SDO6411
JUNE 1964

THE BUNKER-RAMO CORPORATION

340
SYSTEM MANUAL

THE BUNKER-RAMO CORPORATION

340

SYSTEM MANUAL

SDO6411
JUNE 1964

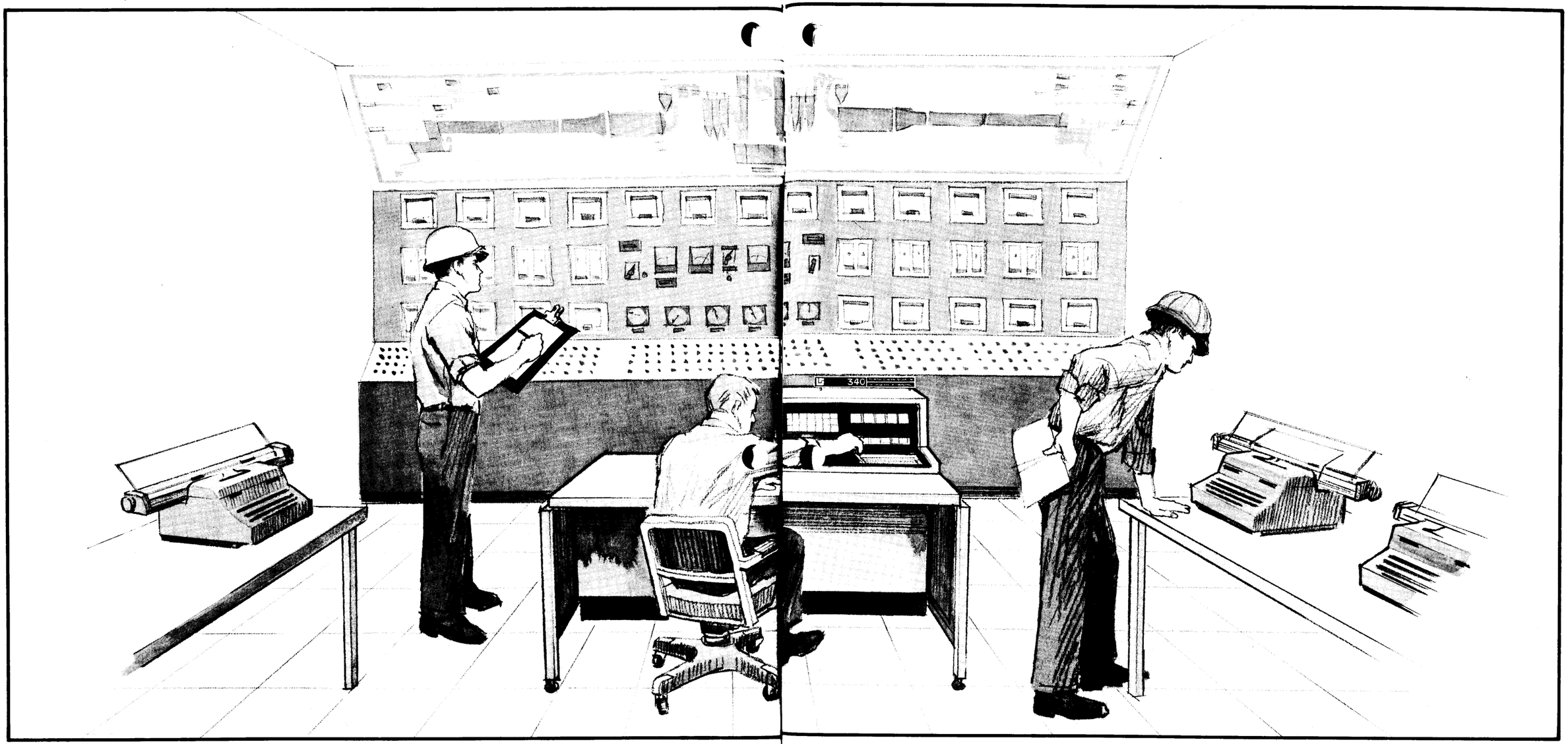
FOR ADDITIONAL INFORMATION, OR FOR THE ADDRESS OF YOUR NEAREST BUNKER-RAMO OFFICE, CONTACT:

INFORMATION SERVICES
PROCESS CONTROL SYSTEMS
THE BUNKER-RAMO CORPORATION
8433 FALLBROOK AVENUE
CANOGA PARK, CALIFORNIA 91304
TELEPHONE 213-346-6000

The Bunker-Ramo Corporation reserves the right to modify this equipment in order to improve its performance and operational flexibility.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS



Typical 340 Control Computer System
(Operations Control Room)

INTRODUCTION

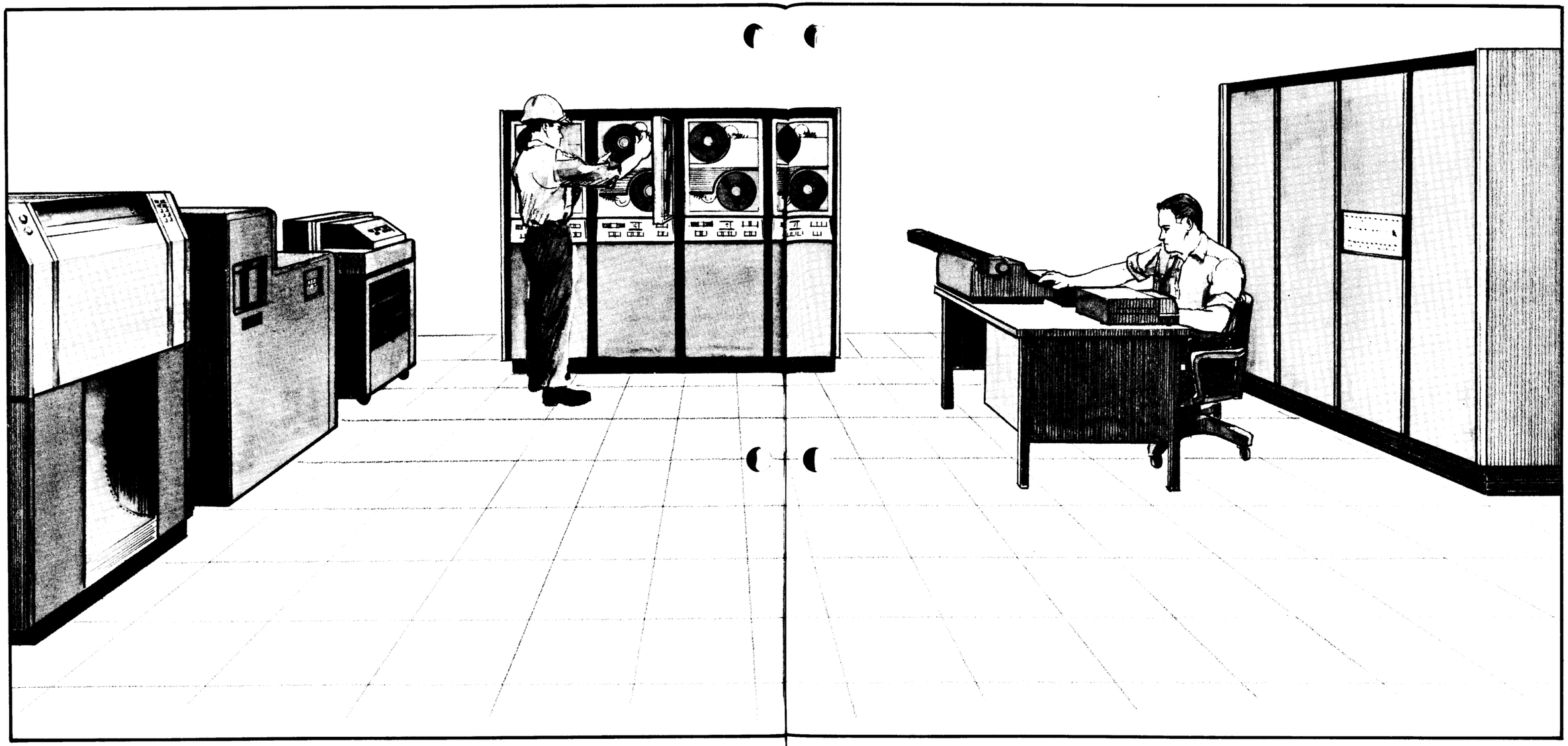
The Bunker-Ramo Model 340 Control Computer System is a 28-bit word, parallel-operated, digital computer designed specifically for use in process control applications. This system combines the most recent advances in computer technology with the present day requirements of high-speed automatic control. Design of the 340 is based on experience gained from successful Bunker-Ramo control computer installations and field-proven techniques.

The 340 employs solid-state components, modular construction, and plug-in circuit cards. This balanced design provides protected and expandable memory, special operations for on-line control problems, and analog and contact input/output flexibility. These and other features simplify and reduce requirements for programming and maintaining a digital control computer system.

Rack-and-panel construction of the 340 equipment permits a system to be assembled in a variety of forms in order to satisfy any process control requirement. 340 systems may be field-expanded as necessary to accommodate an expanding plant or a more extensive control system.

Bunker-Ramo control computer systems are symbolic of the utmost in reliability. Bunker-Ramo computers have operated "on-line" nearly two million hours, and availability records show "uptimes" well in excess of 99 percent.

