each bank is capable of storing 1024 (16 bit) words; therefore, fixed memory has a storage capacity of 36,864 (1024 x 36) 16 bit words. Each core of fixed memory stores 12 16 bit words; therefore, there are 3072 (36,864 ÷ 12) cores in fixed memory. Note that 36 does not divide into 3072 evenly. This means that a given core is not completely associated with a particular bank. See table 9-4.

Table 9-4. Fixed Memory Composition

	/F MEM.	/ROPE	/MODULE	/CORE
ROPES	3	-	-	-
MODULES	6	2	-	-
WORDS	36,864	12,288	6,144	12
CORES	3,072	1,024	512	-
BANKS	36	12	6	-
STRANDS	72	24	12	12
MODULE AREAS	24	8	4	-
SENSE LINES	1,152	384	192	192
INHIBIT LINES	84	28	14	14

Fixed memory cannot be written into, and readout is nondestructive. Information stored in fixed memory is determined solely by the configuration of sense wires running through or around the cores; therefore, the stored information cannot be electrically altered; a rewiring of fixed memory would be necessary to change any programs, constant, or tables stored there.

Each rope module consists of 512 cores and 192 sense lines. The 512 cores are divided into four sections of 128 cores each. The 192 sense lines either thread or bypass each core, i.e., all 192 sense lines are associated with each core. The sense lines are divided into 12 strands of 16 lines each; therefore, a given strand defines one of the twelve words stored in each core, i.e., a given strand in a rope module is associated with 512 16 bit words. When a single core in a rope module is switched, (assume that all cores in the other 5 rope modules can be inhibited from switching), a pulse is induced on those m sense lines threading the core; no pulse is induced on the 192-m sense lines bypassing the core. If only one of the twelve strands is enabled, the location defined by the enabled strand and given core is selected. The method of switching a single core in a rope module is to select one of the four 128 core sections and then inhibit 127 of these cores from switching. The 127 cores are inhibited by information stored in bits 1 through 7 of the S register. Note that 7 binary bits have a capability of defining 128 different configurations.

The contents of register S, F BANK and F EXT are used to select locations in fixed memory for readout. Register S is a 12 bit register while F BANK and F EXT are 5 and 3 bit registers respectively. The addressing scheme employed is capable of defining 65,356 unique locations (table 9-5). The fixed memory address field consists of 64 banks. At the present time, only 36 banks (BANK 00_8 through BANK 43_8) are actually built into the CMC; the addressing scheme, however, allows for the addition of 28 extra banks (BANK 44_8 through BANK 77_8) if necessary.

The first 24 banks (BANK 00_8 through BANK 27_8) are referred to as F EXT-channel X. This group is further divided into two sub-groups: fixed-fixed memory and variable-fixed memory. Fixed-fixed memory consists of BANKS 02_8 and 03_8 and can be addressed by two different methods: (1) bits 12 and 11 of register S are 1 and 0 or 1 and 1 respectively, the contents of F BANK and F EXT are irrelevent; (2) bits 12 and 11 of register S are 0 and 1 respectively, the contents of F BANK are 00010 or 00011, the contents of F EXT are irrelevant. It should be noted, therefore, that each location in fixed-fixed memory can be addressed by two different methods. Variable-fixed memory consists of BANKS 00_8, 01_8 and 04_8 through 27_8. These locations are addressed by the S and F BANK registers only: their locations are not addressed by the contents of F EXT.

BANKS 30_8 through 37_8 are referred to as F EXT-channel 0-3. The locations in these banks are addressed if the F EXT register contains 000_2 through 011_2, the F BANK register contains 11000_2 through 11111_2 and bits 12 and 11 of register S are 0 and 1 respectively. "Channel 0-3" is so designated because 000_2 through 011_2 is equivalent to 0_8 through 3_8.

There are four other F EXT-channel groups: F EXT-channel 4 through F EXT-channel 7. Note that the channel numbers are a result of the contents of the F EXT register, e.g., when F EXT contains $101_2 = 5_8$ then the locations in the F EXT-channel 5 group are addressed.

Register or Location Groups		FMA Octal Address	F EXT Register	F Bank Register	S Register
			7 6 5	16 14 13 12 11	12 11 10 9 8 7 6 5 4 3 2 1
Fixed-Fixed Memory	F-Bank 02	04000-05777	x x x x x x	x x x x x 0 0 0 1 0	1 0 y y y y y y y y y y 0 1 y y y y y y y y y y
	F-Bank 03	06000-07777	x x x x x x	x x x x x 0 0 0 1 1	1 1 y y y y y y y y y y 0 1 y y y y y y y y y y
F EXT Channel X Variable Fixed	F-Bank 00	00000-01777	x x x	0 0 0 0 0	0 1 y y y y y y y y y y
	F-Bank 01	02000-03777	x x x	0 0 0 0 1	0 1 y y y y y y y y y y
	F-Bank 04	10000-11777	x x x	0 0 1 0 0	0 1 y y y y y y y y y y
	F-Bank 05	12000-13777	x x x	0 0 1 0 1	0 1 y y y y y y y y y y
	F-Bank 06	14000-15777	x x x	0 0 1 1 0	0 1 y y y y y y y y y y
	F-Bank 07	16000-17777	x x x	0 0 1 1 1	0 1 y y y y y y y y y y
	F-Bank 10	20000-21777	x x x	0 1 0 0 0	0 1 y y y y y y y y y y
	F-Bank 11	22000-23777	x x x	0 1 0 0 1	0 1 y y y y y y y y y y
	F-Bank 12	24000-25777	x x x	0 1 0 1 0	0 1 y y y y y y y y y y
	F-Bank 13	26000-27777	x x x	0 1 0 1 1	0 1 y y y y y y y y y y
	F-Bank 14	30000-31777	x x x	0 1 1 0 0	0 1 y y y y y y y y y y
	F-Bank 15	32000-33777	x x x	0 1 1 0 1	0 1 y y y y y y y y y y
	F-Bank 16	34000-35777	x x x	0 1 1 1 0	0 1 y y y y y y y y y y
	F-Bank 17	36000-37777	x x x	0 1 1 1 1	0 1 y y y y y y y y y y
	F-Bank 20	40000-41777	x x x	1 0 0 0 0	0 1 y y y y y y y y y y
	F-Bank 21	42000-43777	x x x	1 0 0 0 1	0 1 y y y y y y y y y y
	F-Bank 22	44000-45777	x x x	1 0 0 1 0	0 1 y y y y y y y y y y
	F-Bank 23	46000-47777	x x x	1 0 0 1 1	0 1 y y y y y y y y y y
	F-Bank 24	50000-51777	x x x	1 0 1 0 0	0 1 y y y y y y y y y y
	F-Bank 25	52000-53777	x x x	1 0 1 0 1	0 1 y y y y y y y y y y
	F-Bank 26	54000-55777	x x x	1 0 1 1 0	0 1 y y y y y y y y y y
	F-Bank 27	56000-57777	x x x	1 0 1 1 1	0 1 y y y y y y y y y y
F EXT Channel 0-3	F-Bank 30	060000-061777	0 x x	1 1 0 0 0	0 1 y y y y y y y y y y
	F-Bank 31	062000-063777	0 x x	1 1 0 0 1	0 1 y y y y y y y y y y
	F-Bank 32	064000-065777	0 x x	1 1 0 1 0	0 1 y y y y y y y y y y

Table 9-5. Fixed Memory Addressing
(Sheet 1 of 3)

Register or Location Groups		FMA Octal Address	F EXT Register	F Bank Register	S Register
			7 6 5	16 14 13 12 11	12 11 10 9 8 7 6 5 4 3 2 1
F EXT Channel 0-3	F-Bank 33	066000-067777	0 x x	1 1 0 1 1	0 1 y y y y y y y y y y
	F-Bank 34	070000-071777	0 x x	1 1 1 0 0	0 1 y y y y y y y y y y
	F-Bank 35	072000-073777	0 x x	1 1 1 0 1	0 1 y y y y y y y y y y
	F-Bank 36	074000-075777	0 x x	1 1 1 1 0	0 1 y y y y y y y y y y
	F-Bank 37	076000-077777	0 x x	1 1 1 1 1	0 1 y y y y y y y y y y
F EXT Channel 4	F-Bank 40	100000-101777	1 0 0	1 1 0 0 0	0 1 y y y y y y y y y y
	F-Bank 41	102000-103777	1 0 0	1 1 0 0 1	0 1 y y y y y y y y y y
	F-Bank 42	104000-105777	1 0 0	1 1 0 1 0	0 1 y y y y y y y y y y
	F-Bank 43	106000-107777	1 0 0	1 1 0 1 1	0 1 y y y y y y y y y y
	F-Bank 44	110000-111777	1 0 0	1 1 1 0 0	0 1 y y y y y y y y y y
	F-Bank 45	112000-113777	1 0 0	1 1 1 0 1	0 1 y y y y y y y y y y
	F-Bank 46	114000-115777	1 0 0	1 1 1 1 0	0 1 y y y y y y y y y y
F EXT Channel 5	F-Bank 47	116000-117777	1 0 0	1 1 1 1 1	0 1 y y y y y y y y y y
	F-Bank 50	120000-121777	1 0 1	1 1 0 0 0	0 1 y y y y y y y y y y
	F-Bank 51	122000-123777	1 0 1	1 1 0 0 1	0 1 y y y y y y y y y y
	F-Bank 52	124000-125777	1 0 1	1 1 0 1 0	0 1 y y y y y y y y y y
	F-Bank 53	126000-127777	1 0 1	1 1 0 1 1	0 1 y y y y y y y y y y
	F-Bank 54	130000-131777	1 0 1	1 1 1 0 0	0 1 y y y y y y y y y y
	F-Bank 55	132000-133777	1 0 1	1 1 1 0 1	0 1 y y y y y y y y y y
	F-Bank 56	134000-135777	1 0 1	1 1 1 1 0	0 1 y y y y y y y y y y
	F-Bank 57	136000-137777	1 0 1	1 1 1 1 1	0 1 y y y y y y y y y y
F EXT Channel 6	F-Bank 60	140000-141777	1 1 0	1 1 0 0 0	0 1 y y y y y y y y y y
	F-Bank 61	142000-143777	1 1 0	1 1 0 0 1	0 1 y y y y y y y y y y
	F-Bank 62	144000-145777	1 1 0	1 1 0 1 0	0 1 y y y y y y y y y y
	F-Bank 63	146000-147777	1 1 0	1 1 0 1 1	0 1 y y y y y y y y y y
	F-Bank 64	150000-151777	1 1 0	1 1 1 0 0	0 1 y y y y y y y y y y
	F-Bank 65	152000-153777	1 1 0	1 1 1 0 1	0 1 y y y y y y y y y y
	F-Bank 66	154000-155777	1 1 0	1 1 1 1 0	0 1 y y y y y y y y y y
	F-Bank 67	156000-157777	1 1 0	1 1 1 1 1	0 1 y y y y y y y y y y

Table 9-5. Fixed Memory Addressing
(Sheet 2 of 3)

Register or Location Groups		FMA Octal Address	F EXT Register	F Bank Register	S Register
			7 6 5	16 14 13 12 11	12 11 10 9 8 7 6 5 4 3 2 1
F EXT Channel 7	F-Bank 70	160000-161777	1 1 1	1 1 0 0 0	0 1 y y y y y y y y y y
	F-Bank 71	162000-163777	1 1 1	1 1 0 0 1	0 1 y y y y y y y y y y
	F-Bank 72	164000-165777	1 1 1	1 1 0 1 0	0 1 y y y y y y y y y y
	F-Bank 73	166000-167777	1 1 1	1 1 0 1 1	0 1 y y y y y y y y y y
	F-Bank 74	170000-171777	1 1 1	1 1 1 0 0	0 1 y y y y y y y y y y
	F-Bank 75	172000-173777	1 1 1	1 1 1 0 1	0 1 y y y y y y y y y y
	F-Bank 76	174000-175777	1 1 1	1 1 1 1 0	0 1 y y y y y y y y y y
	F-Bank 77	176000-177777	1 1 1	1 1 1 1 1	0 1 y y y y y y y y y y

x means 0 or 1 which does not have an effect on addressing.

y means 0 or 1 as defined by address.

Table 9-5. Fixed Memory Addressing
(Sheet 3 of 3)

Assume that bits 12 and 11 of register S are 0 and 1 respectively. F BANK 00_8 through F BANK 27_8 (F EXT-channel X) are determined by the contents of register F BANK (figure 9-34). F BANK 30_8 through 37_8 (F EXT-channel 0-3) are also determined by the contents of register F BANK (figure 9-34); note, however, that bit 7 of F EXT must be a 0. F BANK 40_8 through F BANK 77_8 are determined by the contents of the F EXT and F BANK registers (figure 9-35). This figure shows how F BANK 54_8 is addressed. Note that bits 16 and 14 of register F BANK are both 1's for all F BANK's except those in F EXT-channel X.

Table 9-5 lists all of the fixed memory addresses (FMA's). Assume again that bits 12 and 11 or reigster S are 0 and 1 respectively. The location in F BANK 00_8 through 27_8 are designated by a five digit octal number which is determined by the contents of the S and F BANK registers. Figure 9-36 illustrates how location 42113_8 is addressed. Note that bits 11 and 12 of register S are not used in determining address 42113_8. The locations in F BANK 30_8 through 37_8 are designated by a six digit octal number which is determined by the contents of the S and F BANK registers and bit 7 of the F EXT register. Figure 9-37. illustrates how location 074356_8 is addressed. The location in F BANK 40_8 through 77_8 are designated by a six digit octal number which is determined by the contents of the S, F BANK and F EXT registers. Figure 9-38 illustrates how location 151444_8 would be addressed.

Figure 9-39 is a block diagram of the selection logic connected with fixed memory.

A fixed memory address is selected by generating several selection signals:

a. Rope select.

b. Module select.

c. Set.

d. Reset.

e. Strand select.

f. Inhibit select.

Figure 9-34 F-BANK 00_8 Through 27_8 Determination

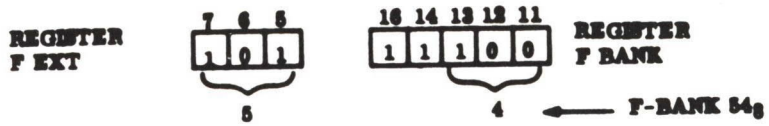


Figure 9-35. F-BANK 40_8 Through 77_8 Determination

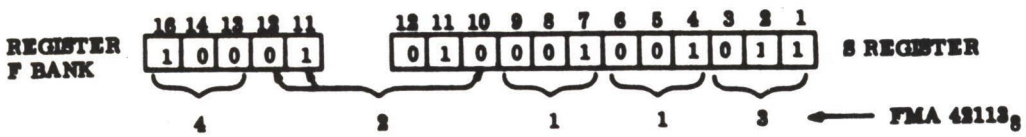


Figure 9-36. FMA 42113_8 Determination

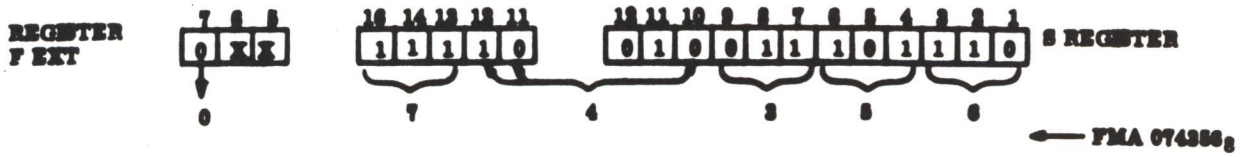


Figure 9-37. **FMA 074356$_8$ Determination**

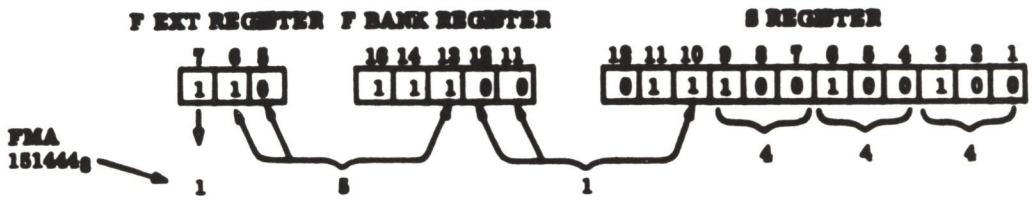


Figure 9-38 **FMA 151444$_8$ Determination**

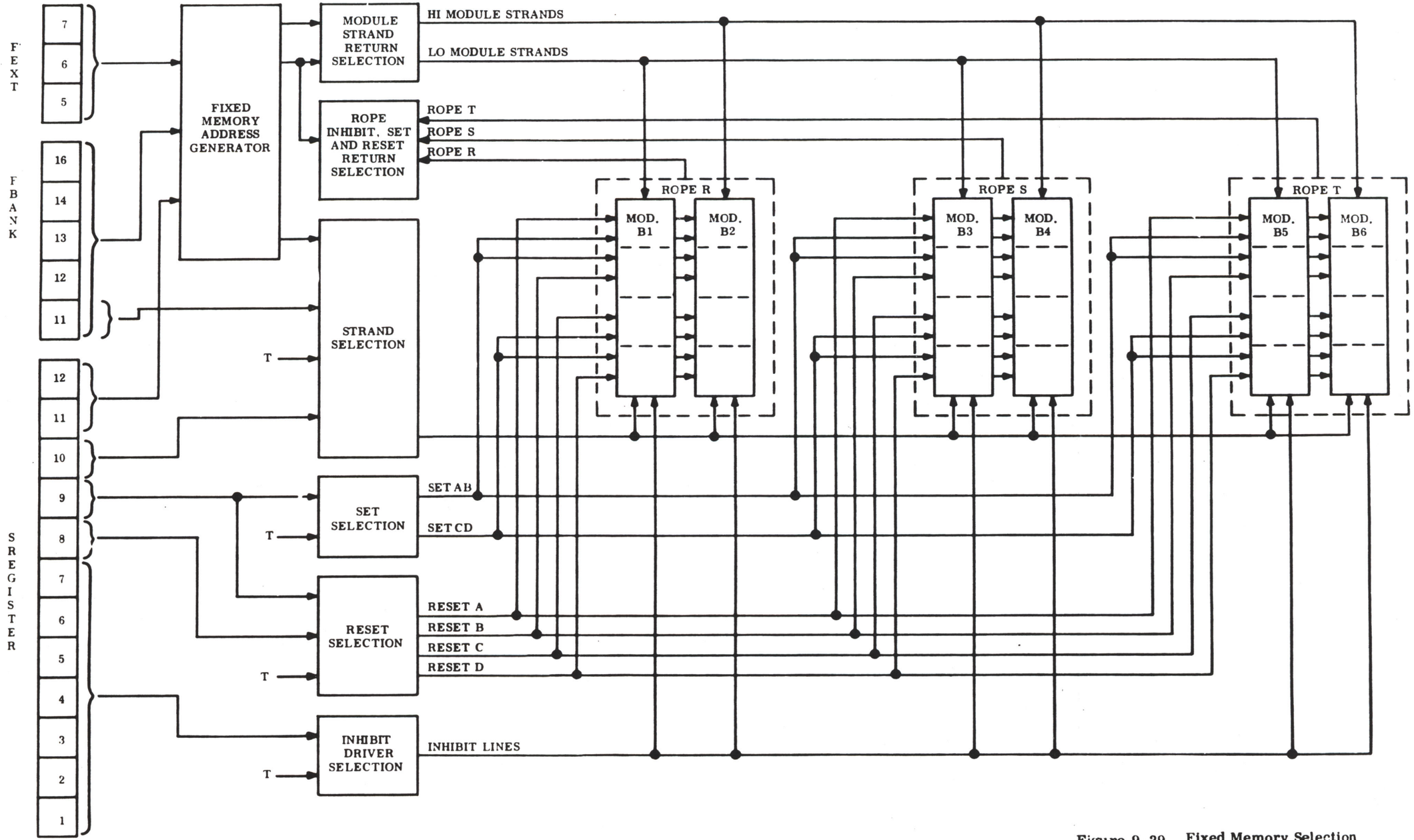


Figure 9-39. Fixed Memory Selection

9.10.6 PRIORITY CONTROL. Priority control consists of three separate and functionally independent areas: start instruction control, counter instruction control, and interrupt instruction control. See figure 9-40.