This manual serves as an introduction to the entire Bunker-Ramo 340 Control Computer System.



Typical 340 Control Computer System
(Equipment and Programming Room)

An industrial process can be operated more efficiently for greater profit under computer control. Even the most experienced operators cannot take into account all the complex interrelationships between process variables, and thus cannot consistently determine the most profitable control settings. The digital control computer, on the other hand, because it can perform thousands of operations a second, makes use of every interrelationship and then adjusts controller setpoints for the most profitable and efficient operation. Furthermore, a computer is not subject to fatigue and distraction that lead to process upsets, yield losses, and substandard product quality. With a computer in command, the process unit performs consistently with predictable uniformity in quality, quantity, cost, and profit.

Since the first Bunker-Ramo control computer went on-line in March 1959 (at Texaco's Port Arthur, Texas, refinery), these computers have closed the loop in industry after industry. By maintaining product quality at minimum cost, Bunker-Ramo computers are returning customer investments, on the average, in less than three years.

SECTION I
SUMMARY OF SYSTEM CHARACTERISTICS

GENERAL

This section summarizes the major characteristics of the 340 Control Computer System and is intended to describe the System in terms of those questions most frequently posed. Information contained in this section is further detailed in other sections of the document.

SYSTEM DESIGN

The 340 Control Computer is not an adaptation of a data processor to provide computer control characteristics, but rather a specifically designed system for process control applications. Its operational structure, accuracy by virtue of a long word length, speed, and advanced input/output philosophy accommodates problems uniquely associated with real-time, on-line process control.

CENTRAL PROCESSOR

A. 28-bit Word Length: This long word length allows a single computer word to contain both the Operation Code and the Operand Address. This provides manipulation of data with 8 digit accuracy and direct addressing of 16,384 words of core memory.

B. Wired-in Instructions: The large complement of logical instructions (approx. 135) permits efficient programming and a minimum of memory accesses to accomplish an operation.

C. Signed Arithmetic: Signs are automatically handled on all arithmetic operations thus relieving the programmer from sign anticipation and corrective subroutines.

D. Double Length Arithmetic Register: The combined A and B registers make up a double length (56 bit) arithmetic register for high accuracy multiplication and division, as well as double precision arithmetic operations.

E. Addressing: Provides direct addressing of up to 16,384 words and block switching of up to 32,768 words. Indirect addressing is a standard feature. The 14-bit Operand Field may also be used for data storage (immediate addressing).

F. Arithmetic Times: The following arithmetic times include one memory access for data. The result is held in the A or combined A and B registers.

1. Signed Addition	12 microseconds
2. Signed Subtraction	12 microseconds
3. Signed Multiplication	68 microseconds (54-bit product)
4. Signed Division	80 microseconds (27-bit quotient)

G. Registers: An accumulator (A), up to six priority interrupts registers (M_n, Q_n), three index registers (I_1, I_2, I_3), plus four major arithmetic registers (B, C, G, X).

Additional non-programmable registers also included are; (N, R, F, FC, V, Z, S, T, U, J, K, L, LS, and Y).

H. Power Failure Protection: The information in core and drum memory is protected against power failure. As power is returned, the 340 either stops or automatically transfers to a restart program. The option is selected by a toggle switch on the maintenance panel.

I. Parallel Register Transfers: All data and instructions are parallel transferred between core memory and registers as well as between respective registers.

J. Solid State Construction: All power supplies and internal circuitry are of solid state construction and DC logic is used throughout.

K. Clock Frequency: Internal functions are synchronous with a 478 kc clock.

CORE MEMORY

A. Word Length: 28 bits plus parity.

B. Type: Coincident current.

C. Read/Write Cycle Time: 6 microseconds.

D. Parity: The 29th bit is used for parity. The parity bit is generated each time a word is stored and odd parity is checked each time a word is read.

E. Expandable: The basic 4,096 word core memory is expandable in 4,096 word blocks to a maximum capacity of 32,768 words.

DRUM MEMORY

A. Size: Drum sizes are available from 16,384 to 98,304 words of storage.

B. Parity: Odd parity is generated and checked each time a word is stored or read.

C. Average Access Time: The average access time for random selection and transfer of one drum word into core memory is 8.3 milliseconds. An immediate transfer mode allows 128 words to be transferred in 16.7 milliseconds.

D. Efficiency of Core/Drum and Drum/Core Transfers: The computer is available 95% of the time during transfers.

E. Memory Protection: Areas of drum memory are inhibited by toggle switches to prevent destruction of stored data by erroneous transfer.

INPUT OUTPUT PHILOSOPHY

Complete tables of data may be transmitted to the appropriate device directly from core memory. A complete list of inputs may be transmitted to memory without interrupting computations. No data passes through the arithmetic registers. Input and output lists may be interleaved so that many I/O functions may take place simultaneously with no attention from the program.

PRIORITY INTERRUPT

The priority interrupt subsystem enables the computer's main program sequence to be stopped in order to; respond to process emergencies, perform critical measurements or calculations, accept or output data, and respond to, leave, and return to other on-and-off-line tasks.

The basic 340 performs the following major interrupt functions:

A. Accepts up to 28 interrupts and determines which is of the highest priority.

B. Determines whether the priority interrupt selected is of higher priority than the program in progress.

C. Either completes the program in progress or branches to the response routine of the highest priority interrupt.

D. Upon completion of C above, returns to any incomplete interrupt response routine or accepts new interrupts of higher priority than incompleted routines.

ANALOG INPUTS

A. Multiplexing: Standard input multiplexing is accomplished through mercury-wetted contact relays. A scanning rate of 100 points per second is considered standard. Scanning rates of 30 and 40 points per second are available using the optional integrator feature.

Special input multiplexing equipment is also optionally available for input sampling at 8,000 points per second.

B. Amplifier: The amplifier is a DC differential amplifier with program selectable gains of 50, 200, 500, and 1000. CMR is 1,000 times the gain setting.

C. Analog-to-Digital Converter: This unit employs a successive approximation method of conversion to a signed, 12-bit number. The conversion rate is 2 μseconds per bit.

D. Signal Types: Thermocouple, RTD, slide-wire, and other voltage and/or current types are acceptable.

E. Thermocouple Inputs: Up to eight types of thermocouples are accepted. Temperature referencing occurs before digitizing. Referencing to plus or minus 0.3°F is provided.

F. Accuracy: Signal conversion accuracy is a function of gain and switching speed, but is normally between 0.1 and 0.5%.

G. Scan and Compare: Analog inputs may be continuously read, compared against high and low limits, and stored with no program intervention. This is a buffered scanning operation.

ANALOG OUTPUTS

A. Resistance-Dividers: Up to 128 resistance-divider analog outputs can be provided with the system.

B. Set Point Stations: Up to 128 set point stations can be provided with the system.

C. Pulse Duration Outputs: Latching relays can be provided to produce pulsed outputs under program control.

CONTACT INPUTS

A. Switching Speed: Input groups of 28 contact inputs may be read and stored every 64 microseconds. The average input speed is greater than 430,000 points per second.

CONTACT OUTPUTS

A. Contact Type: Mercury wetted contact relays provide switch closure outputs.

B. Contact Rating: Standard relays are rated at 100 VA, however, 250 VA relays are also available.

C. Relay Options: Latching or momentary relays are optionally available.

D. Multiple Contact Outputs: Multiple contact outputs are addressable in groups of 28 relays.

PERIPHERAL EQUIPMENT

A. Paper Tape Punch: 110 characters/second.

B. Paper Tape Reader: 300 characters/second.

C. Card Punch: 100 cards/minute.

D. Card Reader: 200 cards/minute.

E. Output Writer: 10 characters/second.

F. Input Keyboard: Photoelectric data input, manual entry.

G. Teletype Printer: 10 characters/second.

H. Magnetic Tape Unit: 15 and 41 kc transfer rate.

I. High Speed Line Printer: 300 lines/minute.

J. Digital Incremental Plotters: Typically 12,000 to 18,000 steps per minute.

POWER REQUIREMENTS

A. Voltage: 115 volts, ±10%.

B. Frequency: 60 cycle, ±3 cycles per second.

C. Power Requirements: Normal System requires 8 K VA.

ENVIRONMENT

A. Temperature: 55-110°F.

B. Humidity: 10-90%.

EQUIPMENT CABINETS

Equipment is assembled in a vertical rack and panel cabinet configuration bolted together to form a convenient installation. Each cabinet is 84 inches high, 23 inches wide, and 24 inches deep. The basic 340 computer requires four such cabinets.

SECTION II DESIGN FEATURES

GENERAL

The 340 Control Computer System incorporates several advanced system design features, several of which are described below.

COMPLETELY BUFFERED INPUT-OUTPUT

The 340 input-output subsystems (analog and contact) are operated in an interleaved manner independent of the computer program. Direct accesses are made to memory as required to process separate data tables for each device or input-output function.

When outputting, tables of packed data for each device are processed in sequence. This enables output devices to be operated at maximum speed. When inputting data, tables are filled automatically. Input-output devices, analog input relays, analog outputs, and, contact inputs and outputs are processed sequentially by direct access techniques providing optimum buffering between slow speed input-output functions and the contents of blocks in core memory. In addition, for immediate access to a group of lines or a single relay, special instructions are provided to bypass the buffering system.

PRIORITY INTERRUPT

A priority interrupt feature enables the computer's main program sequence to be stopped in order to; respond to process emergencies, perform critical measurements or calculations, accept or output data, and respond to, leave, and return to other on-and-off-line tasks.

The basic 340 performs the following major interrupt functions:

A. Accepts up to 28 interrupts and determines which is of the highest priority.

B. Determines whether the priority interrupt selected is of higher priority than the program in progress.

C. Either completes the program in progress or branches to the response routine of the highest priority interrupt.

D. Upon completion of C above, returns to any incomplete interrupt response routine or accepts new interrupts of higher priority than incompleted routines.

DRUM-CORE TRANSFER

Drum-to-core and core-to-drum block transfers are accomplished at the rate of 7,800 words per second (218,400 bits/second). The computer proceeds with the execution of core stored programs while transfers are taking place. The 340 is available for useful computing 95% of the time during actual transfer operations.

HARDWARE RELOCATING OF PROGRAMS

In on-line applications, more than one program may be found competing for a given location in high-speed memory since the sequence of routines to be executed is usually not possible to predict. To avoid writing of one program over the other, or continual reloading from bulk to working storage, the 340 provides hardware facilities to aid in relocating programs. On block transfers between drum and core memory, the 340 automatically shifts programs relative to each other. As a result, any program may be transferred to and executed from any location in core memory.

INSTRUCTION ADDRESS FLEXIBILITY

The programmer is provided with several methods by which data to be operated upon can be obtained, or the location in which data to be stored may be specified. This address flexibility is accomplished through the use of a Normal, Indirect, Immediate Address Lower and Upper, Indexed, X, B, H, Branch, Zero Test, and Mask modes of operation. These modes contribute to efficient memory utilization and fast speeds required in typical on-line programs.

MEMORY PROTECTION

Both drum and core memory power failure protection features are provided. The drum may be selectively protected by toggle switches against inadvertent writing.

INSTRUCTIONS FOR ON-LINE APPLICATIONS

The 340 provides a list of approximately 135 operations, which are specially designed for on-line programs. A number of these are specially designed for process-oriented operations. These operations are available through the use of a special Operation Extension operation. Refer to Section V for more detailed operational information.

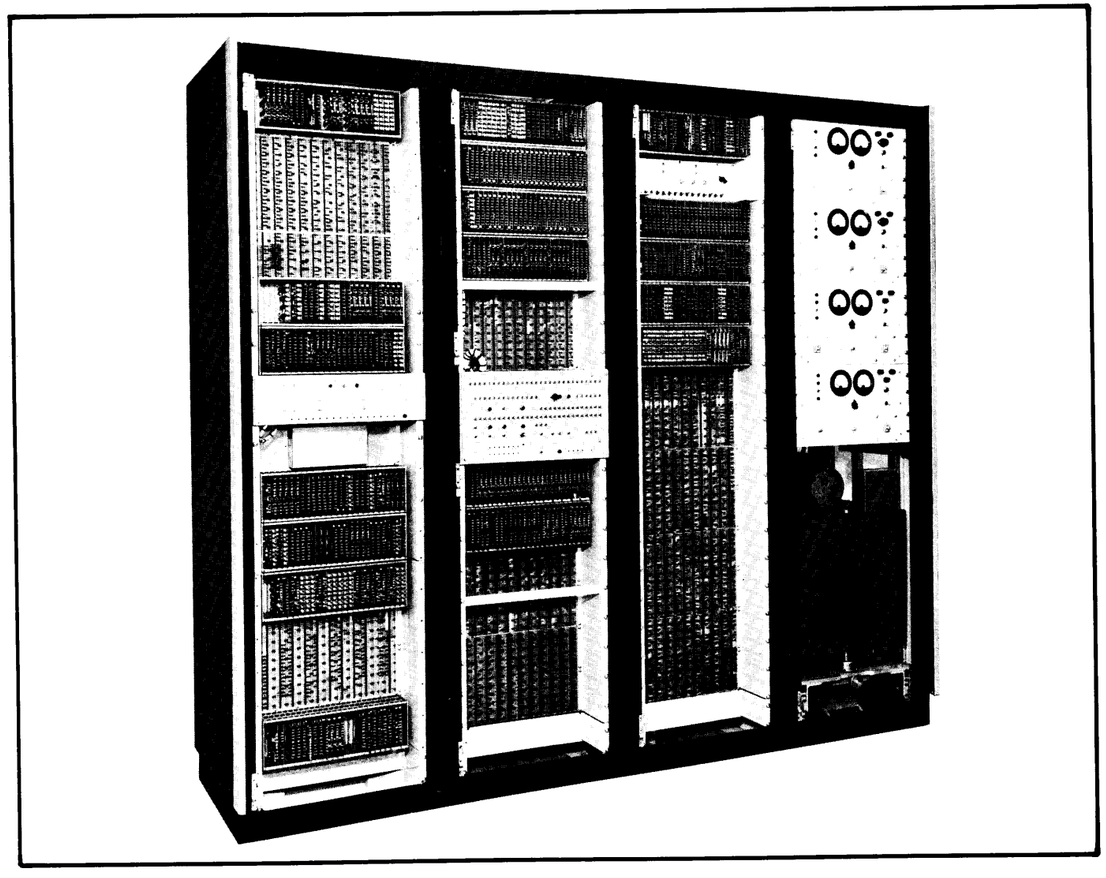
EQUIPMENT EXPANDABILITY

Major components of the 340 Control Computer System are of modular construction for use in fulfilling not only the initial requirements but also facilitate rapid and easy expansion as process control functions change.

SECTION III HARDWARE CONFIGURATIONS

GENERAL

The 340 Control Computer System is designed such that a variety of hardware configurations can be specified for any given application. In order to aid the user in specifying hardware requirements, figure 3-1 (sheets 1 through 3) are included. This set of illustrations depicts the following: 1) Individual subsystems and component options. (Options are shown in the dotted blocks.) 2) Bunker-Ramo model numbers for each integral hardware component. 3) Notations for maximum quantities or restricting parameters. 4) Basic control, signal, and data flow lines between components and subsystems.



Basic 340 Control Computer

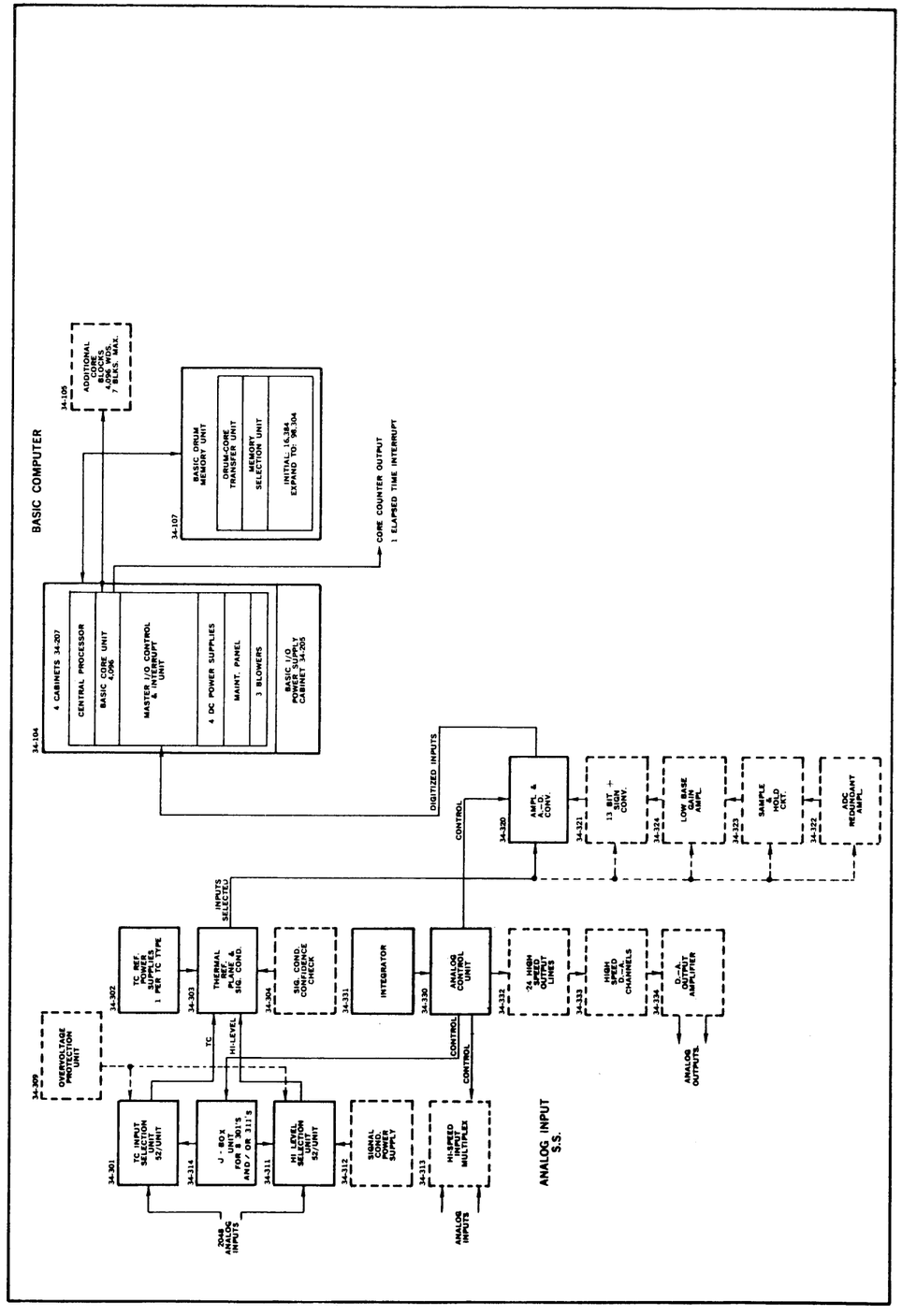


Figure 3-1. 340 Hardware Configurations (1 of 3)