The start instruction control restarts the computer following a hardware or program failure. The counter instruction control updates the various counters in erasable memory upon reception of certain incremental pulses. The counter instruction control is also used during test functions to implement the display and load requests provided by the computer test set. The interrupt instruction control forces the execution of the interrupt instruction (RUPTOR) to interrupt the current operation of the computer in favor of a programmed operation of a higher priority.

9.10.6.1 Start Instruction Control. The start instruction control consists of the logic alarms processor and the start-stop generator. The logic alarms processor detects the presence of any one of several abnormal conditions that may occur within the computer and generates an alarm signal (ALGA) whenever any of these conditions exist. These abnormal conditions are:

a. RUPT lock.

b. TC trap.

c. Parity alarm (PALE)

d. Night watchman fail.

A RUPT lock alarm occurs if a program interrupt has been in progress too long (greater than 160 ms) or if an interrupt has not occurred within 160 ms. A TC trap alarm occurs if transfer control (TC) or transfer control to fixed memory (TCF) instructions do not occur within a period of up to approximately 15 milliseconds of each other, or if too many consecutive TC or counter incrementing (INKL) instructions are executed (continuous TC or INKL). A parity alarm (PALE) indicates that a word read out of fixed or erasable memory contains an even number of ones. A night watchman alarm occurs if address 00067 is not addressed by the computer within a

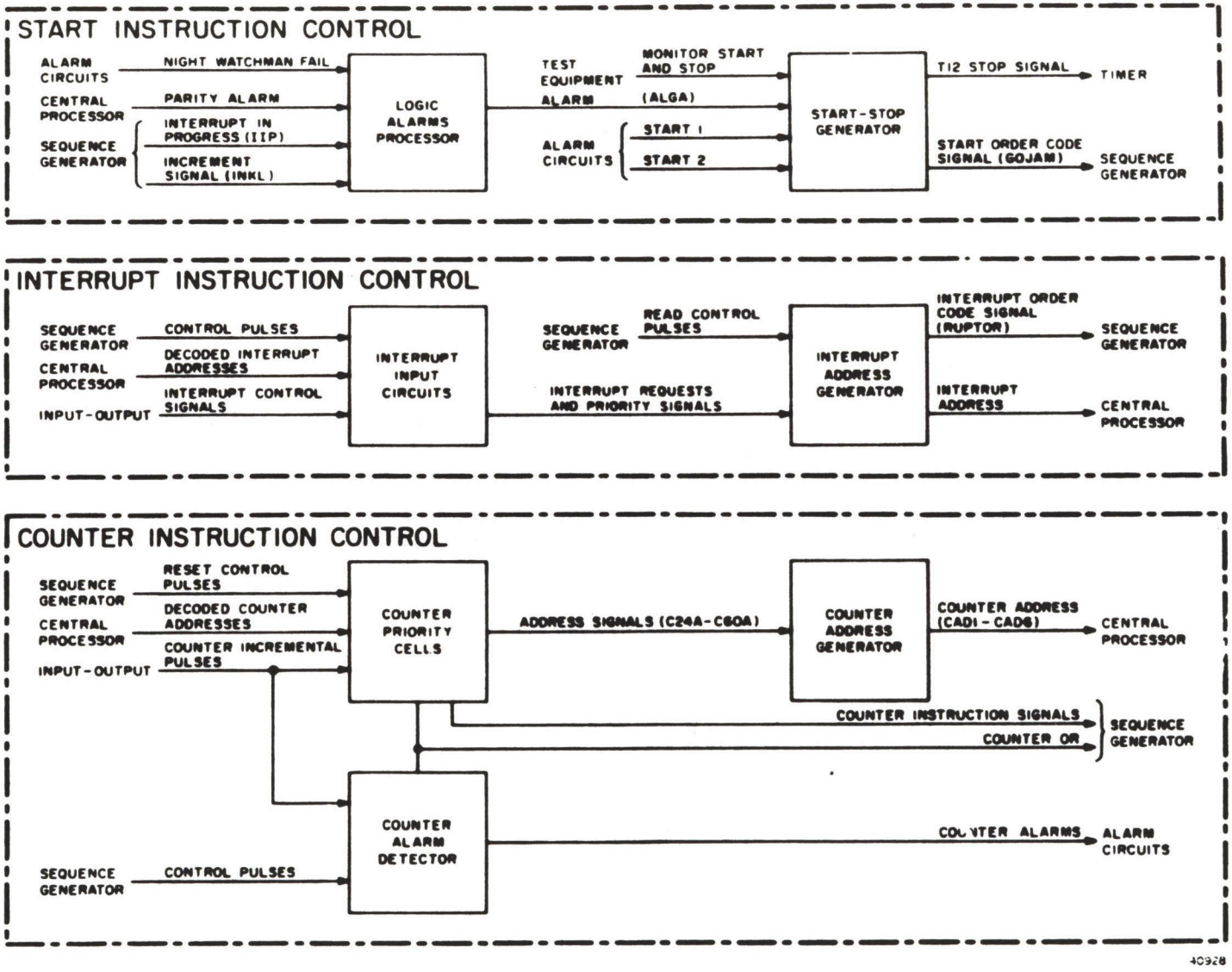


Figure 9-40. Priority Control Functional Block Diagram

period varying from approximately 0.65 seconds to approximately 1.9 seconds. If any of the above alarm conditions exist, alarm signal ALGA is generated by the alarm logic processor and is applied to the start-stop generator.

The start-stop generator generates signal GOJAM to restart the computer in response to the logic alarm signal ALGA at the next T12 time. The restart condition is indicated on the DSKY by the RESTART lamp being illuminated. The start-stop generator simultaneously produces a T12 STOP signal which inhibits the generation of timing pulses $\overline{T01}$ through $\overline{T12}$ in the timer until signal GOJAM has reset all critical circuits in the computer and forces the sequence generator to execute instruction GO. The detection of a computer power supply failure (START 1) or the detection of an oscillator failure (START 2) also causes the computer to restart. In addition, the computer can be started or stopped manually from the peripheral equipment.

9.10.6.2 Counter Instruction Control. A counter is updated when the counter instruction control receives a pulse for a given counter. This digital data consists of changes in velocity, changes in time, changes in gimbal angles, telemetry information, etc. It should be noted that some of these pulses come from within the CMC while others come from various other S/C systems. Counter instruction control may receive pulses for several counters at one time; the CMC, however, can only update one counter at a time. Therefore, the counter instruction control circuitry provides the following five functions.

a. It assigns a priority level to the inputs so that a logical processing order is established when more than one input is received simultaneously.

b. It is able to store an input until the CMC can process it.

c. It generates the address of that location in erasable memory which serves as a counter for that particular input.

d. It develops commands which are sent to the sequence generator to insure that the proper processing occurs. These signals are MINC, PINC, DINC, SHINC, SHANC, PCDU, MCDU and $\overline{INKL}$. Signal $\overline{INKL}$ inhibits the generation of the control pulses defined by the order code held in the SQ register. The sequence generator then develops those pulses necessary to perform a PINC, MINC, etc. instruction.

e. It resets the counter instruction control after the counter instruction has been completed so that other pulse inputs can be processed.

A counter interrupt takes one MCT to be processed. If the counter interrupt occurs during MCT (n), it will be processed during MCT (n + 1) provided, of course, that it is not inhibited by a counter interrupt of higher priority or that the CMC is not in the middle of an instruction.

Table 9-6 lists the 26 counters, their names, priorities, addresses and functions.

Table 9-6. Counter Interrupts

PRIORITY	ERASABLE MEMORY LOCATION (COUNTER)	NAME	REMARKS
1	0024	TIME 2	Stores most significant time word
2	0025	TIME 1	Stores least significant time word
3	0026	TIME 3	Time counter for program waitlist
4	0027	TIME 4	Time counter for T4 RUPT
5	0030	TIME 5	Time counter for thrust vector control
6	0031	TIME 6	Time counter for reaction control
7	0032	ICDU X	Counts ± Δθ pulses from X inertial CDU
8	0033	ICDU Y	Counts ± Δθ pulses from Y inertial CDU
9	0034	ICDU Z	Counts ± Δθ pulses from Z inertial CDU
10	0035	OCDU T	Counts ± Δθ pulses from optics trunnion CDU
11	0036	OCDU S	Counts ± Δθ pulses from optics shaft CDU
12	0037	PIPA X	Counts ± ΔV pulses from X PIPA
13	0040	PIPA Y	Counts ± ΔV pulses from Y PIPA
14	0041	PIPA Z	Counts ± ΔV pulses from Z PIPA
15	0042	BMAG X	Counts ± Δθ pulses from the X BMAG
16	0043	BMAG Y	Counts ± Δθ pulses from the Y BMAG
17	0044	BMAG Z	Counts ± Δθ pulses from the Z BMAG
18	0045	INLINK	Converts serial uplink data into parallel information

(Sheet 1 of 2)

Table 9-6. Counter Interrupts (cont)

PRIORITY	ERASABLE MEMORY LOCATION (COUNTER)	NAME	REMARKS
19	0047	GYRO D	Control pulse bursts which drive the gyros
20	0050	X CDU D	Controls pulse bursts which drive the X CDU
21	0051	Y CDU D	Controls pulse bursts which drive the Y CDU
22	0052	Z CDU D	Controls pulse bursts which drive the Z CDU
23	0053	TRUN CDU D	Controls pulse bursts which drive the optics trunnion CDU
24	0054	SHAFT CDU D	Controls pulse bursts which drive the optics shaft CDU
25	0055	EMSD	Supplies entry velocity to EMS
26	0057	OUTLINK	Converts parallel downlink information into serial data.

NOTE: The lower the priority number, the higher the priority.

(Sheet 2 of 2)

Figure 9-41 is a block diagram of the counter priority control. In this diagram, the inputs are divided into six groups. Each of these groups is associated with the particular counter interrupt instructions which they cause.

GROUP 1: PINC only

GROUP 2: DINC only

GROUP 3: PINC or MINC

GROUP 4: PCDU or MCDU

GROUP 5: SHINC or SHANC

GROUP 6: SHINC only

All of the counter interrupt parameters of each of these groups input to a circuitry block called the priority chain. Included in this block are flip-flops for the storage of the input pulses and the circuitry which establishes the various priorities of the inputs. When a particular input is received, the priority flip-flop associated with that input is set, and the outputs of all the other lower priority flip-flops are inhibited. The request for a particular counter instruction and the address of the appropriate location in erasable memory is then generated. The address generated during the processing of a counter interrupt is also used to reset the appropriate flip-flop in the priority chain. The signal $\overline{\text{INKL}}$ is generated as a result of $\overline{\text{CTROR}}$. $\overline{\text{INKL}}$ is also generated when instructions FETCH, STORE INOTRD or INOTLD are requested by the CTS.

The overall flow of information which occurs during the processing of a counter interrupt input is shown in figure 10-49. In this diagram it is assumed that a PIPA X input is being processed which could be either a plus or minus pulse and result in either a PINC or MINC instruction. These instructions require one MCT to execute.

When a plus or minus PIPA X pulse occurs, it is sent to the appropriate flip-flop in the counter interrupt priority circuits. Assuming plus or minus PIPA X pulse is the highest priority counter interrupt input that requires servicing at the end of the instruction during which it occurred, commands MINC or PINC, signal $\overline{\text{INKL}}$ and address 0037_8 are generated. During the next MCT, the sequence generator develops the control pulses for a MINC or PINC instruction. These instructions cause the six bit address referencing the proper counter in erasable memory, to be sent into the S register through the write amplifiers. Still under control of the MINC or PINC control pulse, the contents of the specified counter is read from erasable memory using the address contained in the S register. The content of the counter is set into the adder circuitry of the CMC where a binary 1 is added or subtracted from it. When the updating process is completed, the updated contents of the counter are stored back into its specified location in erasable memory. This is done under control of the S register which still contains the address of the counter. Having read, updated and written the contents of the counter back into memory, the address held in the S register is used to reset the flip-flop in the counter priority circuitry which requested the servicing. In this case, it is the X PIPA flip-flop. This completes the servicing of the PIPA X input which required one MCT to perform.