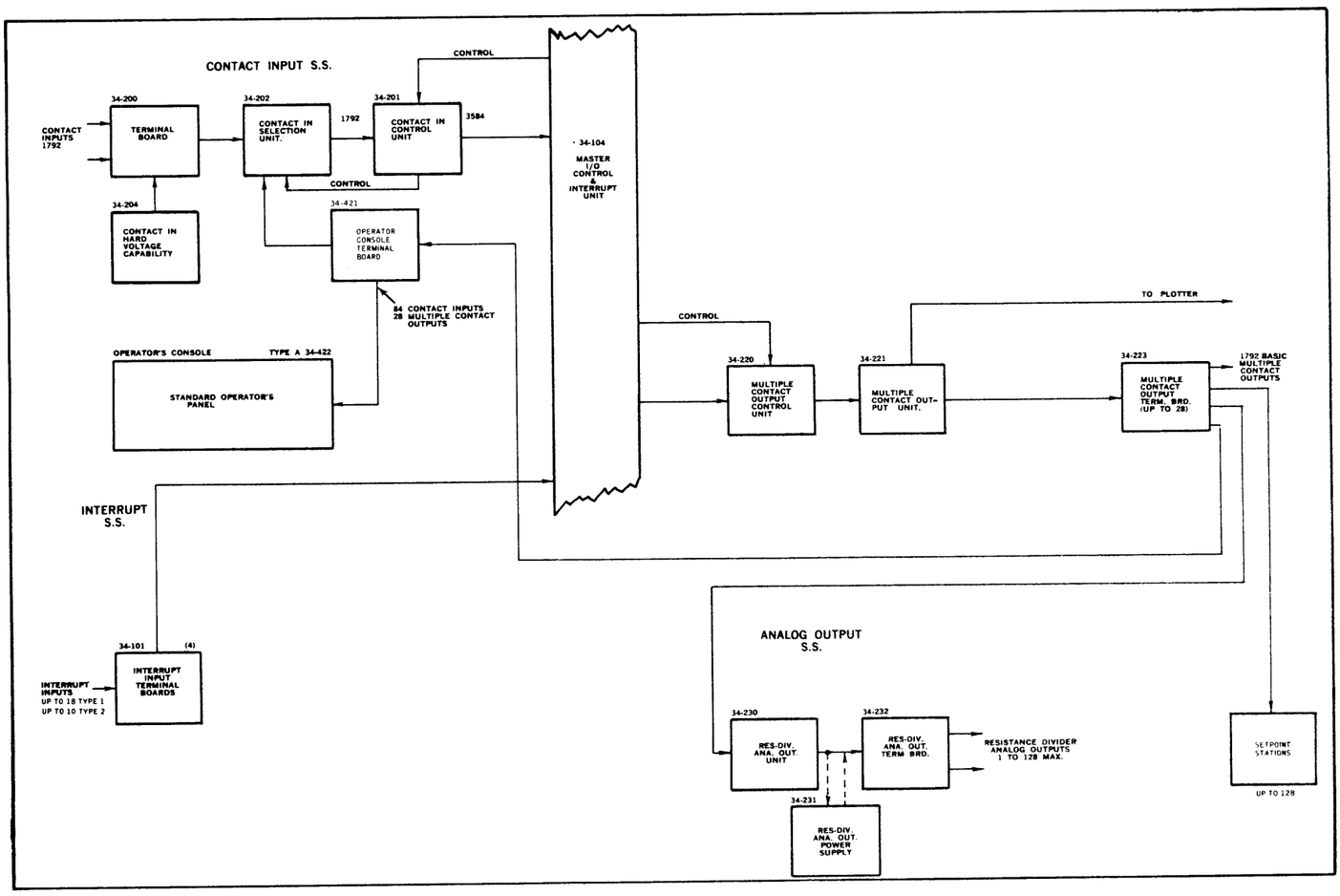


Figure 3-1. 340 Hardware Configurations (2 of 3)

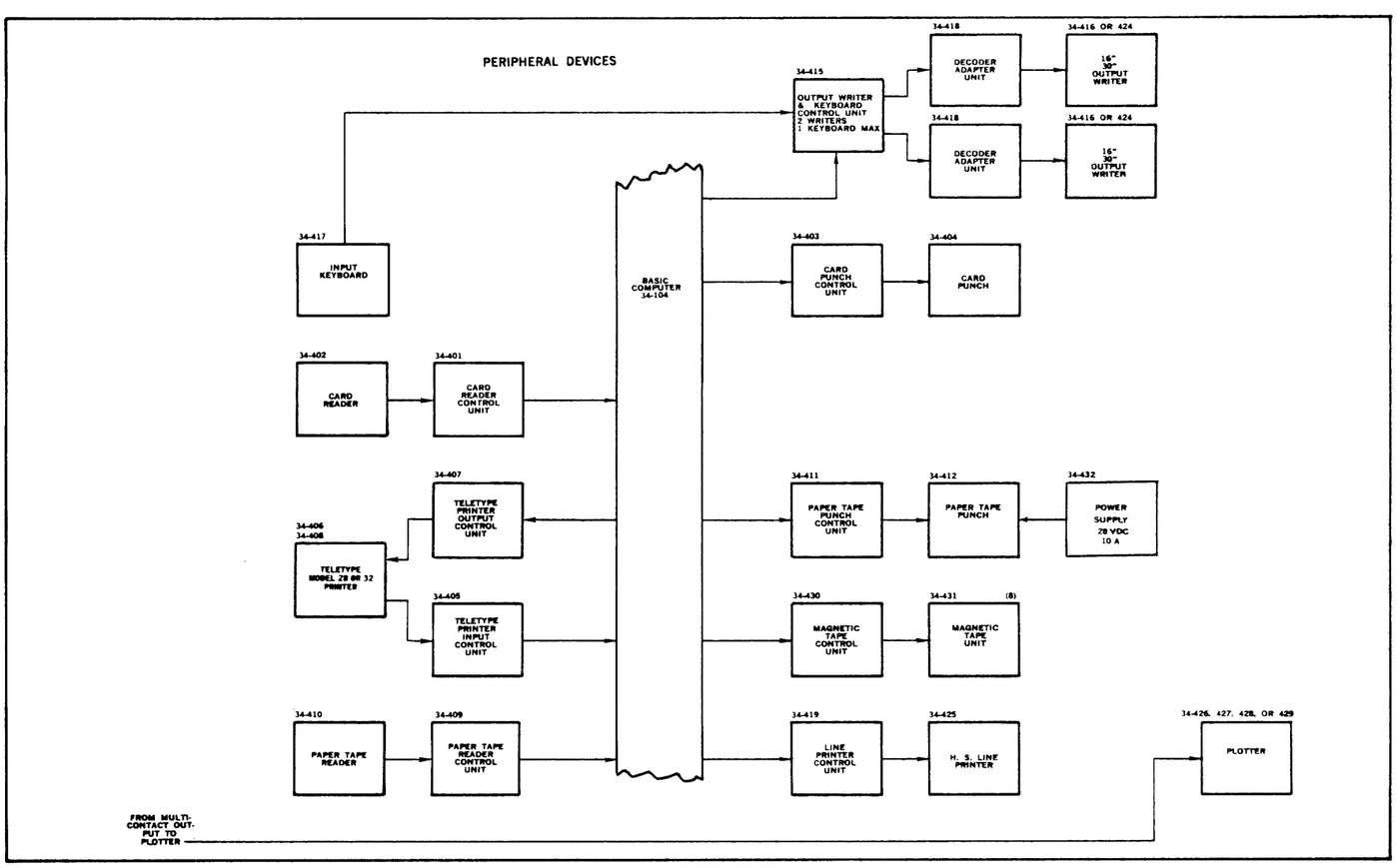


Figure 3-1 340 Hardware Configurations (3 of 3)

SECTION IV SOFTWARE FEATURES

GENERAL

The complete software system, available for the 340, reduces the overall expense and time required to produce real-time system programs. A FORTRAN compiler that accepts FORTRAN II language directly has extended process control oriented capability. Logical bit manipulating and word logic statements make control programming very straightforward. A very capable assembler is provided for computer oriented programs. The On-Line Program Development System allows new control oriented programs or off-line scientific programs written in either FORTRAN or Assembly language to be translated even while the 340 computer is controlling the process. A full range of subroutines and utility routines complement these programming systems providing programmers and analysis personnel with excellent programming aids.

THE 340 ASSEMBLERS

Two assemblers are available for the 340. These are known as MINI and MAX both of which will operate in minimum memory of 4096 words. They are both efficient, one pass assembly systems. The MINI symbolic language is a subset of the MAX language. Both contain symbolic addressing, mnemonic operation codes, pseudo commands, relative addressing, constant declarations, and subroutine calls. They produce relocatable object programs.

THE FORTRAN COMPILER

The FORTRAN compiler operates in 4,096 words of memory and is an efficient one-pass compiler system. It accepts directly "formulae oriented" 1620 FORTRAN II programs without change and has extended logical capability. The logical functions of "AND", "OR", "Exclusive OR" and "Shifting" allow packed binary tables to be used in FORTRAN programs and provide the capability of control-oriented programs to be written in FORTRAN. This can mean a considerable savings in programming expense.

ON-LINE PROGRAM DEVELOPMENT SYSTEM

Programmers are able to develop and check programs using the On-line Program Development System. This work can be performed on the 340 during the normal free periods when a process does not require all of the computer time. These periods may occur momentarily during heavy computer usage or in large periods of time when close control is not required. In any event the programmer, using this facility, may feel that he has an independent computer of his own but one that occasionally runs rather slowly.

The programmer may perform the normal functions of program development. These include assembling or compiling code from source statements, debugging programs not currently associated with the process, and running of programs concurrently with the on-line programs.

The programmers station consists of an Invac Input Keyboard and an Output Writer. A light indicates the status of the keyboard (enabled or disabled) and another light is illuminated when the computer requests use of the Output Writer. All keyboard inputs are monitored on the Output Writer as well as data requested and actions performed. This insures that hard copy is available for any functions performed from the programmers' station.

The same functions may also be performed off-line allowing the station to serve as the main means of programmer and operator access. However, off-line use requires a minimum of 4K of core to be used for assembling, compiling, and station software.

The minimum system complement necessary for keyboard on-line program development includes:

A. Input Keyboard and Output Writer.

B. 4K of core memory in addition to the normal process memory requirements.

C. A high speed paper tape punch and reader, or card reader and punch.

Optional equipment includes:

A. Drum area where available.

B. Card Reader and/or Punch.

C. Line printer.

D. Larger core memory available.

E. Any other equipment which can be assigned specifically for this purpose.

Consider the entries to the computer to be "action" words in that the programmer will be asking for specific actions to be performed.

Generally the actions words and their parameters are input through the keyboard. The Output Writer monitors all action words as well as the actions requested. Provisions are made to allow "automatic" mode where action words and parameters are read through the paper tape input device for bulk jobs that can be set up ahead of time and then left running.

The action words are listed in a table with keys to specify the proper procedures. This table can be added to or deleted from fairly easily, and the procedures are modular so they may be extracted easily.

Suggested action words are:

A. Display

1. Register(s)
2. Memory (core or drum) locations

B. Step (through program X)

1. Step through the program a command at a time and display specified registers and locations including instruction being executed.

C. Execute (a program)

D. Trace

Vertical and Horizontal formatting in snap-shot made.

E. Halt-Proceed

Stop what is going on so programmer may examine output or locations, then proceed (or start something new at discretion of programmer).

F. Load

Input data through specified device as required.

G. Assemble

Enter the Assembler (which must have been previously loaded).

H. Compiler

Enter the FORTRAN Compiler.

I. Enter

1. Insert data to drum or core.
2. Make changes to instructions.

J. Dump

Output from memory in various formats.

1. Binary
2. Octal
3. Decimal
4. Floating Point
5. Mnemonic instruction format.

OPERATIONS CONTROL

The Operations Control System consists of the following three routines:

A. The Executive Routine. This routine provides the basic framework of a system program. This routine keeps time, sets priority of programs and decides when to execute functional programs. The frequency or priority of a given program is readily changed and new functional programs can be added easily. The Executive Routine also organizes the efficient use of drum-core transfer time and core memory space.

B. The Input-Output Routine. This routine coordinates the use of all peripheral equipment by system programs. Messages are queued in tables and transferred to peripheral equipment on a priority basis. Effective use of peripherals and easy input-output programming is provided by this routine.

C. Interrupt Response Routine. This routine insures the organized and effective use of automatic program interrupts. It organizes saving and restoring the contents of all registers in the 340 as interrupt programs are executed.

NON-CONTROL PROGRAMS

The Non-Control Programs are the standard routines that perform scans, logs, alarms, operator responses, and self-checks that are common to most on-line applications. The foundation of a system program can be built with a minimum of effort using these routines.

TEST AND DIAGNOSTIC ROUTINES

Test and Diagnostic Routines check the operation of the TRW-340 subsystems. Core memory, drum memory, arithmetic and control, peripheral input-output, analog input-output, contact input-output test and diagnostic routines insure proper operation and aid tremendously in locating faulty components.

LIST OF LIBRARY ROUTINES

FORTRAN II

TRACE

INPUT ROUTINES

Paper Tape Bootstrap Loader
Manual Entry Binary Loader
Binary Paper Tape Loader Subroutine
Card Bootstrap Loader
Binary Card Loader Subroutine
Card Reader, Binary Mode, Batch

MAX ASSEMBLER

OPERATION EXTENSION

Floating Point Add
Floating Point Subtract
Floating Point Multiply
Floating Point Divide
Square Root
BCD-to-Binary
Binary-to-BCD

OUTPUT ROUTINES

Binary Paper Tape Output Subroutine
Manual Entry Paper Tape Binary Output Subroutine
Binary Card Output Subroutine
Binary Manual Cards
Teletype Punch Binary

GENERAL SUBROUTINES

Sine/Cosine
Cube Root
Arc Tan
Arc Cos
Log/LN
Exponential (e^x)
Double Precision FAD
Double Precision FSU
Double Precision FMU
Double Precision FDV
Double Precision ADD
Double Precision SUB
Double Precision MUL
Double Precision DIV
Float-to-Fix Conversion
Fix-to-Float Conversion

UTILITY ROUTINES

Card-to-Printer
Card-to-Card
Input/Output Translation
Edit and Format Subroutine
Symbolic Dump (to load with MINI)
Memory Dump Typer (Subroutine)
Memory Dump Printer (Subroutine)
Memory Dump Teletype (Subroutine)
Memory Dump Paper Tape Punch (Subroutine)
Memory Dump Card Punch (Subroutine)
Memory Dump Typer Entry Manual
Memory Dump Printer Manual

SPECIAL PURPOSE PROGRAMS

Linear Program I
Linear Program II
Linear Program
Matrix Inversion and Simultaneous Equations Solution
Matrix Inversion and Simultaneous Equations Solution
Simultaneous Equations
Regression Analysis, Stepwise
Polynomial Curve Fitting
Linear Regression Analysis of all Combination of Variables
Exponential Smoothing (Forecasting)
PACK Program

ON-LINE PROGRAM DEVELOPMENT

(Programmers Operating Systems)

OPERATIONS CONTROL

Executive Routine
Input/Output Control Routine
Interrupt Response Routine

SCAN ROUTINES

Analog Monitor
Analog Unpacker/Repacker
Setpoint Coordinator
Limit Interrupt Response

LOGGING ROUTINES

Edit and Format
Packer
Logging
Trends

OPERATOR REQUEST ROUTINES

Request/Cancel Processor
Data Entry
BCD Display
Time Entry
Analog Point Print

MISCELLANEOUS

Vector Manipulators
Self Check
I/O Control Jr.

TEST AND DIAGNOSTIC ROUTINES

Central Processor
Core Memory
Drum Memory
Paper Tape Reader
Paper Tape Punch
Card Reader
Card Punch
Line Printer
TT Printer
Output Typewriter
Keyboard

ANALOG I/O

CONTACT I/O

SECTION V PROGRAMMING FEATURES

GENERAL

The 340 computer incorporates 118 basic instructions. Each of these instructions may be used in one or more different ways known as modes. This feature brings the total usable instruction possibilities (including modes) to nearly 300.

The following descriptions briefly define, in symbolic terms, the normal mode operation of each basic 340 operation. The mnemonic operation code and the normal mode execution time is included with each of the following descriptions. The symbols used in these descriptions are explained below.

DEFINITIONS

The following symbols are used throughout the Model 340 operational descriptions.

Symbol	Definition
CIA	Current Instruction Address
NIA	Next Instruction Address
OPF or OPA	Operand field — Operand Address — the number in the least significant 14 bits of the instruction word. (May be either an operand or the address of a memory location.)
EO	Effective Operand
ER	Effective Register
EOA	Effective Operand Address — the contents of the operand field; or the contents as adjusted by Index and indirect modes as defined by the operation mode.
OEA	Operation Extension Address
n	Specified length of multiply, divide, or shift operations. Specified block length of search or compare tables operations. Specifies number of words or characters to be transferred.
c	Conditional number of places shifted in normalize shifts or number of words searched or scanned.
( )	The contents of the register or memory location indicated.
→	Transfers to, becomes, or replaces
⟷	Exchanged
μs	Microseconds

340 OPERATIONS

LOAD A REGISTER —— LOD A —— 12 μs

(EOA) or (OPF) → (A)

(EOA) or (OPF) remain unchanged.

LOAD B REGISTER —— LOD B —— 12 μs

(EOA) or (OPF) → (B).
(EOA) or (OPF) remain unchanged.

LOAD C REGISTER —— LOD C —— 12 μs

(EOA) or (OPF) → (C).
(EOA) or (OPF) remain unchanged.