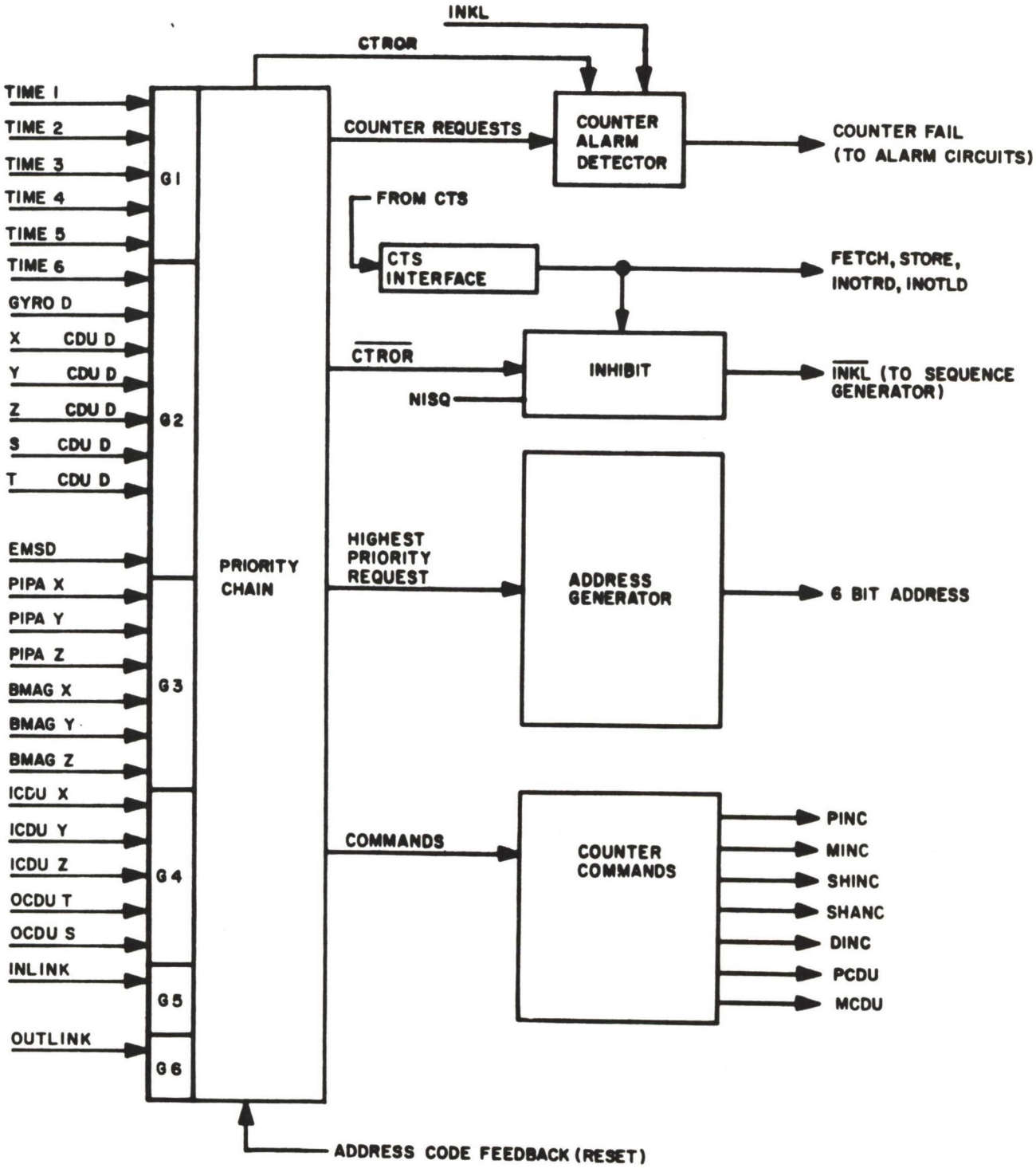


Figure 9-41. **Counter Priority Block Diagram**

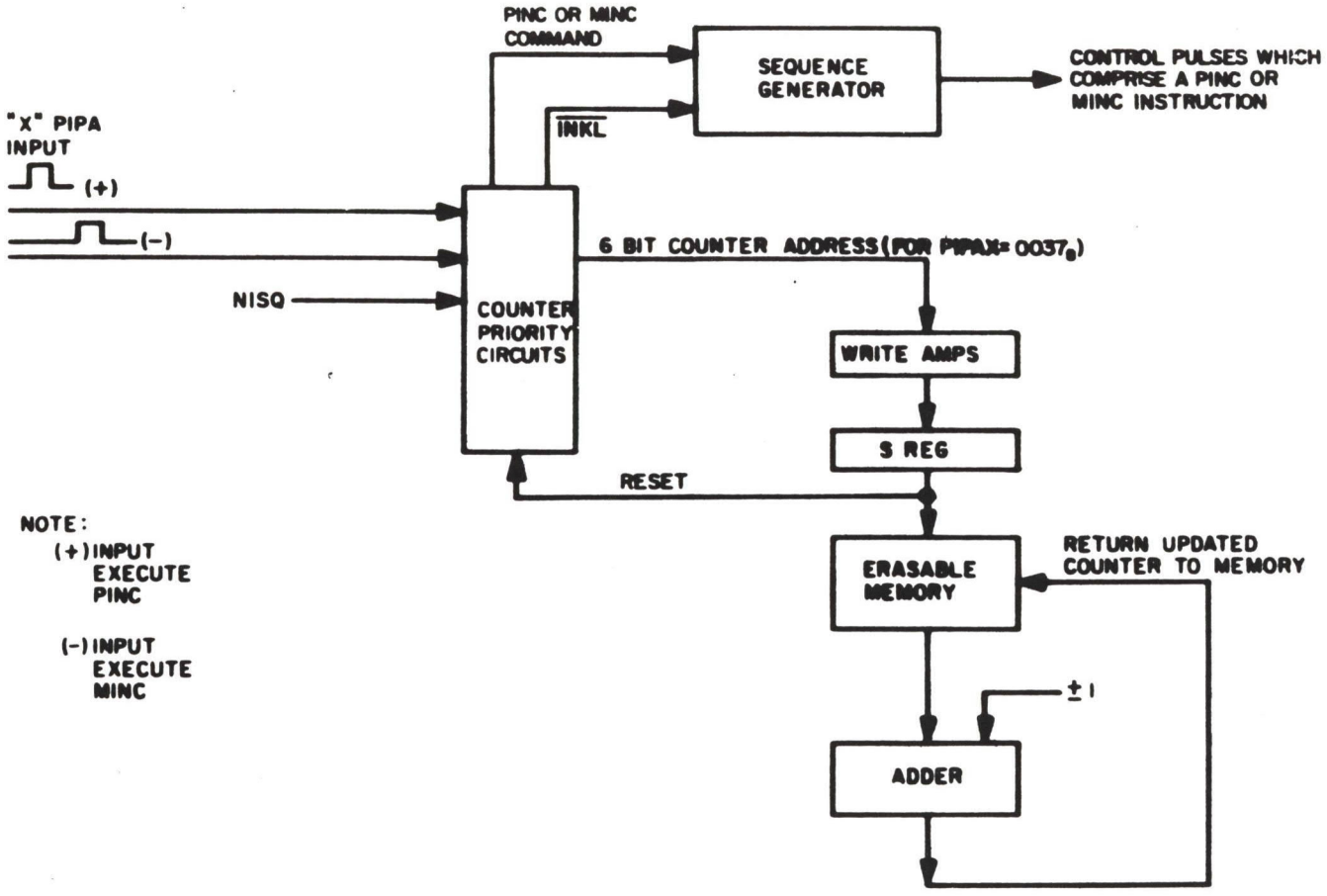


Figure 9-42. Counter Input Interface Flow Diagram

All of the counter interrupt inputs are operated on in essentially the same manner as the PIPA X input. Later in this section, the use of some other counter interrupt inputs and parameters will be presented as they are used in implementing other functions involving the CMC's interface functions.

9.10.6.3 Program Interrupt Priority Control. The T6 RUPT, T5 RUPT, T3 RUPT, T4 RUPT, KEYRUPT 1, KEYRUPT 2, UPRUPT, DOWNRUPT, and HAND CNTRL RUPT interrupt priority control routines are stored in memory and have that order of priority. When a program interrupt request is present, the interrupt priority control produces the address of the appropriate interrupt transfer routine and commands the sequence generator to execute the proper interrupt instructions. Program interrupts cause the suspension of the processing of a particular program and cause a particular routine to be executed. This execution depends upon which input to the program interrupt priority circuitry requires servicing. Table 9-7 lists the program interrupts, their names, initiating events and the actions initiated by their occurrence.

The signals used to request RUPT's 1 through 4 and 7 and 8 have a very short duration (about 1 μ second). However, the requests for RUPT's 5, 6 and 10 have a longer duration. If these long duration signals were used directly to request the processing of an interrupt routine, the request for the routine would still be present when the routine had been processed; therefore, the routine would be processed again. This is an undesirable condition and, consequently, "trap" circuits are used to prevent such a situation. The trap circuitry enables only one 1 μ second pulse to be developed per input request. A reset signal is then required to reset the trap circuitry before another request pulse can be developed. In the case of KEYRUPT 1 or 2, the reset signal is generated by the release of the pushbutton whose depression originally caused the program. In the case of RUPT 10, the reset signals are provided by bits 12, 13 and 14 of output channel 13.

Figure 9-43 is a block diagram of the program interrupt priority control circuitry and its associated inputs.

The request signals for the program interrupts are routed to request flip-flops. The outputs of the flip-flops are interconnected in such a manner that if a higher priority request is present, the outputs of all lower priority request flip-flops are inhibited. Whenever the request for the execution of an interrupt routine is honored, the starting address of the routine is generated and a signal RUPTOR is routed to the sequence generator. The honoring of an interrupt request is enabled at the end of the last MCT of an instruction.

When the request for a program interrupt is honored, the signal RUPTOR forces the order code for the RUPT instruction into the SQ register causing this instruction to be executed. This instruction enables the contents of the Z and B registers to be stored in the erasable memory and, using the generated address, forces control to the starting point of the appropriate interrupt routine. At the beginning of the interrupt routines, the contents of the A and Q registers are stored in the erasable memory. By storing the contents of the B, Z, A and Q registers, the next instruction, the next-next instruction's address, the data being operated on, and the return address of the interrupted program are stored. These quantities are restored in the central processor registers prior to returning control back to the interrupted program when the execution of the interrupt routine is completed.

Table 9-7. Program Interrupts

Interrupt Priority	Name	Initiating Event	Action Initiated
RUPT 1	T6 RUPT	Underflow of TIME 6 counter	Reaction control program termination (fine control)
RUPT 2	T5 RUPT	Overflow of TIME 5 counter	Reaction control program termination (coarse control)
RUPT 3	T3 RUPT	Overflow of TIME 3 counter	Time scheduling of programs to be processed
RUPT 4	T4 RUPT	Overflow of TIME 4 counter	Time scheduling of input/output control programs
RUPT 5	KEYRUPT 1	Depression of pushbutton on main panel DSKY	DSKY keyboard input processing
RUPT 6	KEYRUPT 2	Depression of pushbutton on navigation panel DSKY or depression of the MARK or MARK REJECT pushbutton on G&N indicator control panel	Navigation panel DSKY keyboard input processing or optics data processing for a MARK or MARK REJECT command.
RUPT 7	UPRUPT	A complete word parallelized in the INLINK counter	Uplink data processing
RUPT 8	DOWNRUPT	CMC receipt of telemetry end pulse	Downlink data processing
RUPT 10	HAND CONTROL RUPT	Manipulation of rotational or translational hand controller or throwing a thrust fail switch	Hand and minimum impulse controller input processing.

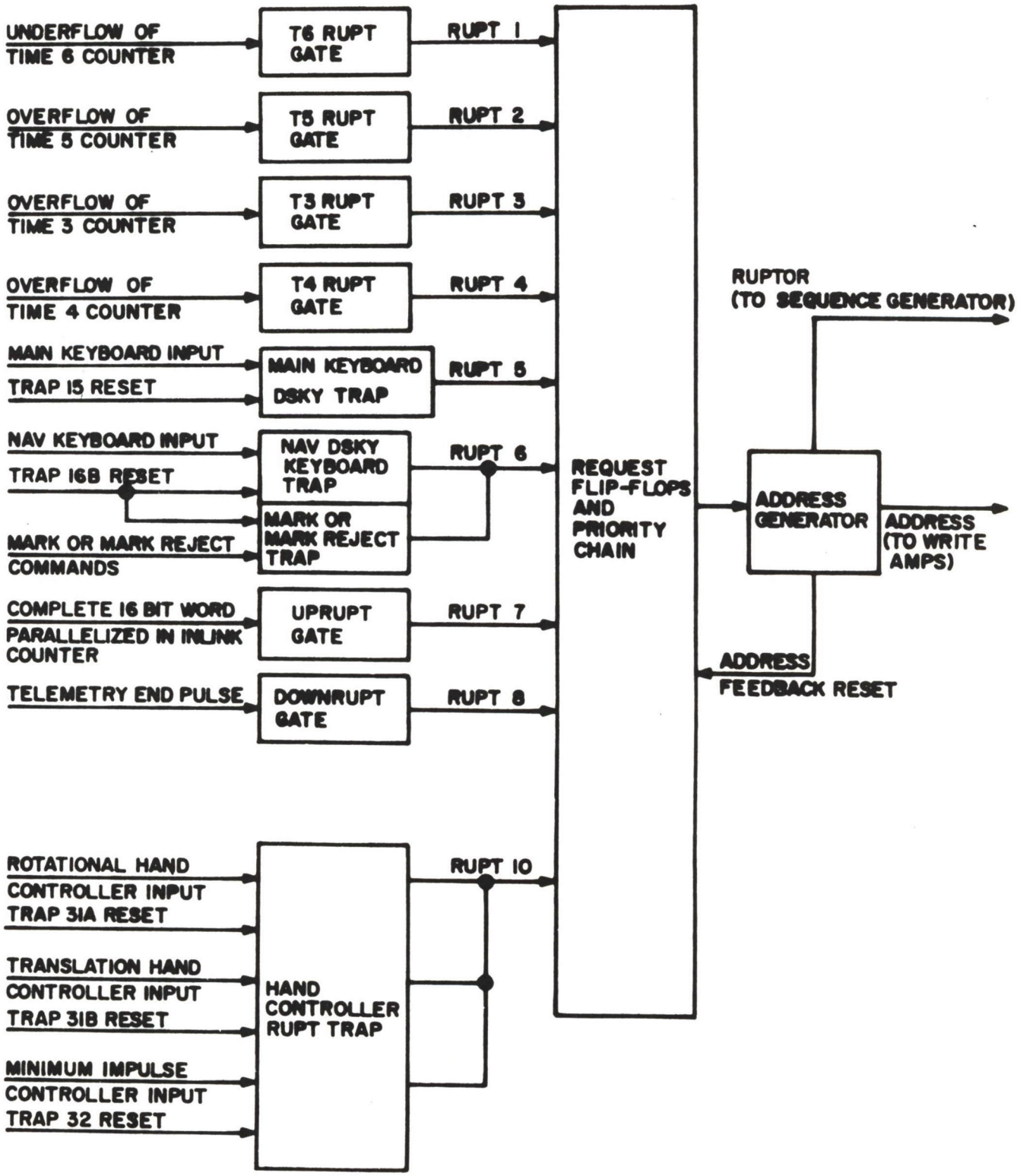


Figure 9-43. Program Interrupt Priority Control

9.10.6.3.1 T6 RUPT Routine. The reaction control program is terminated by the underflow of the TIME 6 counter. This counter is preset to a certain value by program control and then DINCed every 625 μ seconds if bit position 15 of output channel 13 contains a logic one. While the counter is being DINCed, the reaction control program is in process. When the TIME 6 counter underflows, the T6 RUPT routine is terminated.

9.10.6.3.2 T5 RUPT Routine. The T5 RUPT routine is terminated by the overflow of the TIME 5 counter (address 0030_8 in erasable memory). This counter, which is incremented every 10 ms through counter interrupt action (PINC's), is used to time the duration of the reaction control program. The TIME 5 counter is set to overflow minus the time delay required. Overflow terminates the T5 RUPT routine. When the reaction control program is completed, program control is returned to the interrupted program.

9.10.6.3.3 T3 RUPT Routine. The T3 RUPT routine is initiated by overflow of the TIME 3 counter. This counter, which is incremented every 10 ms through counter interrupt action, is used to time the time to initiation of a processing function. The counter is set to overflow minus the time delay required. When the desired time has elapsed, the TIME 3 counter will overflow which initiates the execution of the T3 RUPT routine. The T3 RUPT routine then routes control to the processing function specified to be performed at this time. When completed, control is returned to the T3 RUPT routine which returns control to the interrupted program.

9.10.6.3.4 T4 RUPT Routine. The T4 RUPT routine is executed whenever the TIME 4 counter overflows. Normally, this counter overflows every 120 ms. This counter, like the TIME 3, is incremented every 10 ms.

The T4 RUPT routine performs the following functions:

a. Controls the information displayed on the DSKY's.

b. Monitors and verifies the optics mode switching.

c. Monitors IMU temp, IMU turn-on, IMU, ICDU, and PIPA failures.

d. Monitors for gimbal lock indication.

e. Monitors IMU turn-on.

f. Monitors for downlink and uplink high rate failure indications.

9.10.6.3.5 KEYRUPT 1 Routine. This routine is initiated whenever a key of the main panel DSKY is depressed. The KEYRUPT 1 routine provides for CMC acceptance of the keycode input and initiation of the keycode processing.

9.10.6.3.6 KEYRUPT 2 Routine. The KEYRUPT 2 routine is initiated whenever a key of the navigation panel DSKY is depressed. It is also initiated if the MARK or MARK REJECT pushbutton on the G&N indicator control panel is pushed. The KEYRUPT 2 routine provides for CMC acceptance of the key code input and initiation of the key code processing. If this routine is executed as a result of a MARK REJECT, the associated data (time, gimbal angles, optical angles) is recorded or rejected by the CMC.

9.10.6.3.7 UPRUPT Routine. The UPRUPT routine is initiated whenever a binary 1 is shifted into bit position 16 of the INLINK counter through counter interrupt action. This counter provides a serial to parallel conversion of information sent to the CMC by uplink telemetry. This uplink information is in the form of key codes and the UPRUPT routine performs essentially the same function as KEYRUPT routines for keyboard inputs. The UPRUPT routine accepts the uplink word, checks the validity of the transmission and initiates the processing of the keycode input.

9.10.6.3.8 DOWNRUPT Routine. The DOWNRUPT routine is initiated by the telemetry end pulse from the downlink telemetry converter. This pulse occurs at either 10 or 50 times per second; therefore, the DOWNRUPT routine is executed at these rates. This routine selects the appropriate CMC data to be transmitted downlink and loads it into output channels 34 and 35. The information is then gated out of the CMC in a serial fashion.

9.10.6.3.9 HAND CNTRL RUPT Routine. This will described in the next section (CMC's input and output interface).

9.10.6.4 Alarm Detection Circuits. The alarm detection circuits consist of temperature, voltage, scaler, double frequency scaler oscillator, memory clamping and warning filter and integrator circuits. See figure 9-44.

The IMU stable member temperature is monitored through the temperature alarm circuit. Signal TEMPIN, an indication that the stable member temperature has exceeded its design limits, causes signal TMPCAU to be generated and routed to the DSKY for display (TEMP).

The voltage alarm circuit monitors the +28 vdc, +14 vdc and +4 vdc power and generates signal VFAIL for an out of tolerance or complete failure of any one of the power inputs. Signal VFAIL is conditioned by timing signals and routed to the priority control circuits as signal STRT1 provided it is not inhibited by the CTS signal MVFAL. Signal STRT1 is used in the start instruction control portion of priority control to generate GOJAM. Simultaneously, if the computer is in the standby mode, signal VFAIL and signal STNDBY cause the warning integrator to generate signals CMCWAR and AGCWAR. Signal CMCWAR is routed to a DSKY relay which energizes and causes the CMC warning indicator to illuminate on the caution and warning panel. Signal AGCWAR, a flip-flop output is routed to bit position 14 of channel 33. Signal MVFAIL is also available to the CTS for monitoring.

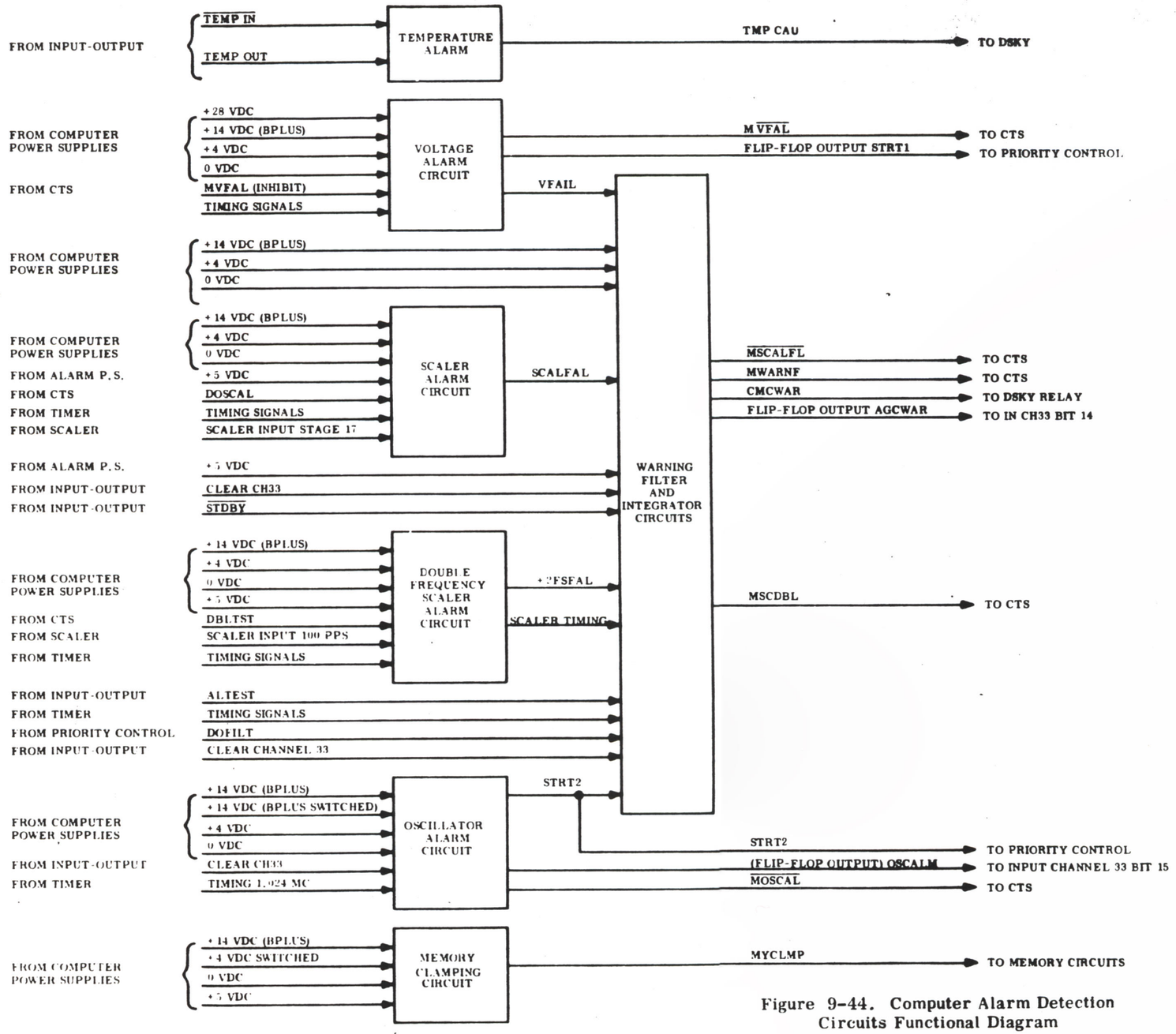


Figure 9-44. Computer Alarm Detection Circuits Functional Diagram

The scaler alarm circuit monitors stage 17 of the scaler and generates signal SCAFAL if stage 17 fails to produce pulses. Signal SCAFAL is used in the warning integrator directly to generate signals CMCWAR and AGCWAR.