LOAD X REGISTER —— LOD X —— 12 μs

$(EOA)_{14-1}$ or (OPF) → $(X)_{14-1}$.
(EOA) or (OPF) remain unchanged.

LOAD INDEX REGISTER 1 —— LOD I1 —— 12 μs

$(EOA)_{14-1}$ or (OPF) → $(I1)_{14-1}$.
(EOA) or (OPF) remain unchanged.

LOAD INDEX REGISTER 2 —— LOD I2 —— 12 μs

$(EOA)_{14-1}$ or (OPF → $(I2)_{14-1}$.
(EOA) or (OPF) remain unchanged.

LOAD INDEX REGISTERS 3 —— LOD I3 —— 12 μs

$(EOA)_{14-1}$ or (OPF) → $(I3)_{14-1}$.
(EOA) or (OPF) remain unchanged.

LOAD INTERRUPT MASK REGISTER M —— LOD M —— 12 μs

(EOA) or (OPF) → effective M bits.
(EOA) or (OPF) remain unchanged.

LOAD INTERRUPT AND GUARD CONTROL REGISTER Q —— LOD Q —— 12 μs

(EOA) or (OPF) → effective Q bits.
(EOA) or (OPF) remain unchanged.

STORE A REGISTER —— STR A —— 12 μs

(A) → (EOA).
(A) remains unchanged.

STORE B REGISTER —— STR B —— 12 μs

(B) → (EOA).
(B) remains unchanged.

STORE C REGISTER —— STR C —— 12 μs

(C) → (EOA).
(C) remains unchanged.

STORE X REGISTER —— STR X —— 12 μs

$(X) \rightarrow (EOA)_{14-1}$.
(X) remains unchanged.

STORE INDEX REGISTER 1 —— STR I1 —— 12 μs

$(I1) \rightarrow (EOA)_{14-1}$.
(I1) remains unchanged.

STORE INDEX REGISTER 2 —— STR I2 —— 12 μs

$(I2) \rightarrow (EOA)_{14-1}$.
(I2) remains unchanged.

STORE INDEX REGISTER 3 —— STR I3 —— 12 μs

$(I3) \rightarrow (EOA)_{14-1}$.
(I3) remains unchanged.

STORE INTERRUPT MASK REGISTER M —— STR M —— 12 μs

(M) → (EOA) effective bits.
(M) remains unchanged.

STORE DATA TOGGLES —— STR T2 —— 12 μs

The bit configuration (setting) of the 28 control panel data toggle switches → (EOA).
Toggles remain unchanged.

MULTIPLY —— MPY n —— 14 + 2n μs

(A) are multiplied by n least significant bits of (EOA) or (OPF). $(A)_{27-1}$ = most significant part of product. $(B)_{27-1}$ = least significant part of the product. A_{28} and B_{28} contain algebraic sign of the product. Length, n, is specified by bits 22 - 26 of the MPY instruction. n may = 0 to 27.

DIVIDE —— DIV n —— 26 + 2n μs

(A) are divided by (EOA) or (OPF). An n bit quotient is developed in the n least significant bits of A. A_{28} contains the algebraic sign of the quotient. B_{28} contains the sign of the remainder and B_{27-1} contains the remainder. Length, n, of the quotient is specified in bits 22-26 of the DIV instruction. n may equal 0 to 27.

ADD —— ADD —— 12 μs

EO is added to (A).
Algebraic sum → (A).
Overflow can occur on this instruction.

SUBTRACT —— SUB —— 12 μs

EO is subtracted from (A).
Algebraic result → (A).
Overflow can occur on this instruction.

HALF WORD ADD —— HWA —— 12 μs

Algebraic sum of (A) plus (EOA) or (OPF) → (A). No carry can be generated from bit 14 to bit 15. Overflow can occur on this instruction.

ADD TO MEMORY —— ADM —— 18 μs

(EOA) added algebraically to (A); the sum is stored in (EOA) and (C).
(A) remain unchanged.
Overflow can occur on this instruction.

SUBTRACT FROM MEMORY —— SBM —— 18 μs

(A) are subtracted from (EOA) and the result is stored in (EOA) and (C).
(A) remain unchanged.
Overflow can occur on this instruction.

DECREMENT INDEX REGISTER 1 —— DIX I1 —— 12 μs

(I1) is decremented by (EOA) or (OPF). The result → (I1).
No indicators are set.

DECREMENT INDEX REGISTER 2 —— DIX I2 —— 12 μs

(I2) is decremented by (EOA) or (OPF). The result → (I2).
No indicators are set.

DECREMENT INDEX REGISTER 3 —— DIX 13 —— 12 μs

(13) is decremented by (EOA) or (OPF). The result → (13).
No indicators are set.

EXTRACT —— EXT —— 12 μs

(A) and (EOA) or (OPF) are compared, bit by bit.
(A) retains a binary one where corresponding bits in both (A) and (EO) are ones.

EXTRACT TO MEMORY —— EXM —— 18 μs

(A) and (EOA) are compared, bit by bit.
(EOA) and (C) retain binary ones where corresponding bits of both (EOA) and (A) are ones.

MERGE —— MRG —— 12 μs

(A) and (EOA) or (OPF) are compared, bit by bit.
(A) retains a binary one where corresponding bits in either (A) or (EO) are ones.

MERGE TO MEMORY —— MGM —— 18 μs

(A) and (EOA) are compared, bit by bit.
(EOA) and (C) retain binary ones where corresponding bits in either (A) or (EOA) are ones.

EXCLUSIVE OR —— XOR —— 12 μs

(A) and (EOA) or (OPF) are compared.
(A) retains binary ones where corresponding bits of (A) and (EO) are different.

COMPARE EQUAL —— COM EQ —— 12 μs

(A) are algebraically compared with (EOA).
If (A) = (EOA), the NIA = CIA + 2.
If (A) ≠ (EOA), the NIA = CIA + 1.

COMPARE NOT EQUAL —— COM NE —— 12 μs

(A) are algebraically compared with (EOA).
If (A) = (EOA), the NIA = CIA + 1.
If (A) ≠ (EOA), the NIA = CIA + 2.

COMPARE GREATER THAN — COM GR — 12 μs

(A) are algebraically compared with (EOA).
If (A) > (EOA), the NIA = CIA + 2.
If (A) < (EOA), the NIA = CIA + 1.

COMPARE LESS THAN — COM LS — 12 μs

(A) are algebraically compared with (EOA).
If (A) < (EOA), the NIA = CIA + 2
If (A) > (EOA), the NIA = CIA + 1.

SCAN EQUAL — SCN EQ — 6 + 6c μs

(A) are algebraically compared with n words of a table beginning with the EOA.
(X) specifies n.
If (A) = table word; NIA = CIA + 2, and (X) specifies the table word address.
If (A) ≠ table word; NIA = CIA + 1, and (X) specified the last table word address.
If (X) initial = 0; (X) = EOA_1.

SCAN NOT EQUAL — SCN NE — 6 + 6c μs

(A) are algebraically compared with n words of a table beginning with the EOA
(X) specifies n.
If (A) ≠ table word; NIA = CIA + 2, and (X) specifies the table word address.
If (A) = table word; NIA = CIA + 1, and (X) specifies the last table word address.

SCAN GREATER THAN — SCN GR — 6 + 6c μs

(A) are algebraically compared with n words of a table beginning with the EOA.
(X) specifies n.
If (A) > table word; NIA = CIA + 2, and (X) specifies the table word address.
If (A) < table word; NIA = CIA + 1, and (X) specifies the last table word address.

SCAN LESS THAN — SCN LS — 6 + 6c μs

(A) are algebraically compared with n words of a table beginning with the EOA.
(X) specifies n.
If (A) < table word; NIA = CIA + 2, and (X) specifies the table word address.
If (A) > table word; NIA = CIA + 1, and (X) specifies the last table word address.

COMPARE TABLES EQUAL — CMT EQ — 6 + 12c μs

The contents of a reference table, the start of which is specified by the EOA, are algebraically compared with the contents of an unknown table, specified by (C). The number of words, n, to be compared are specified in (X). When the contents of a word in the reference table equals the contents of the corresponding word in the unknown table, the NIA = CIA + 2. If the two words are not equal, the NIA = CIA + 1, and (X) = O.

If conditions are met; (A) = contents of ref. table word, (C) = address of unknown table word, $(X) = (C)_{14-1}$, EOA is unchanged.

If conditions are not met; (A) = contents of ref. table word, (C) = address of last word in unknown table, $(X) = (C)_{14-1}$, EOA is unchanged.

COMPARE TABLES NOT EQUAL — CMT NE — 6 + 12c μs

Same as CMT EQ except coincidence is found when a word in the reference table is greater than the corresponding word in the unknown table.

COMPARE TABLES GREATER THAN — CMT GR — 6 + 12c μs

Same as CMT EO except coincidence is found when a word in the reference table is greater than the corresponding word in the unknown table.

COMPARE TABLES LESS THAN — CMT LS — 6 + 12c μs

Same as CMT EQ except coincidence is found when a word in the reference table is less than the corresponding word in the unknown table.

REPLACE A WITH C — RAW C — 6 μs

(C) →(A).
(C) are unchanged.