Signal DOSCAL, from the CTS, is used to test the operation of the scaler alarm circuit. Signals MSCAFL and MWARNF are available from the warning integrator circuit for CTS monitoring. Signal MSCAFL indicates a scaler failure and MWARNF indicates a CMC warning condition.

The double frequency scaler alarm circuit monitors the 100 pps scaler output and generates signal 2FSFAL whenever the scaler stage is not operating properly. Signal 2FSFAL is used in the warning filter and integrator to generate CMCWAR and AGCWAR. Signal DBLTST, from the CTS is used to test the double frequency scaler alarm circuit. Signal 2FSFAL is also used in the warning filter to generate signal MSCDBL for CTS monitoring.

The oscillator alarm circuit monitors the computer oscillator and generates signals STRT2, OSCALM, and MOSCAL if the oscillator fails. Signal STRT2, is applied to the priority control circuits to generate signal GOJAM and also to a flip-flop to generate OSCALM and MOSCAL. Signal OSCALM is applied to channel 33, bit position 15 and is used internal to the computer. The flip-flop is reset with signal CCH33 (clear channel 33). Signal MOSCAL is available to the CTS for monitoring purposes.

The memory clamping circuit monitors the +4 vdc power and generates signal MYCLMP upon loss of this power. Signal MYCLMP is applied to the memory circuits to inhibit any access to memory.

The warning filter circuit performs logic gating for signals:

a. $\overline{\text{VFAIL}}$ and $\overline{\text{STNDBY}}$

b. 2FSFAL

c. DOFILT

d. ALTEST

Timing signals allow only one of these signals to be gated to the integrator at a time.

The warning integrator circuit monitors the output of the warning filter and generates signal CMCWAR and AGCWAR if 5 successive pulses are received from the warning filter. Signals CMCWAR and AGCWAR will also be generated if signal SCALFAL is present. Signal AGCWAR is a flip-flop output which is applied to channel 33, bit position 14 for use internal to the computer. Signal CMCFAL is routed to the DSKY where it energizes a DSKY relay. The relay causes the CMC warning indicator to light on the caution and warning panel.

9.10.7 CMC'S INPUT AND OUTPUT CHANNEL INTERFACE. In addition to the counter interrupt and the program interrupts previously described, the CMC has a number of other inputs derived from its interfacing hardware. These inputs are a result of the functioning of the hardware or an action by the operator of the spacecraft. The counter interrupts in most cases enable the CMC to process inputs representative of data parameters such as changes in velocity. The program interrupt inputs to the CMC are used to initiate processing of functions which must be processed a relatively short time after a particular function is present. The other inputs to the CMC, in general, enable the CMC to be cognizant of "conditions" which exist in its environment. These inputs are routed to CMC and are available to the CMC's programs through the input channels.

The outputs of the CMC fall in one of the following categories: (1) data, (2) control, (3) condition indications. Some of these outputs are controllable through the CMC's program while others are present as a function of the CMC circuitry. All of the outputs which are controlled by the CMC's programs are developed through the CMC's output channels.

9.10.7.1 CMC Input/Output Channel Bit Assignments. Table 9-8 shows the CMC input/output channel bit assignments. A brief description of the channels and reference information concerning them appears in table 9-9.

9.10.7.2 PIPA Precount Logic. The PIPA precount logic consists of three forward-backward counters and PIPA failure detection circuitry. The forward-backward counters eliminate the PIPA pulses due to the PIPA 3-3 moding so that only PIPA pulse due to an acceleration will cause counter instructions to occur. The PIPA failure detection circuitry generates a PIPA fail (PIPAFL) signal if anyone of three abnormal conditions occur.

When a PIPA is sensing no acceleration, the PIPA loop in the ISS continuously torques the PIPA with a series of three + pulses followed by a series of 3- pulses. The foward-backward counter associated with the particular PIPA loop receives these + and - pulses. The counter counts forward three and then backward three without generating an output. Only when the counter receives more than three positive or negative pulses in a sequence does it generate an output. The output is dependent on the input, in that, a plus output pulse (PIPXP, PIPYP or PIPZP) is generated if the plus input pulses (PIPGX+, PIPGY+ or PIPGZ+) exceed three and a minus output pulse (PIPXM, PIPYM, or PIPZM) is generated if the minus input pulses (PIPGX-, PIPGY- or PIPGZ-) exceeds three. The operation of one of the three forward-backward counters is illustrated in figure 9-45 and table 9-10.

CHANNEL		NAME	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	CHANNEL
CP	1	L						CP REGISTER L, BITS 16-1										1
	2	Q						CP REGISTER Q, BITS 16-1										2
	3	HISCALAR						HIGH - ORDER SCALER CHANNEL, BITS 14-1										3
	4	LOSCALAR						LOW - ORDER SCALER CHANNEL, BITS 14-1										4
OUT	5	PYJETS							S/M →	+X - YW	-X+YW	-X-YW	+X+YW	+X-P	-X+P	-X-P	+X +P	5
									C/M →	-YW -X - P	+YW -X - P	-YW -X -P	+YW -X + P	+R-P+Z	+P -X - YW	-R-P+Z	+P-X+YW	
	6	ROLLJETS							S/M →	+Y -R	-Y+R	-Y-R	+Y+R	+Z-R	-Z +R	-Z-R	+Z +R	6
									C/M →					-R-YW+Z	+R+YW-Z	-R-YW-Z	+R+YW+Z	
CP	7	SUPERBNK									FE7	FE6	FE5					7
OUT	10	OUTO	RELAY ADRS 4	RELAY ADRS 3	RELAY ADRS 2	RELAY ADRS 1	RELAY BIT 11	RELAY BIT 10	RELAY BIT 9	RELAY BIT 8	RELAY BIT 7	RELAY BIT 6	RELAY BIT 5	RELAY BIT 4	RELAY BIT 3	RELAY BIT 2	RELAY BIT 1	10
	11	DSALMOUT			SPS ENGINE ON			CAUTION RESET	TEST CONNECTOR OUTBIT		OPERATOR ERROR LAMP	VN FLASH	KEY REL LAMP	TEMP CAUTION LAMP	UPLINK ACTY LAMP	COMP ACTY LAMP	ISS WARNING	11
	12	CHAN12	ISS TURNON DELAY COMPLETE	SIV B CUTOFF	SIV B INJ. SEQ. START		DENG OPTICS DAC	ZERO OPTICS	SIV B TAKEOVER ENABLE	TVC ENABLE		ENABLE IMU ERROR COUNTER	ZERO IMU CDU'S	COARS ALIGN ENABLE		ENABLE OPT ERROR COUNTER	ZERO OPTICS CDU'S	12
	13	CHAN13	ENABLE T6RUPT	RESET TRAP 32	RESET TRAP 31B	RESET TRAP 31A	ENABLE STANDBY	TEST ALARMS		BMAG CTR ENABLE	DNLNK WD ORD	BLOCK INLINK						13
	14	CHAN14	DRIVE CDUX	DRIVE CDUY	DRIVE CDUZ	DRIVE CDU T	DRIVE CDU S	GYRO ACTY	GYRO c	GYRO a	GYRO b	GYRC ENABLE						14
IN	15	MNKEYIN											MKEY5	MKEY4	MKEY3	MKEY2	MKEY1	15
	16	NAVKEYIN									MARK REJECT	MARK	NKEY5	NKEY4	NKEY3	NKEY2	NKEY1	16
	30	*CHAN30	TEMP IN LIMITS	ISS TURNON REQUEST	IMU FAIL	ICDU FAIL	IMU CAGE	S/C CONTROL OF SAT	IMU OPERATE		OPTICS CDU FAIL	GUID REF RELEASE	LIFTOFF	SIV B SEPARATE, OR ABORT	SPS READY	SM/CM SEPARATE	ULLAGE THRUST PRESENT	30
	31	*CHAN31	G & N AUTOPILOT CONTROL	FREE	HOLD	-Z TRANS	+Z TRANS	-Y TRANS	+Y TRANS	-X TRANS	+X TRANS	RHC -ROLL	RHC +ROLL	RHC -YAW	RHC +YAW	RHC -PITCH	RHC +PITCH	31
	32	*CHAN32		PROCEED			LM ATTACHED					MNIM -ROLL	MNIM +ROLL	MNIM -YAW	MNIM +YAW	MNIM -PITCH	MINIM +PITCH	32
	33	*CHAN33	OSC ALARM	COMPUTER WARNING	PIPA FAIL	DNLNK TOO FAST	UPLNK TOO FAST	BLOCK UPLINK					CMC CONTROL	ZERO OPTICS				33
OUT	34	DNTM1						FIRST OF TWO WORDS										34
	35	DNTM2						SECOND OF TWO WORDS										35
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

* INVERTED LOGIC USED

Table 9-8. CMC Input/output Channel Bit Assignments

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Description	References
0			Vacant	
1	1-16	L	Central processor register L	para. 10.5.2.1
2	1-16	Q	Central processor register Q	para. 10.5.2.1
3	1-14	HISCALER	High order scaler - furnishes a 14 bit positive number whose least significant bit has a weight of 5.12 seconds. The period of the number is 23.30 hours. Because of logic propagation delays, this channel should be interrogated more than once to resolve erroneous reads at the time of transition.	
4	1-14	LOSCALER	Low order scaler - furnishes a 14 bit positive number whose least significant bit has a weight of 1/3200 second. The period of the number is 5.12 seconds. As above this channel should be interrogated more than once to resolve erroneous reads.	
5	1-8	PYSETS 1 2 3 4 5 6 7 8	In the service module this channel has 8 bit positions +X+P -X-P -X+P +X-P +X+Y -X-Y -X+Y +X-Y and is associated with the RCS jets. The first letter contained in the bit positions of the figure refers to a translational motion, the second to a rotational motion. For instance, if it were desired to perform a pure positive translation along the X-axis, several different logic configurations could be associated with the bit positions of this channel. a. 1) Bit positions 1 and 4 equal to a logic one 2) All other bit positions equal to a logic zero b. 1) Bit positions 5 and 8 equal to a logic one 2) All other bit positions equal to a logic one c. 1) Bit positions 1, 2, 5, and 8 equal to a logic one 2) All other bit positions equal to a logic zero	(Cont)

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Description	References
			Note that for a pure translation, the logic levels of the bit positions must be chosen in such a fashion that any rotational motions are cancelled out.	
5	1 2 3 4 5 6 7 8		+P-X+YW -P+Z-R +P-X-YW -P+Z+R +YW-X+P -YW-X-P +YW-X-P -YW-X+P In the command module this channel is associated with the RCS jets. The letters to the right, indicate the command module motion corresponding to a logic 1 in a particular bit location. If, for example, a logic 1 is placed in bit position 5 the CM would experience a + yaw, a + pitch and a translation along the -X axis.	
6	1-8	ROLL JETS 1 2 3 4 5 6 7 8	RCS+Z+R RCS-Z-R RCS-Z+R RCS+Z-R RCS+Y+R RCS-Y-R RCS-Y+R RCS+Y-R In the service module this channel is associated with the RCS jets. For instance, if a CM roll is desired it can be accomplished by several different logic configurations in the channel bit positions. a. 1) Bit positions 1 and 3 equal to a logic 1 2) All other bit positions equal to a logic 0 b. 1) Bit positions 5 and 7 equal to a logic 1 2) All other bit positions equal to a logic 0	

(Cont)

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Designation	References
6	1-8	Roll JETS 1 2 3 4	+R+Y+Z -R-Y-Z +R+Y-Z -R-Y+Z	In the command module this channel is associated with 4 RCS jets. The letter to the right indicate the command module motion corresponding to a logic 1 in a particular bit location.	
7	3 5 6 7	SUPERBNK	This channel is used to increase fixed memory addressing capability. FE 7 FE 6 FE 5 High Banks 0 X X 30-37 1 0 0 40-43 1 0 1 EMP. 1 1 0 EMP. 1 1 1 EMP.	
10	16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15/16	OUTO	Relay Bit 1 Relay Bit 2 Relay Bit 3 Relay Bit 4 Relay Bit 5 Relay Bit 6 Relay Bit 7 Relay Bit 8 Relay Bit 9 Relay Bit 10 Relay Bit 11 Relay Address 1 Relay Address 2 Relay Address 3 Relay Address 4	The information contained in this channel is routed to the DSKYs. The different configurations light various displays on the DSKY's. Bits 1-11 are associated with column selection signals and bits 12-15 are associated with roll selection signals determining a specific position in a relay matrix.	Para. 11.1 Para. 11.3

(Cont)

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Designation	Reference
11	15	DSALMOUT	The information in this channel is routed to the DSKY.	
	1		ISS Warning - lights a DSKY caution light	Table 11-3
	2		Light Computer Activity Lamp	Table 11-3
	3		Light Uplink Activity Lamp	Table 11-3
	4		Light Temperature Caution Lamp	Table 11-3
	5		Light Keyboard Release Lamp	Table 11-3
	6		Flash verb and noun lamps	Table 11-3
	7		Light operator error lamp	Table 11-3
	8		Spare	
	9		Test connector outbit	
	10		Caution reset	
	11		Spare	
	12		Spare	
	13		Engine On	
	14		Engine Off	
	15		Spare	
12	15	CHAN12	This channel consists of 15 bits. The outbits are dc signals sent to the S/C and PGNC systems.	
	1		Zero CDU optics	
	2		Enable optical error counter	
	3		Spare	
	4		Coarse align enable	
	5		Zero IMU CDU	
	6		Enable IMU error counter	
	7		Engine on/off	Figure 4-14
	8		Thrust vector control enable	Figure 4-14
	9		SIVB takeover enable	Figure 4-12
	10		Zero optics	
	11		Disengage optic DAC	
	12		Spare	
	13		SIVB Inj. Seq. start	
	14		SIVB cutoff	
	15		ISS turn on delay complete	

(Cont)

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Description	References
13	15	CHAN13		
	1-4		Spares	
	5		Uplink inhibit	
	6		Block inlink	
	7		Downlink word order	
	8		BMAG CTR enable	
	9		Spare	
	10		Test alarms	
	11		Enable standby	
	12		Reset Trap 31A	
	13		Reset Trap 31B	
	14		Reset Trap 32	
	15		Enable T6 RUPT	
14	15	CHAN14		
	1		Spare	
	2-5		Spare	
	6		Gyro enable	
	7		Gyro b	
	8		Gyro a	
	9		Gyro c	
	10		Gyro activity	
	11		Drive CDU S	
	12		Drive CDU T	
	13		Drive CDU Z	
	14		Drive CDU Y	
	15		Drive CDU X	
15	5	MNKEYIN	This channel consists of 5 bit positions.	
	1		KEY 1M	Whenever a key on the
	2		KEY 2M	DSKY is pressed, a
	3		KEY 3M	unique 5 bit code is
	4		KEY 4M	generated and entered
	5		KEY 5M	into this channel. The RUPT 5 interrupt routine is also developed whenever a key is depressed.
16	7	NAVKEYIN DSKY	This channel consists of 7 bit positions. The	
	1		KEY 1N	function of the first
	2		KEY 2N	5 positions are the
	3		KEY 3N	same as those in
	4		KEY 4N	channel 15.
	5		KEY 5N	

(Cont)

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Designation	References
16	6		MARK	
	7		MARK REJECT	
			If the mark pushbutton is depressed, a 1 is put in bit position 6 and would cause a KEYRUPT 2 (RUPT6) interrupt routine. If the MARK REJECT pushbutton is depressed, a logic is entered in bit 7 and would cause a KEYRUPT 2 interrupt routine.	
30	15	*CHAN30		
	1		Ullage thrust present	
	2		SM separate	
	3		SPS ready	
	4		SIVB separate, abort	
	5		Lift off	
	6		Guidance reference release	
	7		Optics CDU fail	
	8		Spare	
	9		IMU Operate	
	10		S/C control of SAT	
	11		IMU cage	
	12		ICDU fail	
	13		IMU fail	
	14		ISS trunnion request	
	15		Temp in limits	
31	15	*CHAN31	Consists of 15 bit positions	
	1		MANR+P	
	2		MANR-P	
	3		MANR+Y	
	4		MANR-Y	
	5		MANR+R	
	6		MANR-R	
	7		+X TRANS	
	8		-X TRANS	
	9		+Y TRANS	
	10		-Y TRANS	
	11		+Z TRANS	
	12		-Z TRANS	
	13		HOLD FUNCTION	
	14		FRE FUNCTION	
	15		S/C AUTO CONTROL	

(Cont)

Table 9-9. Input/Output Designation and Reference

Channel	Bit Pos	Name	Designation	Reference
32	15	*CHAN32	Consists of 15 bit positions.	
	1		MNIM+P	
	2		MNIM-P	
	3		MNIM+Y	
	4		MNIM-Y	
	5		MNIM+R	
	6		MNIM-R	
	7		Spare	
	8		Spare	
	9		Spare	
	10		Spare	
	11		LM attached	
	12		Spare	
	13		Spare	
	14		Spare	
	15		Spare	
33	15	*CHAN33		
	1		Spare	
	2		Spare	
	3		Spare	
	4		Zero optics	
	5		CMC control	
	6		Spare	
	7		Spare	
	8		Spare	
	9		Spare	
	10		Block uplink	
	11		Uplink too fast	
	12		Dnlink too fast	
	13		PIPA fail	
	14		Warning	
	15		OSC alarm	
34	15	DNTM1	First of two, 15 bit words	
35	15	DNTM2	Second of two, 15 bit words	

* Inverted Logic.