REPLACE A WITH I1, I2, or I3 — RAW I1 — 6 μs
RAW I2
RAW I3

$(I_i)_{14-1} \rightarrow (A)_{14-1}$.
$(A)_{28-15}$ are set to 0.
(I_i) remain unchanged.

REPLACE A WITH X, HOLD UPPER — RAW XH — 6 μs

$(X)_{14-1} \rightarrow (A)_{14-1}$.
$(A)_{28-15}$ and (X) remain unchanged.

REPLACE A WITH X, CLEAR UPPER — RAW XC — 6 μs

$(X)_{14-1} \rightarrow (A)_{14-1}$.
$(A)_{28-15}$ are set to 0.
(X) remain unchanged.

REPLACE A WITH M or Q — RAW M — 6 μs
RAW Q

(M or Q) → (A)
(M or Q) remain unchanged.

REPLACE A WITH G (RELATIVE ADDRESS REGISTER) — RAW G — 6 μs

$(G)_{14-1} \rightarrow (A)_{14-1}$.
$(A)_{28-15}$ are set to 0.
(G) remain unchanged.

REPLACE A WITH PANEL ADDRESS TOGGLES — RAW T1 — 6 μs

$(\text{Panel address toggles})_{14-1} \rightarrow (A)_{14-1}$.

$(A)_{28-15}$ are set to 0.

REPLACE A WITH PANEL DATA TOGGLES — RAW T2 — 6 μs

$(\text{Panel data toggles})_{28-1} \rightarrow (A)_{28-1}$.

REPLACE C WITH A — RWA C — 6 μs

(A) → (C).
(A) remain unchanged.

REPLACE I1, I2, or I3 with A — RAW I1 — 6 μs
RAW I2
RAW I3

$(A)_{14-1} \rightarrow (I_i)$.
(A) remain unchanged.

REPLACE X WITH A — RWA X — 6 μs

$(A)_{14-1} \rightarrow (X)$.
(A) remain unchanged.

REPLACE M or Q WITH A — RWA M — 12 μs
RWA Q

The logical results of the combination of (A) and (B) and M or Q → (M or Q).
(A) remain unchanged.

REPLACE G WITH A — RWA G — 6 μs

(A) effective → (G).
(A) remain unchanged.

EXCHANGE A WITH B — EAW B — 6 μs

(A) ↔ (B).

EXCHANGE A WITH C — EAW C — 6 μs

(A) ↔ (C).

EXCHANGE A WITH X, HOLD UPPER — EAW XH — 6 μs

$(A)_{14-1} \leftrightarrow (X)$.
$(A)_{28-15}$ remain unchanged.

EXCHANGE A WITH X, CLEAR UPPER — EAW XC — 6 μs

$(A)_{14-1} \leftrightarrow (X)$.
$(A)_{28-15}$ are set to 0.

EXCHANGE A WITH I1, I2, or I3 — EAW I1 — 12 μs
EAW I2
EAW I3

$(A)_{14-1} \leftrightarrow (I_i)$.
$(A)_{28-15}$ are set to 0.

EXCHANGE A WITH M or Q — EAW M — 12 μs
EAW Q

(A) ↔ (M or Q).
Unused bits of M or Q are assumed to be ones.

BRANCH UNCONDITIONALLY — BUN — 6 μs

The condition is always met.
NIA = EOA.

BRANCH IF (A) = 0 — BZE — 6 μs

The condition is met if $(A)_{28-1} = 0$

BRANCH IF A MASKED = 0 — BZM — 6 μs

The condition is met if $(A)_{28-1}$, when masked by $(B)_{28-1}$, = 0.

BRANCH IF A NOT ZERO — BNZ — 6 μs

The condition is met if $(A)_{28-1}$ contains a one bit.

BRANCH IF A MASKED NOT ZERO — BNM — 6 μs

The condition is met if $(A)_{28-1}$, as masked by $(B)_{28-1}$, contains a one bit.

BRANCH IF A IS POSITIVE — BPO — 6 μs

The condition is met if $(A)_{28} = 0$; that is, if $(A)_{27-1}$ is a positive number.

BRANCH IF A IS NEGATIVE — BNG — 6 μs

The condition is met if $(A)_{28} = 1$; that is if $(A)_{27-1}$ is a negative number

BRANCH IF A CONTAINS A LOW BIT — BLB — 6 μs

The condition is met if $(A)_1 = 1$.

BRANCH IF I1, I2 or I3 NOT ZERO — BNX I1 — 6 μs
BNX I2
BNX I3

The condition is met if $(I_i) \neq 0$.

BRANCH IF OVERFLOW INDICATOR ON — BOF — 6 μs

The condition is met if the OVERFLOW indicator is "on" or set to one.
This instruction turns the OVERFLOW indicator "off" or sets to zero.

BRANCH IF CARRY INDICATOR ON — BCY — 6 μs

The condition is met if the CARRY indicator is "on".
This instruction turns the CARRY indicator "off".

BRANCH IF CORE PARITY ERROR — BCP — 6 μs

The condition is met if the CORE PARITY INSTRUCTION and/or the CORE PARITY OPERAND indicators are "on".
Both indicators are turned "off" by this instruction.

BRANCH IF DRUM OR DIRECT ACCESS PARITY ERROR — BDP — 6 μs

The condition is met if the DRUM PARITY and/or the DIRECT ACCESS PARITY indicators are "on"
This instruction turns the DRUM PARITY, DIRECT ACCESS PARITY, and the CORE PARITY OPERAND indicators "off"

SHIFT A LEFT ARITHMETIC — ALA — 8 + 2n μs

$(A)_{27-1}$ are shifted left n places and zeros enter n least significant bits of (A). Sign, A_{28}, remains unchanged
Overflow occurs if a one bit is shifted out of A_{27} when $(A)_{27} \neq (A)_{28}$.

SHIFT A LEFT LOGICAL OPEN — ALO — 8 + 2n μs

$(A)_{28-1}$ are shifted left n places and zeros are entered in the n least significant bits of (A).
No overflow occurs on this instruction.

SHIFT A LEFT LOGICAL CLOSED — ALC — 8 + 2n μs

$(A)_{28-1}$ are shifted left n places and bits leaving $(A)_{28}$ are entered into $(A)_1$ on each shift.
No overflow occurs on this instruction.

SHIFT A RIGHT ARITHMETIC — ARA — 8 + 2n μs

$(A)_{27-1}$ are shift right n places
$(A)_{28}$ remains unchanged but is propagated right, $A_{28} \rightarrow A_{27}$, on each shift.
Bits shifted out of A_1 are lost.

SHIFT A RIGHT LOGICAL OPEN — ARO — 8 + 2n μs

$(A)_{28-1}$ are shifted right n places and zeros are entered in the n most significant bits of (A).

SHIFT A, B LEFT ARITHMETIC — BLA — 8 + 2n μs

$(A)_{27-1}$ and $(B)_{27-1}$ are shifted left n places such that $(B)_{27}$ enters $(A)_1$ and zeros enter $(B)_1$
$(A)_{28}$ and $(B)_{28}$ remain unchanged.
Overflow occurs when a shift out of $(A)_{27} \neq (A)_{28}$.

SHIFT A LEFT ARITHMETIC, B LEFT LOGICAL OPEN — BLL — 8 + 2n μs

$(A)_{27-1}$ and $(B)_{28-1}$ are shifted left n places such that on a shift $(A)_{28}$ remains unchanged, $(B)_{28} \rightarrow (A)_1$, $(B)_{27} \rightarrow (B)_{28}$, and zeros enter $(B)_1$.
Overflow occurs when a shift out of $(A)_{27} \neq (A)_{28}$.

SHIFT A, B LEFT, A LOGICAL, B LOGICAL, OPEN —— BLO —— 8 + 2n μs

$(A)_{28-1}$ and $(B)_{28-1}$ are shifted left n places such that on a shift $(B)_{28} \rightarrow (A)_1$, $0 \rightarrow (B)_1$, $(A)_{27} \rightarrow (A)_{28}$, and $(A)_{28}$ is lost.
No overflow occurs on this instruction.

SHIFT A, B LEFT, A LOGICAL, B LOGICAL CLOSED —— BLC —— 8 + 2n μs

$(A)_{28-1}$ and $(B)_{28-1}$ are shifted left n places such that on a shift $(B)_{28} \rightarrow (A)_1$ and $(A)_{28} \rightarrow (B)_1$.
No overflow occurs on this instruction.

SHIFT A, B RIGHT ARITHMETIC —— BRA —— 8 + 2n μs

$(A)_{27-1}$ and $(B)_{27-1}$ are shifted right n places such that on a shift $(A)_{28} \rightarrow (A)_{27}$, $(A)_1 \rightarrow (B)_{27}$, and $(A)_{28}(B)_{28}$ remain unchanged.

SHIFT A, B RIGHT - A ARITHMETIC, B LOGICAL OPEN —— BRL —— 8 + 2n μs

$(A)_{27-1}$ and $(B)_{28-1}$ are shifted right n places such that on a shift $(A)_{28} \rightarrow (A)_{27}$, $(A)_1 \rightarrow (B)_{28}$, $(B)_{28} \rightarrow (B)_{27}$, and $(A)_{28}$ remains unchanged.