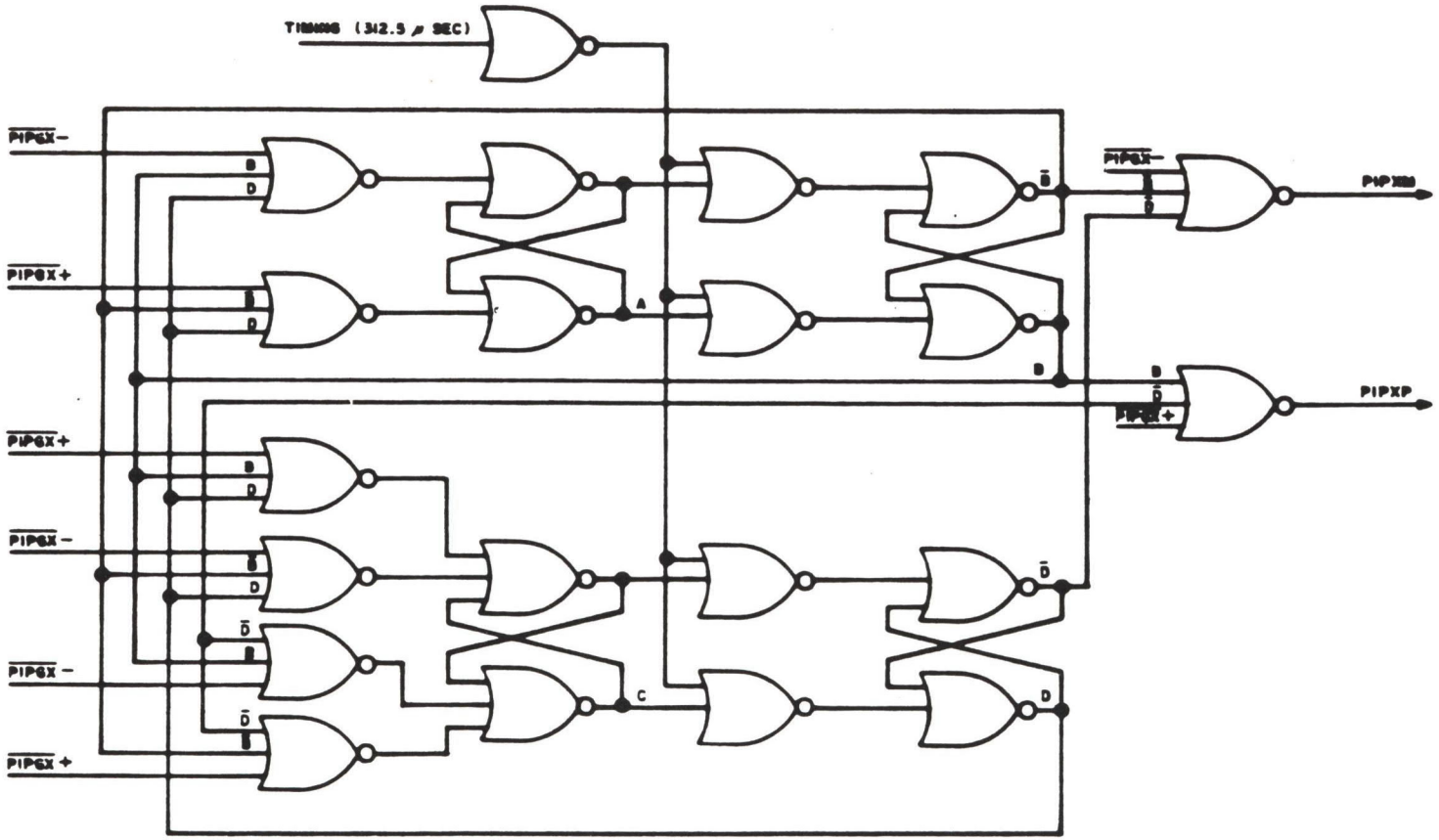


Figure 9-45. PIPA Forward-Backward Counter

Table 9-10. PIPAX Counter Truth Table

Initial Conditions		Input Signal	Resulting Conditions		
Signal B	Signal D		Signal B	Signal D	Output Signal
1	1	PIPGX+	1	0	-
1	0	PIPGX+	0	0	-
0	0	PIPGX+	0	1	-
0	1	PIPGX+	0	1	PIPXP
0	1	PIPGX-	0	0	-
0	0	PIPGX-	1	0	-
1	0	PIPGX-	1	1	-
1	1	PIPGX-	1	1	PIPXM

The PIPA failure detection circuitry monitors for the following conditions:

a. That each PIPA have output pulses (plus and minus) within a given period of time.

b. That no PIPA has both plus and minus pulses at the same time.

c. That each PIPA has a plus or a minus pulse within a given time.

If any of these three conditions exist, a PIPA fail (PIPAFL) signal is generated. This signal is applied to channel 33 bit position 13 and is used internal to the computer.

The circuit that monitors for plus and minus pulses within a given time period is illustrated in figure 9-46. When a PIPA pulse is generated, the associated input flip-flop is reset and generates signal NOXP, NOXM, NOYP, NOYM, NOZP or NOZM equal to logic zero. All six NO__ signals are applied to an "NOR" gate and if all inputs are logic "ZERO" the output of the NOR gate will be a logic "ONE". A logic "ONE" applied to the next NOR gate will force its output to be a logic "ZERO". If all inputs to the output flip-flop are logic "ZERO", the flip-flop will remain reset (signal PIPAFL = 0). At time F18AX, the input flip-flops are all set (signals NO__ = 1). If any PIPA does not produce both a plus and a minus pulse within the following 1.3 seconds, the particular input flip-flop will not be reset; therefore, the next NOR gate will have a logic "ZERO" output. This logic "ZERO" output is applied to the next NOR gate and at time $\overline{F18B}$ the gate will produce a logic "ONE" which sets the output flip-flop and generates signal PIPAFL.

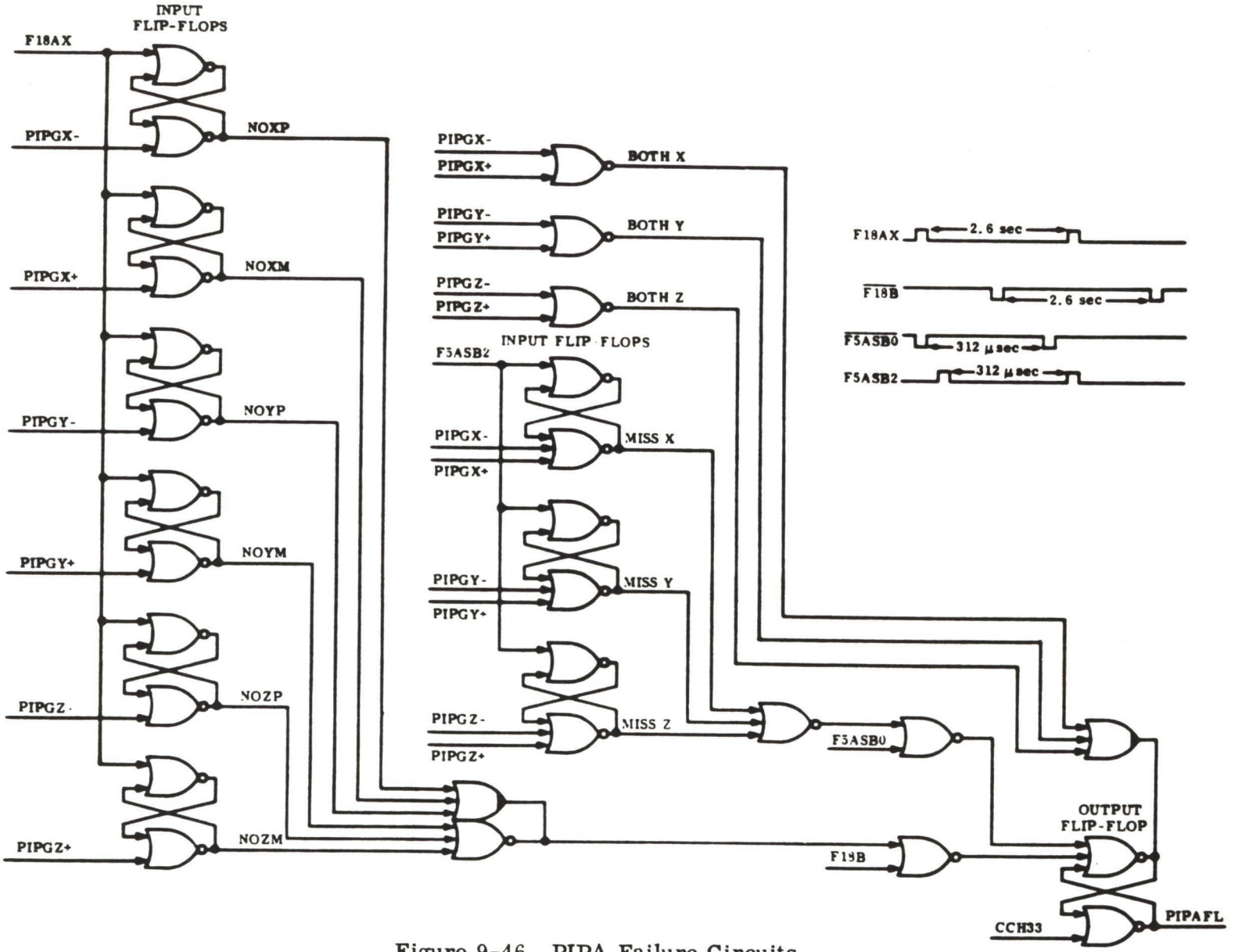


Figure 9-46. PIPA Failure Circuits

The circuit that monitors for a plus or a minus PIPA pulse within a given period of time will generate the PIPA fail signal (PIPAFL) if each PIPA does not generate either a plus or a minus pulse for every PIPA data pulse generated. Timing signal F5ASBZ sets the input flip-flops associated with this circuitry every time that a PIPA data pulse is applied to the PIPA binary current switches. Each PIPA loop must produce a plus or a minus pulse for every PIPA data pulse. The plus or minus pulses reset the input flip-flops prior to timing signal F5ASBO. If all PIPA loops generate a plus or a minus pulse. All three of the input flip-flops will be reset (MISSX, MISSY and MISSZ = 0). The next NOR gate will generate a logic "ONE" and prevent the setting of the output flip-flop. If one of the PIPA loops fails to produce a plus or a minus pulse prior to time F5ASBO, the output flip-flop will be set (PIPAFL = 1).

The circuit that monitors for both plus and minus PIPA pulses occurring at the same time will generate the PIPAFL signal be setting the output flip-flop through the extended NOR gate. Signal Both X, Both Y, or Both Z will set the output flip-flop.

9.10.7.3 Interface Circuits. The interface circuits provide interfacing between the computer and the DSKY's, the spacecraft, and the remainder of the PGNCS. All input to and output from the computer are routed through these circuits. See figure 10-54.

There are seven types of interface circuits that provide the proper voltage levels and impedance matching between the computer and other spacecraft systems. These seven circuit types are designated circuits A, C, D, P, XT, R, and Y.

The A type circuit is an analog to digital converter which is used in the LGC only in conjunction with propulsion rate commands.

The C type circuit is a transistor driver circuit which buffers discrete type output from the computer to the DSKY's and the reaction control system.

The D type circuit consists of an RC filter, the output of which is applied directly to the computer logic circuits. The input signals involved are discrete type signals from the DSKY, other PGNCS subsystems, and the other spacecraft systems.

The P type circuit filters the +14 vdc input to the XT circuits.

The R type circuit is a series resistor and is used to provide short circuit protection for output from the power supply.

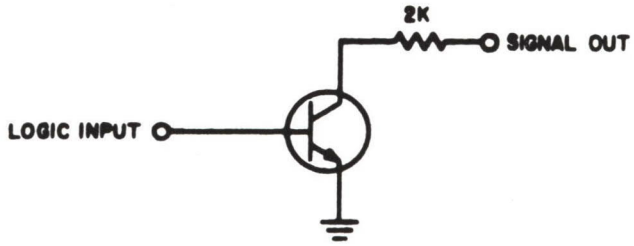
The XT type circuit consists of a pulse transformer driven by an input transistor circuit. It provides the required output impedance to match spacecraft circuits and other subsystem circuits. It also is used as an input interface circuit to provide impedance matching and the voltage level necessary to operate the logic in the computer.

All signals interfacing with the computer have an alphanumeric code rather than the functional name when routed outside of any particular subsystem. This code is used in the interface drawings and designates the interface circuits type, signal type and number (for example, DE 040). The first letter designates the interface circuit type:

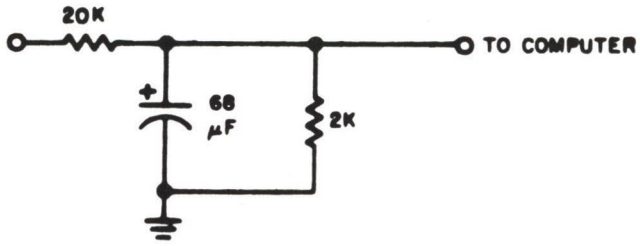
A analog-to-digital converter

C discrete type output circuit

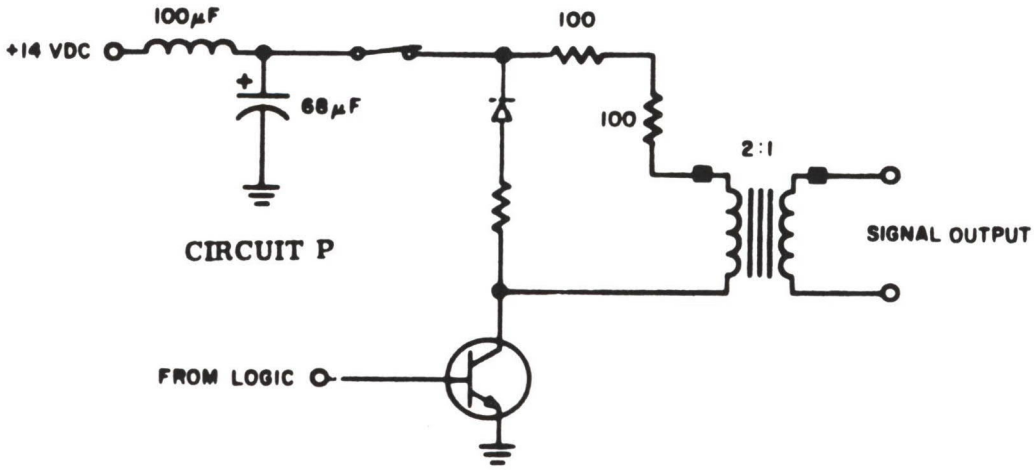
D discrete type input circuit



CIRCUIT C

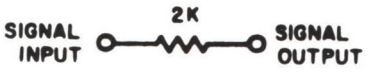


CIRCUIT D

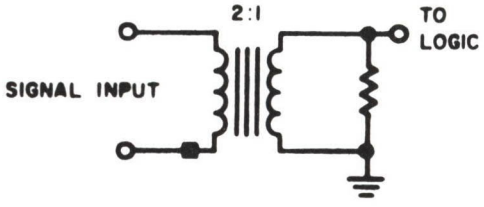


CIRCUIT P

CIRCUIT XT



CIRCUIT R



CIRCUIT Y

Figure 9-47. Interface Circuit Types

R resistor in series

S switch closure

W connecting wire

X transformer coupled output circuit (XT)

Y transformer coupled input circuit

The second letter designates the type of signal:

A indicates the signal is under counter control (each output pulse counted)

B indicates the signal is under program control

C indicates the signal is continuous

D indicates the signal is a dc level

E indicates the signal goes to an IN bit or comes from an OUT bit

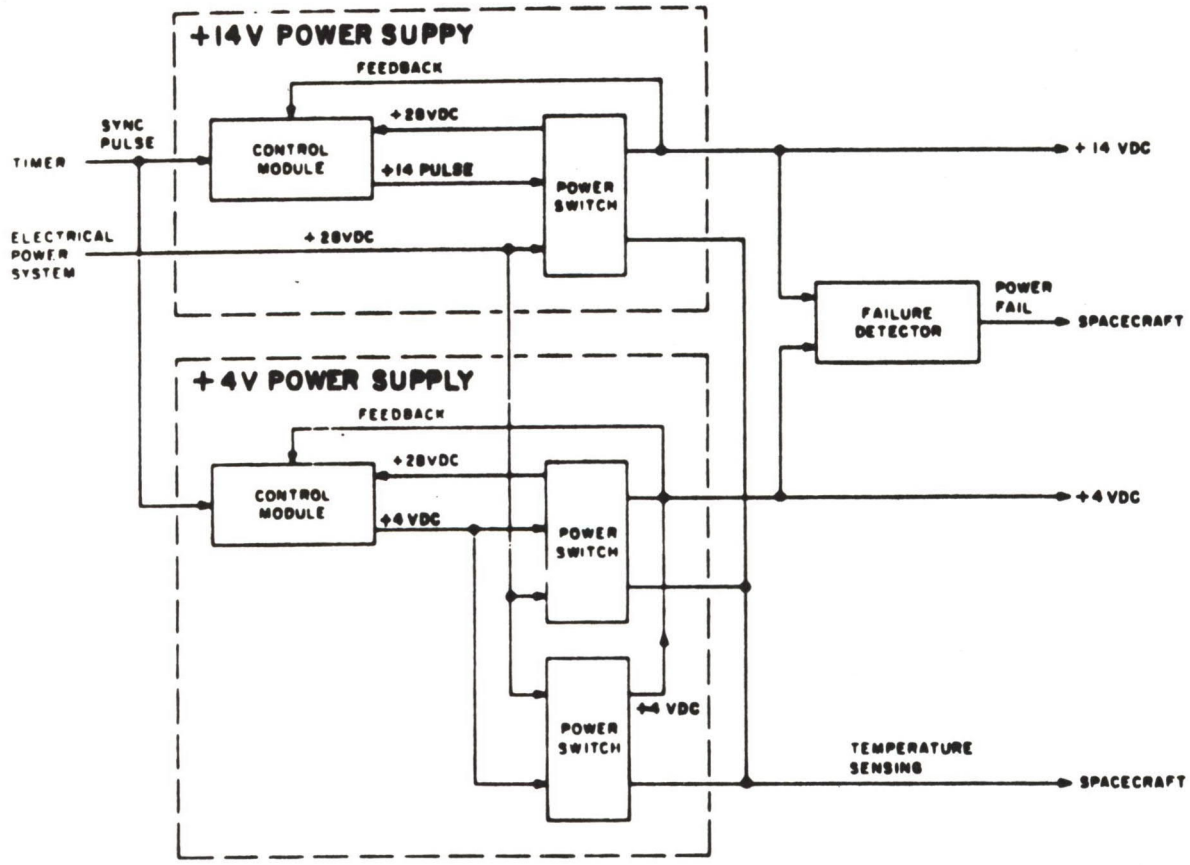
G indicates the signal goes to a counter

The digits indicate interface signals between specific systems:

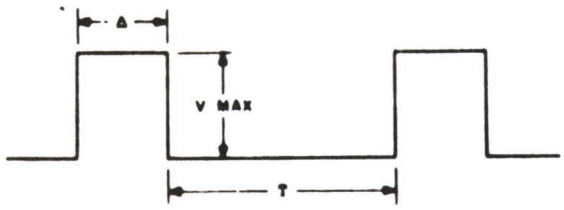
001 - 100	computer/spacecraft
100 - 200	computer/PGNCS
200 - 300	computer/DSKY
300 - 400	DSKY/PGNCS
400 - 500	DSKY/spacecraft
600 - 700	computer/GSE
700 - 800	computer/GSE
800 - 900	computer/spacecraft
900 - 1000	computer/PGNCS

9.10.8 POWER. CMC power (figure 9-48) is furnished by two switching-regulator power supplies: a +4 volt and a +14 volt power supply which are energized by fuel cells in the Electrical Power System.

Input voltage from the Electrical Power System is chopped at a variable duty cycle and then filtered to produce the required voltages. Chopping is accomplished by varying the pulse width of a signal having a fixed repetition rate and known amplitude. Referring to figure 10-55, V_{out} can be stated as $V_{out} = V_{max} \frac{\Delta}{A}$ Where: A is the fixed repetition rate, Δ is the pulse width, V_{max} is the known input amplitude. Therefore, V_{out} can be varied by changing the pulse width, Δ. Operation of the +14 volt power supply, which is typical of both, is described.



(a)



(b)

Figure 9-48. Computer Power Supply

Source voltage, +28 vdc, is supplied from the Electrical Power System through the power switch to the control module. The control module, essentially a pulse generator, detects the difference between the primary feedback output of the power supply and a reference voltage. (A secondary feedback path is connected to the CTS for marginal-voltage test operations.) A differential amplifier detects any change in the output voltage from the desired level. The output of the differential amplifier and a 51.2 kilocycle sync pulse from the timer drive a one-shot multivibrator in the control module. The differential amplifier output determines the multivibrator pulse width. The resultant +14 volt pulse is supplied to the power switch.

The power switch filters the control module output to produce the desired dc voltage. Additional filtering action protects the Electrical Power System from the wide load variations caused by the chopping action of the power supply. The power switch also contains a temperature sensing circuit. The +4 volt power supply requires two power switches due to load requirements.

The power supply outputs are monitored by a failure detector consisting of four differential amplifiers. There are two amplifiers for each power supply, one for overvoltage and one for undervoltage detection. If an overvoltage or undervoltage condition exists, a relay closure signal indicating a power fail is supplied to the spacecraft.

9.11 DSKY FUNCTIONAL OPERATION

The DSKY (figures 9-49 and 9-50) consist of a keyboard, power supply, decoder relay matrix, status and cuation circuits and displays.

The keyboard contains the key controls with which the astronaut operates the DSKY. Each of the key controls is lighted by 115 vac, 400 cps. Inputs to the CMC initiated from the keyboard are processed by the program. The results are suppled to either the decoder and relay matrix or to the status and caution circuits for display. Each key when depressed, with the exception of standby, will produce a 5 bit code. The keycode enters into the CMC and initiates an interrupt to allow the data to be accepted. The key reset signal (+ 28 vdc) is generated each time a key is released, and conditions the CMC to accept another keycode. The reset code and signal (+ 28 vdc) is used when the astronaut wishes certain display indicators to go out. It also checks on whether a particular indicator is transient or permanent. The clear code is used when the astronaut wishes to clear displayed sign and digit information. Key release turns the control of displaying information on the DSKY over to the CMC. The standby signal (+ 28 vdc) initiates putting the CMC into the standby mode. It also initiates putting the CMC into operate mode when pressed a second time.

The power supply utilizes + 28 vdc and + 14 vdc from the CMC power supply and an 800 cps sync signal from the timer to generate a 250 volt, 800 cps display voltage. The display voltage is applied to the displays through the relay matrix and status and caution circuits.

The decoder receives a four bit relay word (bits 12 through 15) from channel 10 in the CMC. The decoded relay word, in conjunction with relay bits 1 through 11 from channel 10, energizes specific relays in the matrix. The relays are energized by the coincidence of a selection signal from the diode matrix in the decoder which produces a row selection signal, and relay bits which produce column selection signals. Relay selection allows the display voltage (250 vac) from the power supply to be routed to the proper sign and digit indicators. Relay selection also allows the alarm common (0 vdc) or + 5 vac from the PGNCS, or from the CM, to be routed through the relay to the PGNCS or to the CM (caution signals) or to the proper status and caution indicators respectively. The PGNCS caution signals from the relay matrix, represented by 0 vdc, are PROG CAUTION and GIMBAL LOCK. The status and caution indicators, lit by the + 5 vac are: PROG, GIMBAL LOCK and NO ATT. All relays associated with the relay matrix are the latching type.

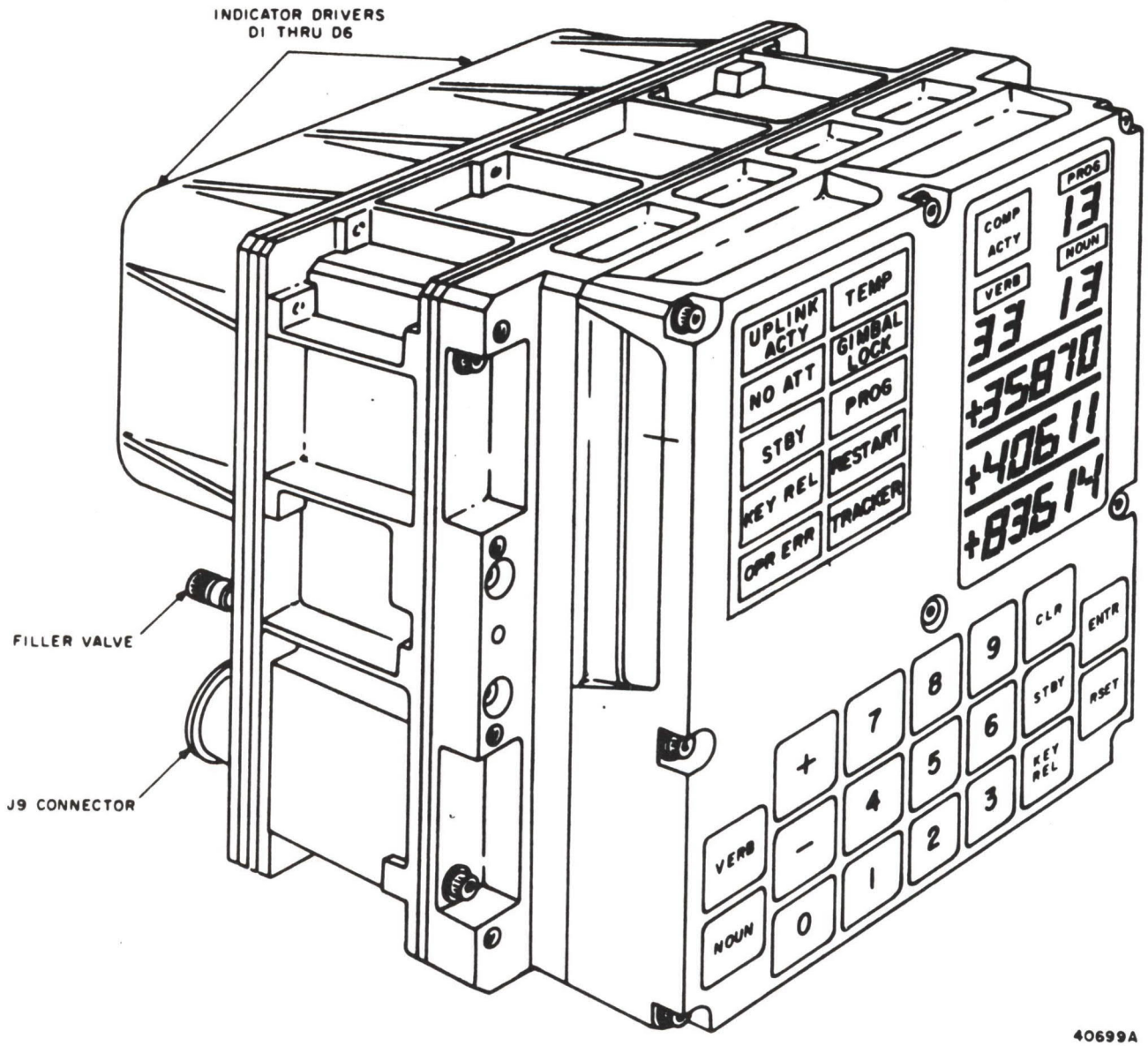


Figure 9-49. Main Panel and Navigation Panel DSKY

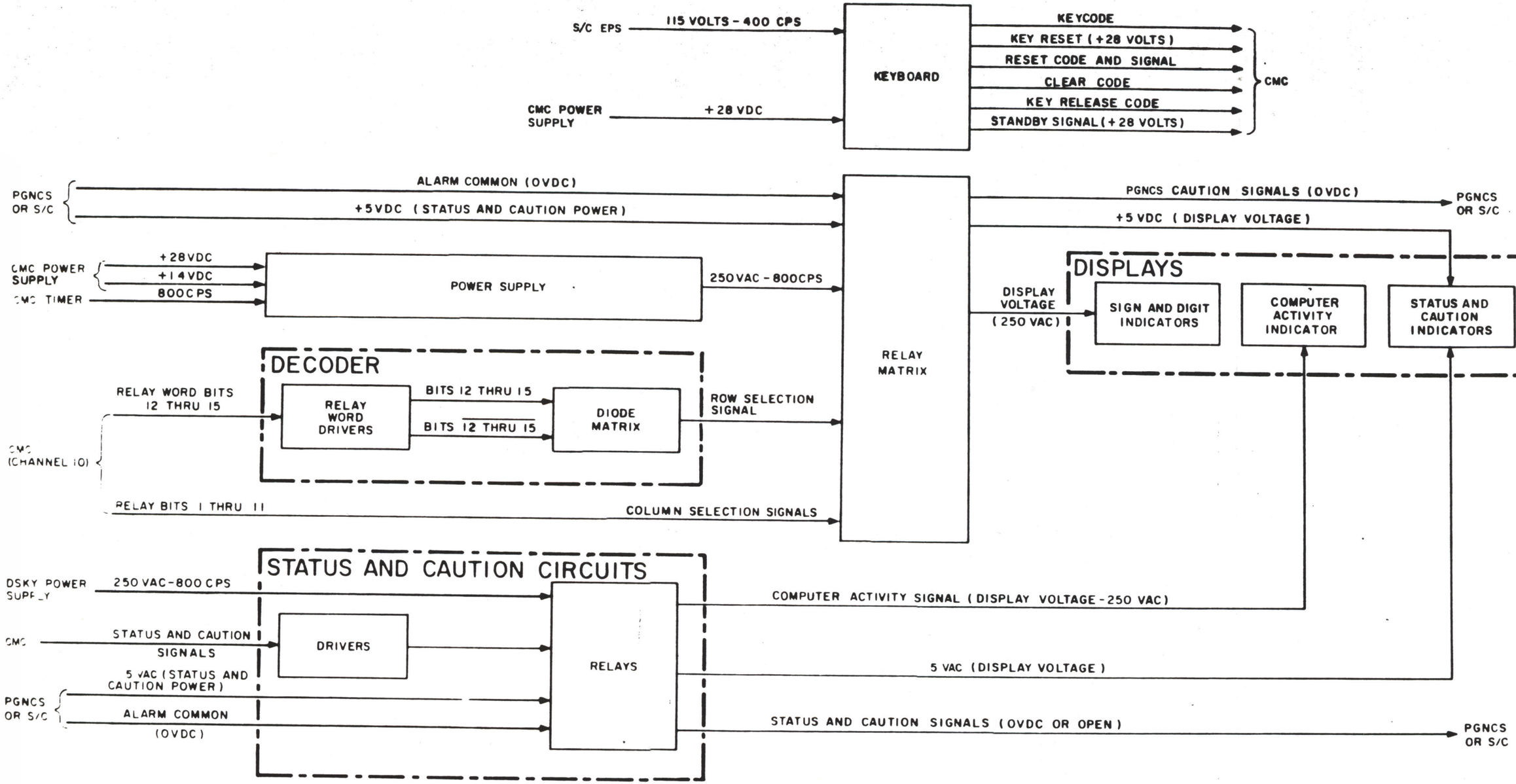


Figure 9-50. DSKY Functional Diagram

The status and caution circuits receive all CMC status and caution signals. Each signal is applied to a driver circuit and to an associated relay. When a relay is energized, it allows the voltage from the DSKY power supply (250 vac), or +5 vac or 0 vdc from the PGNCS or CM to be routed to the proper display indicators or equipment. The voltage from the power supply is routed through a relay to the computer activity indicator (COMP ACTY). The +5 vac is routed through relays to the following status and caution indicators: UPLINK ACTY, RESTART, OPR ERR, KEY REL, and TEMP. The status and caution signals, represented by 0 vdc or an open circuit, are ISS WARNING, STBY, CMCWAR, TEMP CAUTION, and RESTART. All relays associated with the status and caution circuits are the non-latching type.

The displays consist of sign and digital (operational and data display) and status and caution indicators. The sign and digital indicators allow the astronaut to observe the data entered or requested from the keyboard. The status and caution indicators present an indication of any variance from certain normal operations.

9.12 KEYBOARD

The keyboard consists of ten numerical keys (pushbuttons) labeled 0 through 9, two sign keys (+ or -) and seven instruction keys: VERB, NOUN, CLR (clear), STBY (standby), KEY REL (key release), ENTR (enter) and RSET (reset). Table 9-11 lists these keys (pushbuttons) and their functions.

Whenever a key is depressed, +14 vdc is applied to a diode encoder which generates a unique five bit code associated with that key. There is, however, no five bit code associated with the STBY key. These keys and their respective codes are shown in figure 9-51. If a key on the main panel DSKY is pressed, the five bit code associated with that key is entered into bit positions 1 through 5 of Input Channel 15 of the CMC. Note that this input will cause a request for the KEYRUPT 1 program interrupt. If a key on the navigation panel DSKY is pressed, the five bit code associated with that key is entered into bit position 1 through 5 of Input Channel 16 of the CMC. Note that this input will cause a request for the KEYRUPT 2 program interrupt.

The switches for the keys are wired in series to insure that only one input at a time is presented to the diode encoder and, consequently, only one code at a time to the appropriate input channel. Trap reset signals are associated with each DSKY. When a key is released on this main panel DSKY, signal TRAP 15 RESET is sent to trap circuitry in the CMC associated with the KEYRUPT 1 program priority interrupt; when a key is released on the navigation panel DSKY, signal TRAP 16B RESET is sent to the trap circuitry in the CMC assocated with the KEYRUPT 2 program priority interrupt.

9.13 DISPLAY INDICATORS

There are 24 display indicators on the DSKY; 21 digit display indicators and three sign display indicators. The locations of the digit display indicators are shown on figure 9-52. The indicators and their functions are described in table 9-12.

Table 9-11. DSKY Pushbuttons

Pushbutton	Function
0 through 9 pushbuttons	Enter numerical data, noun codes and verb codes into the CMC.
+ and - pushbuttons	Inform the CMC that the following numerical data is decimal and indicate the sign of the data.
NOUN pushbutton	Conditions the CMC to interpret the next two numerical characters as a noun code and causes the noun display to be blanked.
CLEAR pushbutton	Clears data contained in the data displays. Depressing this key clears the data display currently being used. Successive depressions clear the other two data displays.
STBY pushbutton	Commands the CMC to the standby mode when depressed the first time. An additional depression commands the CMC to resume regular operation.
KEY REL pushbutton	Releases the DSKY displays initiated by keyboard action so that information supplied by the CMC program may be displayed.
ENTR pushbutton	Informs the CMC that the assembled data is complete and that the requested function is to be executed.
RSET pushbutton	Extinguishes the lamps that are controlled by the CMC.
VERB pushbutton	Conditions the CMC to interpret the next two numerical characters as a verb code and causes the verb display to be blanked.

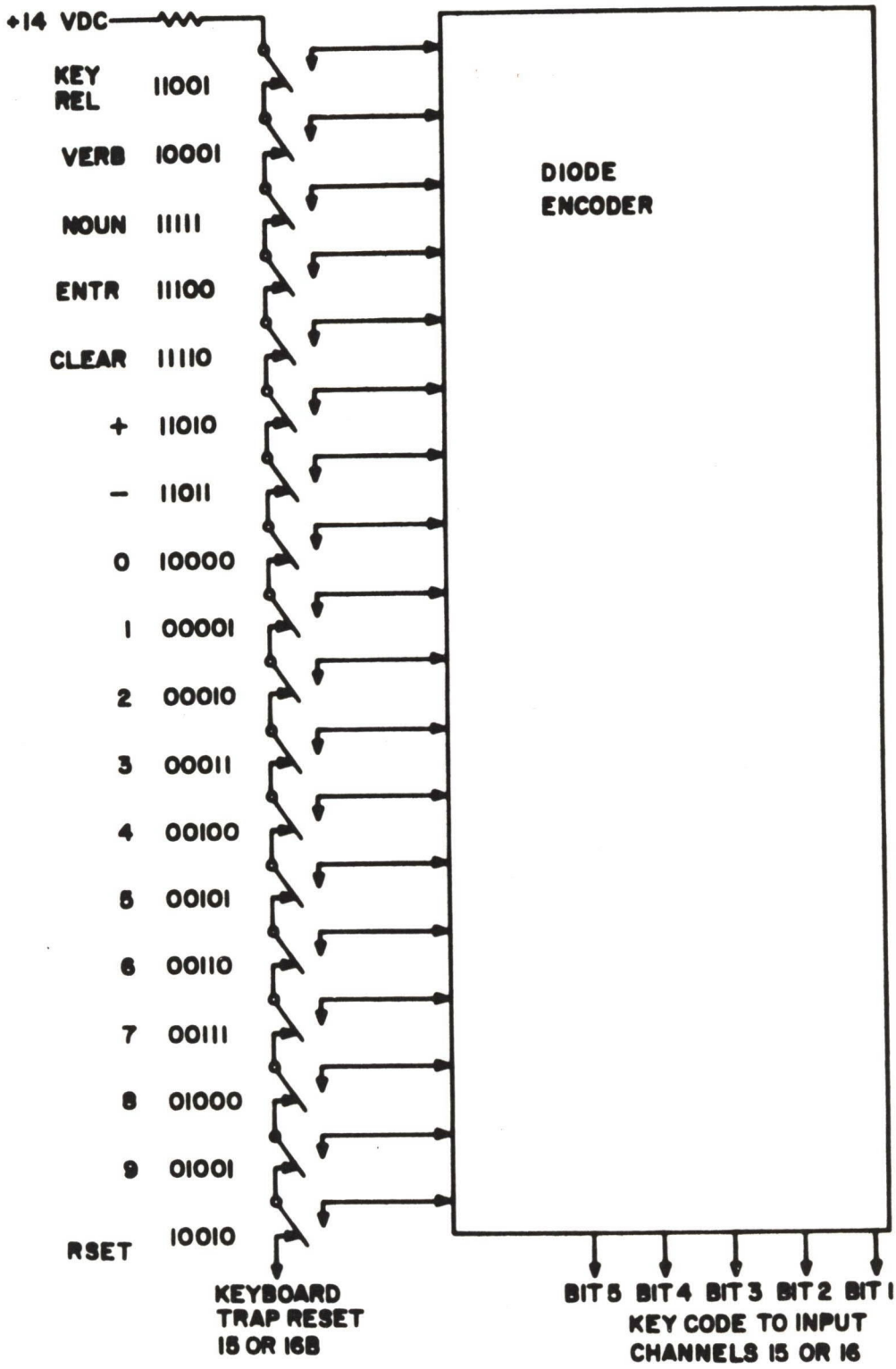


Figure 9-51. Diode Encoder and Associated Codes

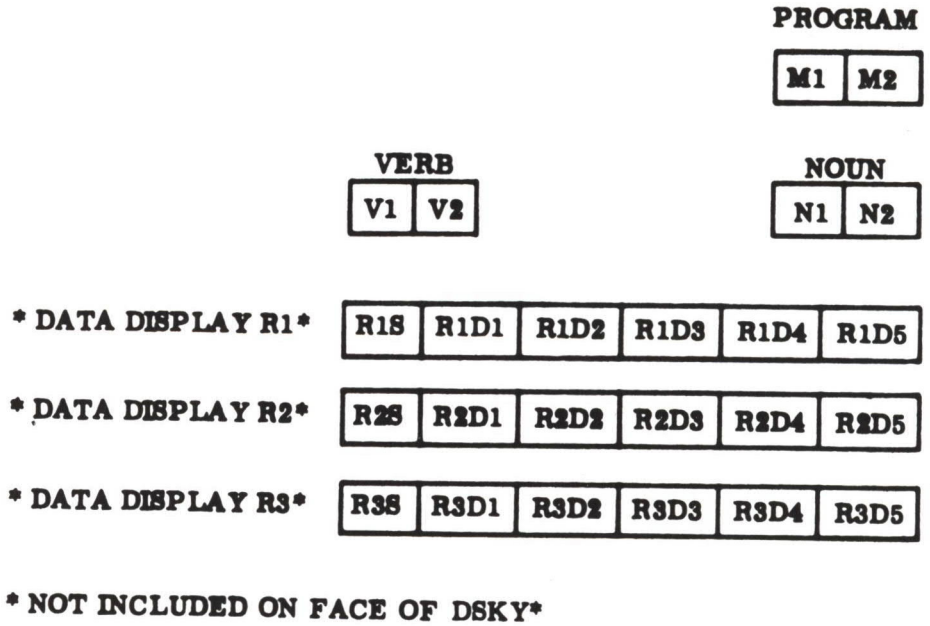


Figure 9-52. Display Indicators

Table 9-12. Display Indicators and Functions

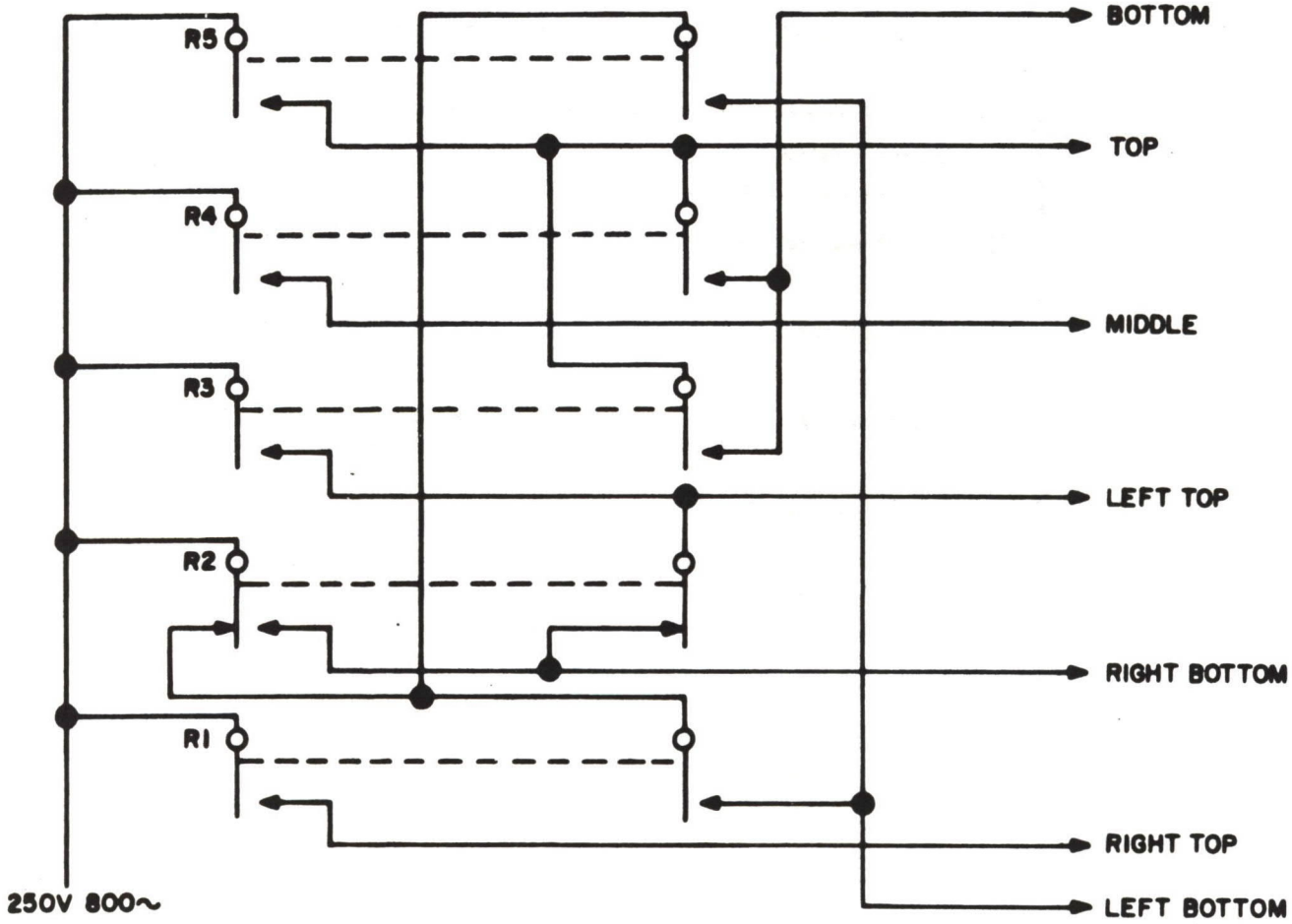
Display Indicator	Function
PROGRAM indicators	Indicate program being processed by CMC.
VERB indicators	Indicate verb code entered at keyboard or commanded by CMC.
NOUN indicators	Indicate noun code entered at keyboard or commanded by CMC.
DATA DISPLAY indicators	Indicate numerical data entered at keyboard or commanded by CMC and sign associated with this numerical data if it is in decimal.

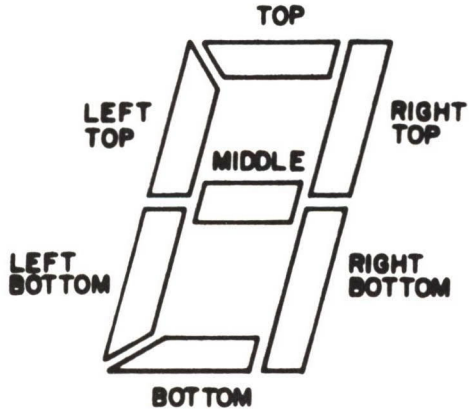
Any digit can be formed on a digit display indicator by lighting a proper combination of electro-luminescent lamps. There are seven of these lamps associated with each digit display indicator and each set of seven lamps is controlled by a set of five bistable relays (figure 9-53. Each sign display is controlled by a pair of bistable relays (one for +, one for -). Setting the proper bistable relays lights the proper sign.

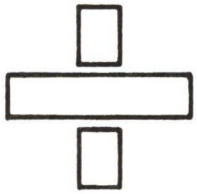
The relays mentioned in the preceding paragraph are controlled by the CMC through output channel 10 (figures 9-54 and 9-55). There are other bistable relays in the DSKY in addition to those mentioned above. These relays are used for various command functions.

The relay matrix in figure 10-61 consists of 14 banks of 11 relays. Bits 15 through 12 of output channel 10 form a four bit code which is sent to a bank of four relay drivers. Each relay driver then generates two logic level outputs for its particular input: (1) the logic level of the input and (2) the complement of the logic level of the input. This eight bit code is sent to a diode decoder when it then supplies a ground to one of the fourteen banks of 11 bistable relays. This bank is uniquely specified by the eight bit code; consequently, the other 13 banks receive no ground. The outputs of the relay drivers associated with bits 11 through 1 of output channel 10 are tied to the set and reset coils of the relays in each of the 14 banks. Therefore, the outputs of the relay drivers associated with bits 11 through 1 of output channel 10 will set or reset those bistable relays in that bank to which ground has been applied.

The relays shown in figure 9-54 are divided into R relays and A relays. The R relays are used in conjunction with the display indicators and condition indicators. The A relays are used in conjunction with the condition indicators on the DSKY and switching associated with downline spacecraft warning and mode indications. Some of R and A relays can be enabled by inputs from other sources than the diode decoder and output channel 10.







RELAYS					DIGIT
R5	R4	R3	R2	RI	DISPLAYED
1	0	1	0	1	0
0	0	0	1	1	1
1	1	0	0	1	2
1	1	0	1	1	3
0	1	1	1	1	4
1	1	1	1	0	5
1	1	1	0	0	6
1	0	0	1	1	7
1	1	1	0	1	8
1	1	1	1	1	9

Figure 9-53. Digit and Sign Displays

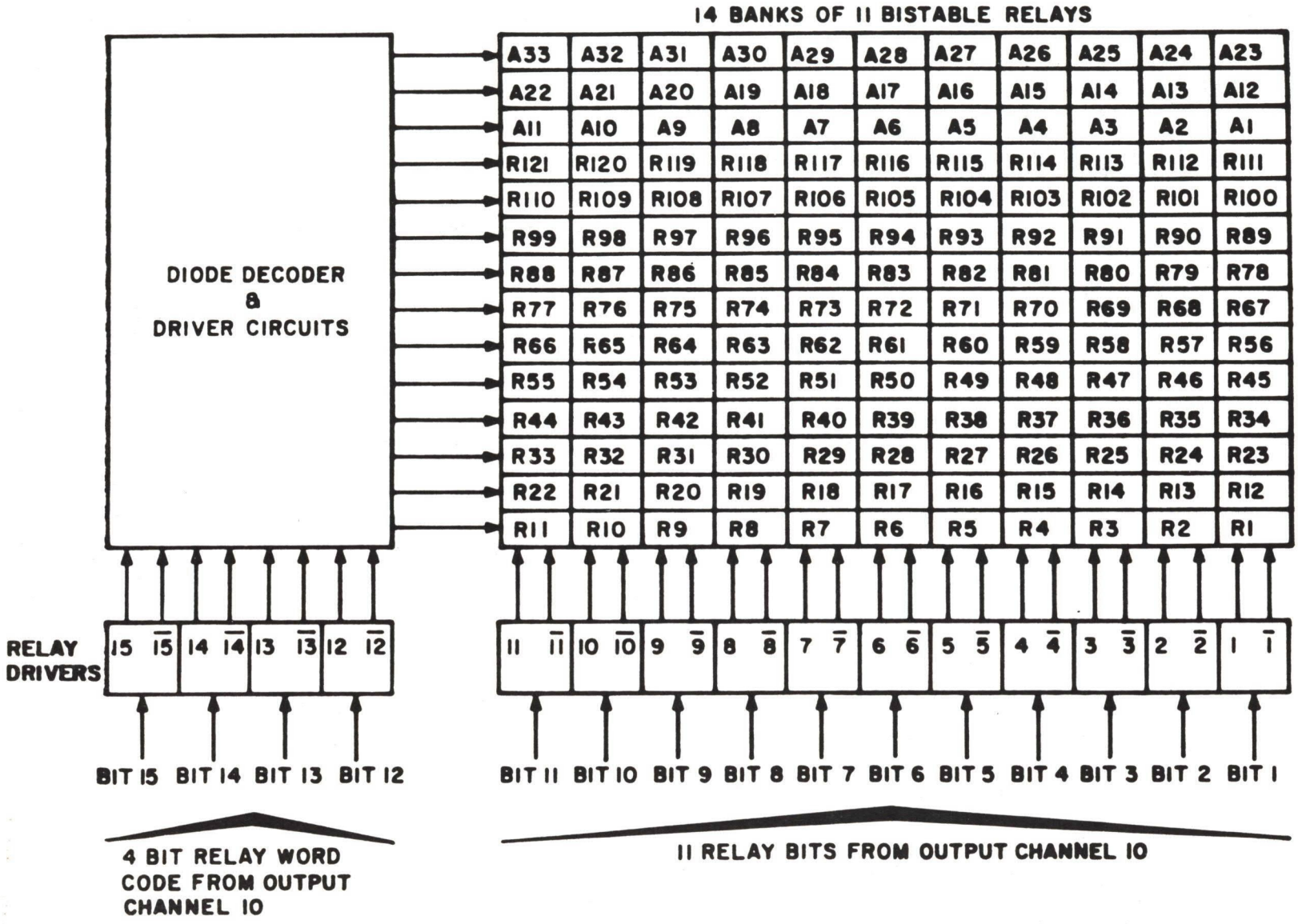


Figure 9-54. DSKY Display and Command Relay Circuitry

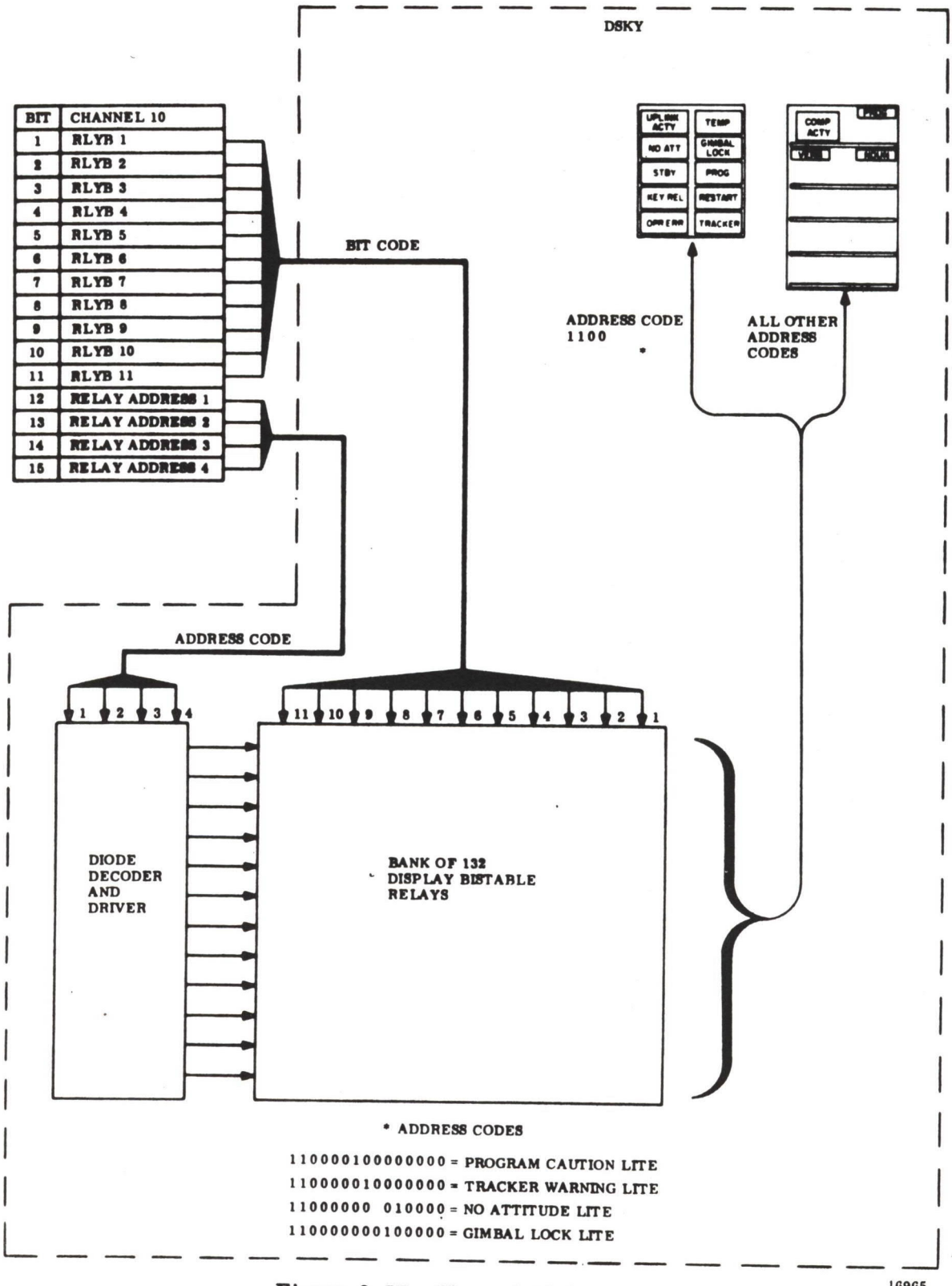


Figure 9-55. Channel 10 Interface

The following is a list of the R relays which are associated with the various display indicators.

R98 - R94	: M1	R82 - R78	: V1	R71 - R67	: N1
R93 - R89	: M2	R87 - R83	: V2	R76 - R72	: N2
R110	: R1S +	R55	: R2S +	R22	: R3S +
R66	: R1S -	R44	: R2S -	R11	: R3S -
R60 - R56	: R1D1	R32 - R28	: R2D1	R5 - R1	: R3D1
R65 - R61	: R1D2	R38 - R34	: R2D2	R10 - R6	: R3D2
R104 - R100	: R1D3	R43 - R39	: R2D3	R16 - R12	: R3D3
R110 - R105	: R1D4	R49 - R45	: R2D4	R21 - R17	: R3D4
R115 - R111	: R1D5	R54 - R50	: R2D5	R27 - R23	: R3D5

9.14 DSKY CONDITION INDICATORS

There are ten condition indications displayed on the DSKY. (Figure 9-56 and 9-57). Table 9-14 lists the indicators and their functions.

The UPLINK ACTY indicator will light if bit position 3 of Output Channel 11 contains a logic 1.

The TEMP indicator will light if:

(1) Bit position 4 of Output channel 11 contains a logic 1, or

(2) Bit position 15 of Output channel 30 contains a logic 0.

The GIMBAL LOCK indicator will light if bit position 6 of Output channel 10 contains a logic 1 and bit positions 15 through 12 of the same channel are 1, 1, 0, 0 respectively.

The PROG indicator will light if bit position 9 of Output channel 10 contains a logic 1 and bit positions 15 through 12 of the same channel are 1, 1, 0, 0 respectively.

The RESTART indicator will light if a GOJAM signal is generated.

The NO ATT indicator will light if bit position 4 of Output channel 10 contains a logic 1 and bit positions 15 through 12 of the same channel are 1, 1, 0, 0 respectively.

The STANDBY indicator will light if the STANDBY circuit is enabled. The indicator will also light if a light test is performed.

The KEY REL indicator will light if bit position 5 of Output channel 11 is a logic 1. This indicator is modulated by the flash signal.

The OPR ERR indicator will light if bit position 7 of Output channel 11 is a logic 1. This indicator is also modulated by the flash signal.

The COMP ACTY indicator will light if bit position 2 of Output channel 11 is a logic 1.

The TRACKER indicator will light if bit position 7 of channel 30 contains a logic 1.

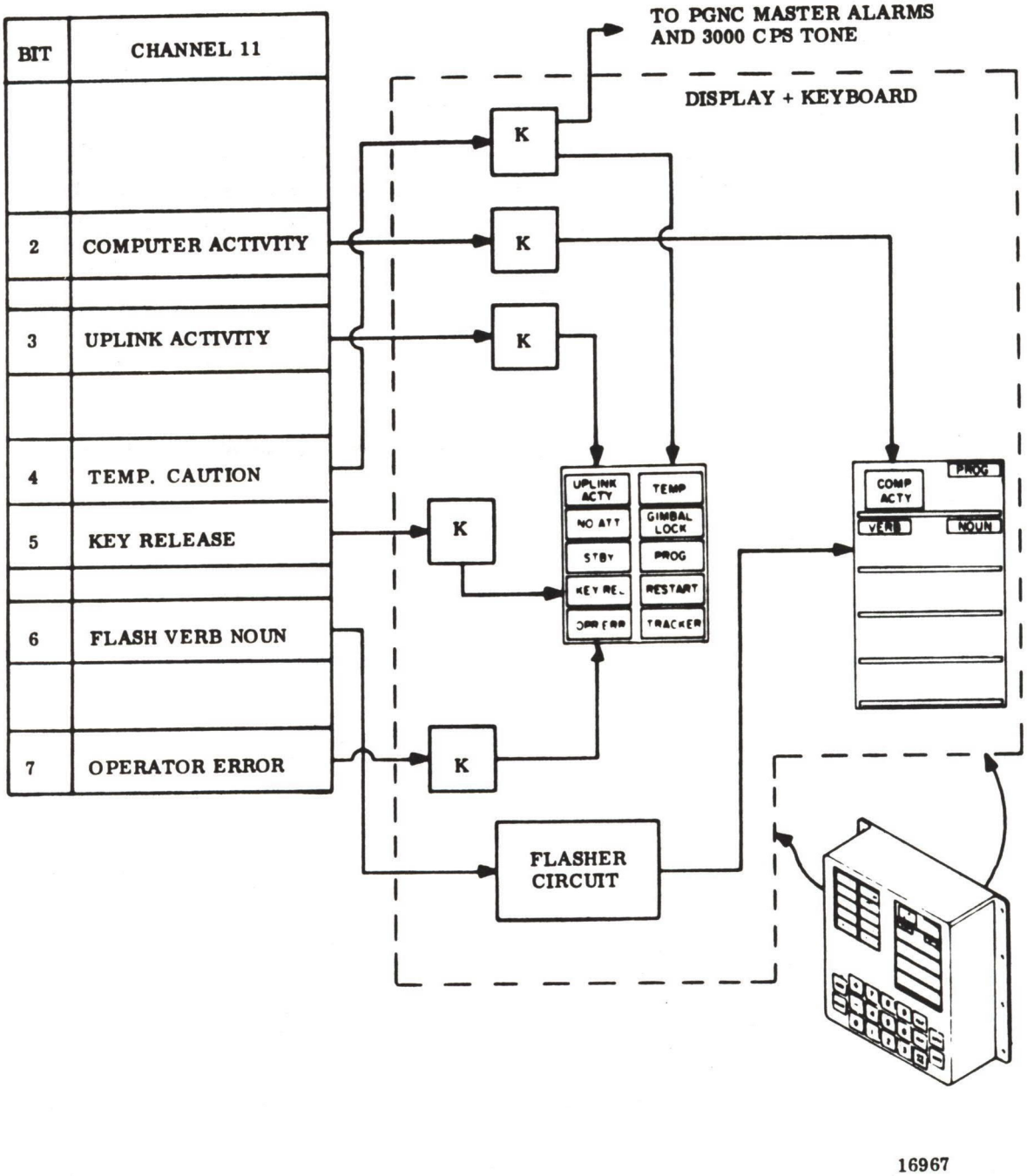


Figure 9-56. Channel 11, Bits 2 through 7 Interface

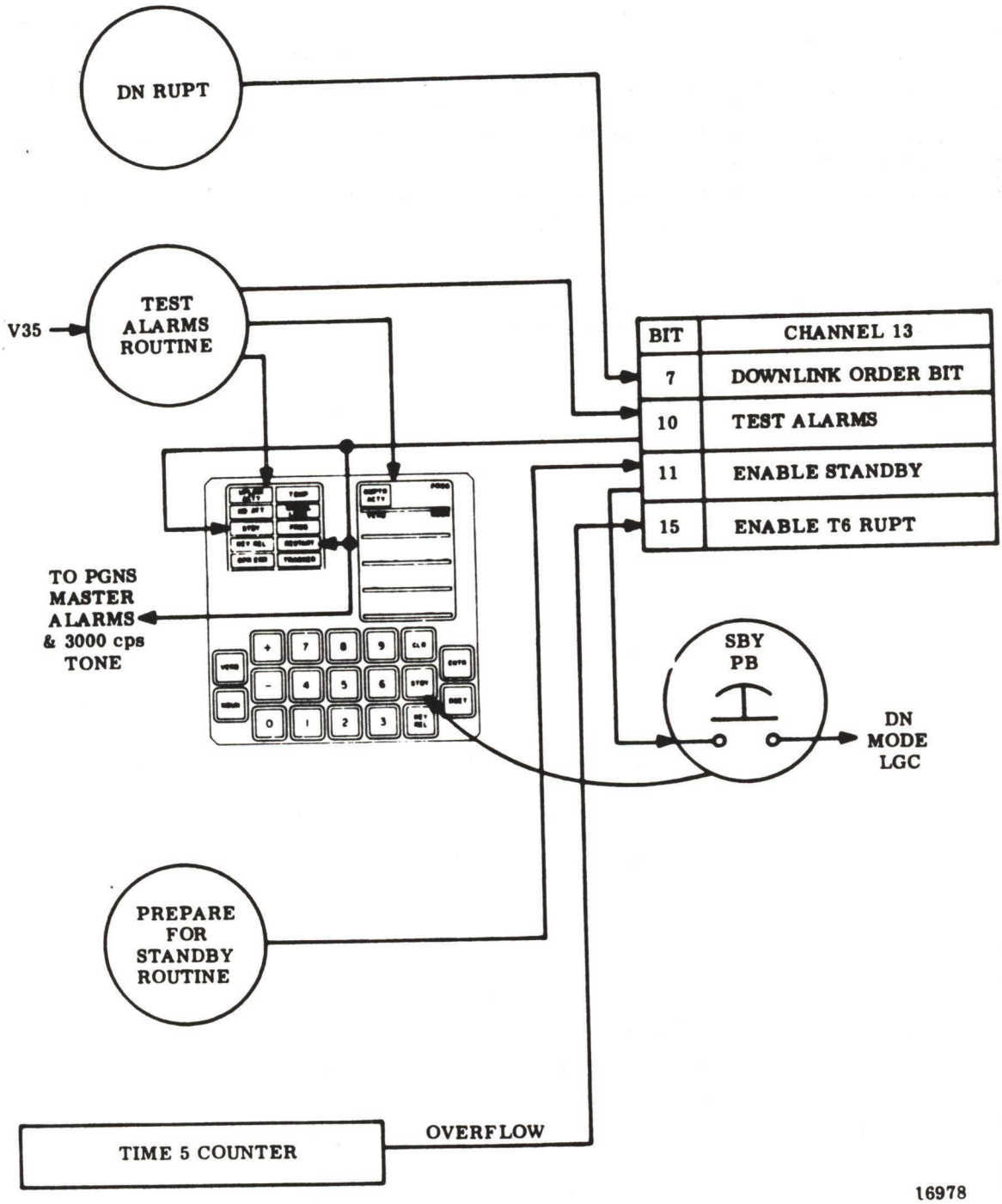


Figure 9-57. Channel 13, Bits 7, 10, 11 and 15 Interface

Table 9-13. DSKY Condition Indicators

Indicator	Function
UPLINK ACTY indicator	Indicates that information is being received via UPLINK.
TEMP indicator	Indicates that the stable member temperature has exceeded its design limit by +5°F.
GIMBAL LOCK indicator	Indicates that the middle gimbal has driven greater than ± 75° from its zero position.
PROG indicator	Indicates that a program check has failed.
RESTART indicator	Indicates: 1. That a word has been incorrectly transferred from memory - Parity Fail. 2. That (a) too many consecutive TC or TCF instructions have occurred or (b) TC or TCF instructions have occurred too infrequently. 3. That interrupts are too long or too infrequent - RUPT Lock. 4. That the CMC has not accomplished a new job within a period whose duration is program dependent and various from .64 to 1.92 seconds - Night Watchman. 5. That a test alarm has been generated by program control.
NO ATT indicator	Indicates that the ISS is not suitable as an attitude reference.
TRACKER indicator	Indicates an optics CDU failure.
STBY indicator	Indicates that the computer is in the standby condition.
KEY REL indicator	Indicates that the CMC program has attempted to use the DSKY and found it busy.
OPR ERR indicator	Indicates that an illegal keyboard operation has been performed.
COMP ACTY indicator	Indicates that the CMC is in a program other than dummy job and that the CMC is not in the standby mode.

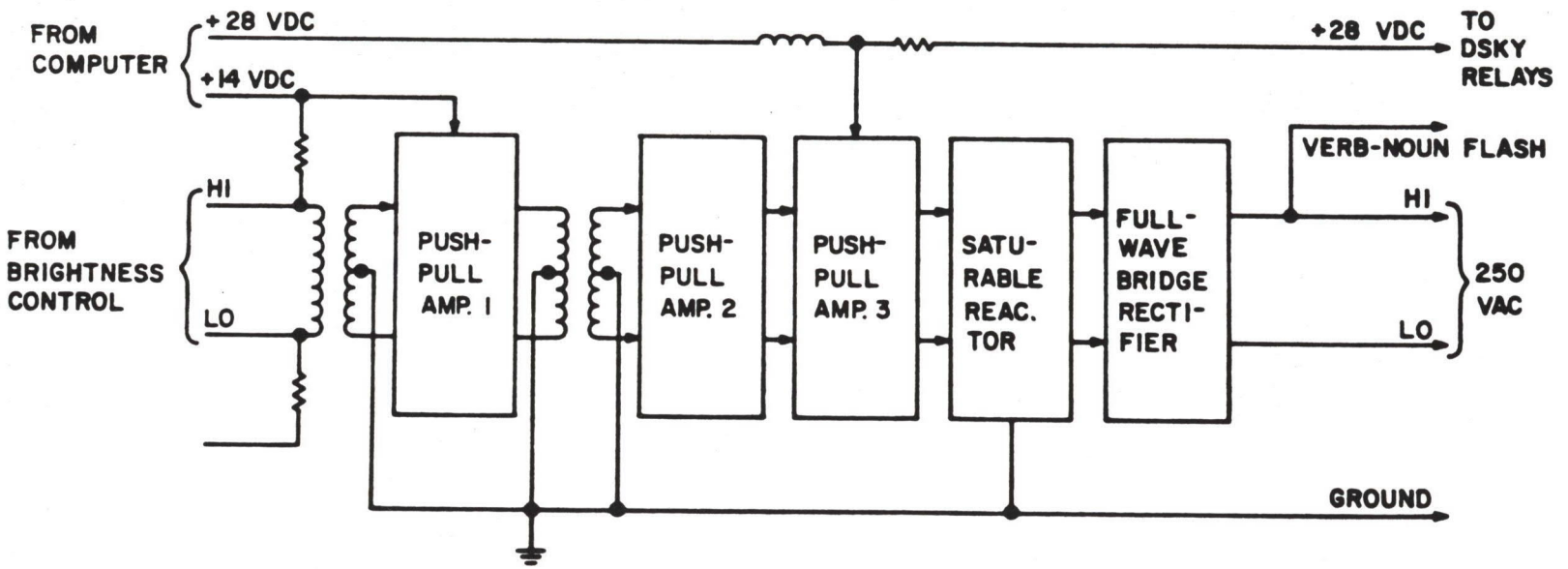


Figure 9-58. DSKY Power Supply

9.15 DSKY POWER SUPPLY

The DSKY power supply inputs are +28 vdc and +14 vdc from the CMC power supply and 800 cps from the CMC timer. The output of 250 volts, 800 cps, is used to light the DSKY display registers.

The power supply (figure 9-58) contains three transformer-coupled, push-pull amplifiers; a saturable reactor; and a full-wave bridge rectifier. The input to the first stage is an 800 cps square wave which varies about a +14 vdc level. The dc level is controlled by the brightness control on the astronauts' control panel. The two transformers step up the voltage applied to their primaries. The three amplifiers also increase the amplitude of the input voltage. The output from the third push-pull amplifier is applied to the saturable reactor.

The saturable reactor and its associated circuitry regulate the voltage applied to the display registers. The displays act as a variable capacitive load which varies as a function of the number of register indicators which are lighted. Changes in the capacitive load are reflected back to the control winding of the reactor through the full-wave rectifier bridge. As the number of lighted indicators increases, the voltage applied to the reactor control winding increases. This increased control winding voltage drives the reactor further into saturation and keeps the output relatively constant. If the load decreases, the control winding voltage decreases and the reactor becomes less saturated.

Computer 28 vdc is filtered and used to operate the DSKY relays.

9.16 SUMMARY

The CMC is a core memory, digital computer which serves as the central processing element of the PGNCS. It permanently stores all the navigation tables, trajectory parameters, programs and constants necessary to solve the guidance and navigation problems. The CMC processes data and issues discrete control signals, both for the PGNCS and other spacecraft systems. It is a control computer with many features of a general purpose computer. As a control computer, the CMC aligns the stable platform of the IMU, positions the optical unit and issues commands to the spacecraft. As a general purpose computer, the CMC solves the PGNCS calculations required for the Apollo mission. In addition, the CMC also monitors the operation of the PGNCS.

The CMC can be functionally divided into a timer, sequence generator, central processor, memory and priority control. The timer provides all of the necessary synchronization pulses to insure a logical flow of data from one area to another inside the CMC. The sequence generator combines basic timing pulses defining action times with the order codes to produce a set of sequential control pulses. The central processor performs the necessary arithmetic and logical data manipulations. The memory area provides a storage capacity of 38,912 (16 bit) words. The priority control time shares the computer to process the different input and output requirements.

The DSKY consists of a keyboard, power supply, decoder, relay matrix, status and caution circuits, and displays.

The keyboard consists of ten numerical keys, two sign keys and seven instruction keys. When a key is pressed, a five bit keycode is entered into input channel 15 of the CMC.

The displays consist of 21 digit displays, three sign displays, and ten condition displays. Any digit can be formed by the lighting of the proper combination of electroluminescent panels. The lighting of these panels is controlled by CMC output channel 10.