NORMALIZE A ARITHMETIC —— NAA —— 8 + 2n μs

$(A)_{27-1}$ is shifted left until $(A)_{27} \neq (A)_{28}$, or until A = $1400000000)_8$.
On each shift, $0 \rightarrow (A)_1$, (X) - 1 → (X), and $(A)_{28}$ remains unchanged.

NORMALIZE A LOGICAL OPEN —— NAO —— 8 + 2n μs

$(A)_{28-1}$ are shifted left until $(A)_{28} = 1$.
No shift occurs if $(A)_{28}$ is initially a one.
On each shift, $0 \rightarrow (A)_1$, and (X) - 1→ (X).

NORMALIZE A LOGICAL CONTINUED —— NAC —— 8 + 2n μs

$(A)_{28-1}$ are shifted left once only, and continued until $(A)_{28} = 1$ or (X) = 0.
(Shift stops if condition is met on initial shift.)
On each shift $0 \rightarrow (A)_1$, and (X) - 1 → (X).

NORMALIZE A AND B ARITHMETIC —— NBA —— 8 + 2n μs

$(A)_{27-1}$ and $(B)_{27-1}$ are shifted left until $(A)_{27} \neq (A)_{28}$, or until (A) = $1400000000)_8$. $(A)_{28}$ and $(B)_{28}$ remain unchanged.
On each shift $0 \rightarrow (B)_1$, $(B)_{27} \rightarrow (A)_1$, and (X) - 1 → (X).

NORMALIZE A, B, A ARITHMETIC, B LOGICAL, OPEN —— NBL —— 8 + 2n μs

$(A)_{27-1}$ and $(B)_{28-1}$ are shifted left until $(A)_{27} \neq (A)_{28}$ or until (A) = $1400000000)_8$. $(A)_{28}$ remains unchanged and $(B)_{28} \rightarrow (A)_1$.
On each shift, $0 \rightarrow (B)_1$ and (X) - 1 → (X).

NO OPERATION —— NOP —— 6 μs

Hold register status and proceed to NIA = CIA + 1.

STOP AND BRANCH —— STP —— 6 μs

When this instruction is executed, the computer stops, and upon depressing the RESUME button, the program continues to NIA = EOA.

SET RETURN AND BRANCH —— SRB —— 12 μs

Branch to EOA + 1 after saving return address, CIA + 1, in EOA.

SET —— SET —— 6 μs

Provides a means of setting various indicators or functions as specified by 1 bits contained in the OPF.

RESET —— RST —— 6 μs

Provides a means of resetting the same functions specified in the SET instruction.

SET CORE ADDRESS —— SCA —— 6 μs

Specified which 4K block of core is to be used. The positioning of one bits in bits 1 - 4 of the OPF makes this selection. The block addressed is in effect until the next SCA instruction.

ENABLE —— ENA —— 12 μs

Bits 1 - 7 of the OPF specify a device or function to be enabled.
Bits 9 and 10 also specify mode and motor "on" control.

DISABLE —— DIS —— 12 μs

Bits 1 - 7 of the OPF specify a device or function to be disabled.
Bit 10 specifies motor "off" control. A DIS instruction must follow the ENA and output instructions for each device used.

DIRECT CONTACT INPUT —— CIN —— 18 μs

Selects a group of contact input lines to be loaded into the A register. Addresses the group specified in OPF bits 1 - 7.

START DRUM-TO-CORE TRANSFER — SDC — 18 - 24 μs

Specifies a data transfer from drum-to-core. The OPF specifies the address of the first of two or three consecutive core cells which contain; number of words to be transferred, starting core and drum addresses, and address modifier to be stored in (G) register. The operation code field of the instruction also contains data for relativization mode, immediate mode, parity stop, and load (G) indirect.

START CORE-TO-DRUM TRANSFER — SCD — 18 - 24 μs

Specifies a data transfer from core-to-drum. The OPF specifies the address of the first of two or three consecutive core cells which contain; number of words left to be transferred, starting core address of last word + 1, starting drum address of last word + 1, and relative address stored in (G). The operation code field contains the same type of mode data specified in SDC above.

HALT DRUM TRANSFER — HDT — 18 - 24 μs

Terminates a transfer of data from core-to-drum or drum-to-core. The OPF specifies the address of two or three consecutive core cells which, after the HDT, will contain; the number of words yet to be transferred, core address of last word + 1, drum address of last word + 1, and relative address stored in (G) register. The operation code field specifies the type of operation being stopped.

OPERATION EXTENSION — OEX — 12 μs

The operation extension operation code designates not an operation but the address of a subroutine to be entered. Automatic linkage with subroutines and effective operand addresses is provided.

SECTION VI
INTRODUCTION TO 340 SUBSYSTEMS

GENERAL

The 340 Control Computer System is composed of the following major subsystems or parts:

A. Basic Computer, including:
   1. Central Processor
   2. Master I/O Control and Interrupt Unit
   3. Core Memory Unit
   4. I/O Power Supply
   5. DC Power Supplies

B. Memory Subsystem

C. Analog Input Subsystem

D. Analog Output Subsystem

E. Contact Input Subsystem

F. Contact Output Subsystem

G. Priority Interrupt Subsystem

H. Input-Output Peripheral Devices

A simplified block diagram of these major subsystems is shown in figure 6-1.

The following sections describe basic system and subsystem operations and all information is limited to the intermediate subsystem block diagram level. No attempt is made to cover circuit or major component theory of operation. This type of information is beyond the scope of the manual; it is covered in the 340 Theory of Operation and Maintenance Manual.

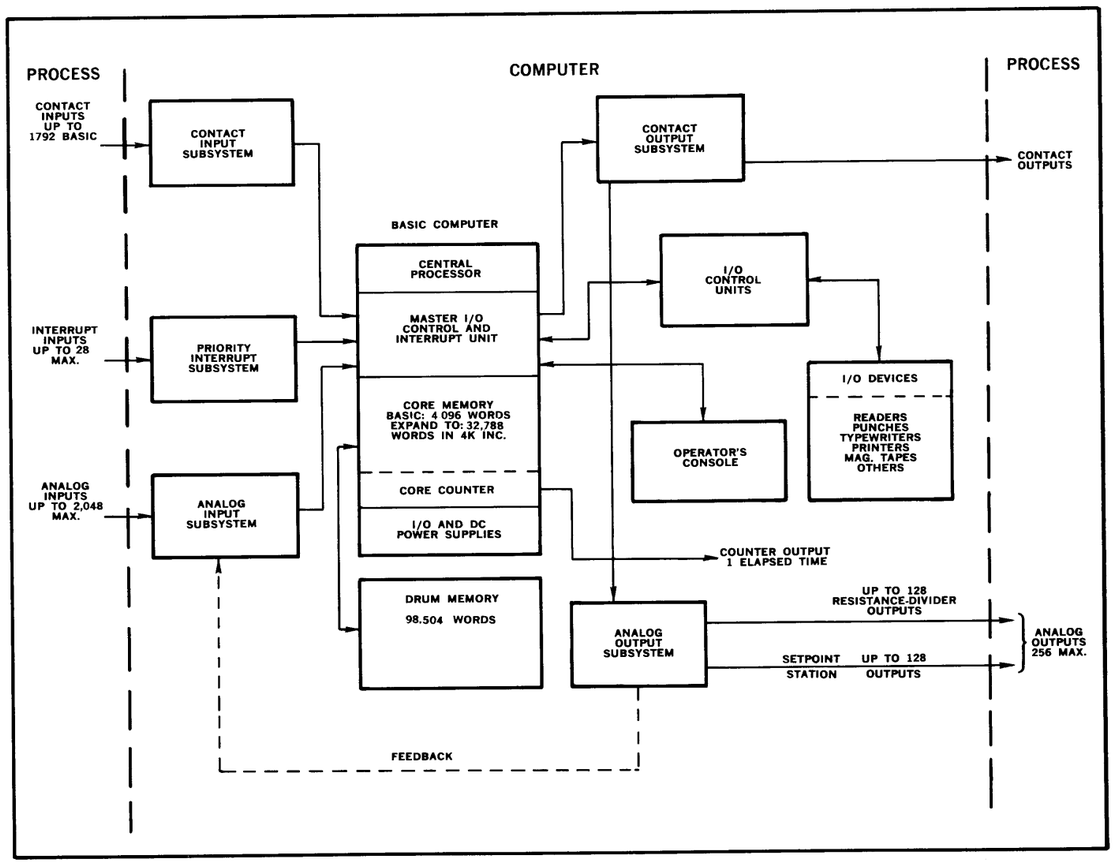


Figure 6-1. 340 System, Simplified Block Diagram

SECTION VII BASIC COMPUTER

GENERAL

The Basic Computer 34-104 performs three primary functions which are: 1) Computing and manipulating digital quantities associated with arithmetic and logical operations; 2) Controlling and directing internal functions related to input-output, core memory, and priority interrupts; and 3) Providing storage space for program instructions, data, and process originated information.

CENTRAL PROCESSOR

The Central Processor accomplishes item (1) of the three basic computer functions listed above. All arithmetic and logical functions handled by this unit are processed at a very high rate of speed; the rate of which is influenced by core memory access time and a very flexible set of program instructions and register operations. Registers involved in Central Processor operations are listed and described in Table 7-1. Figure 7-1 functionally illustrates the major 340 register complement.

In simple terms, as the list of program instructions are examined by the Central Processor control circuits, the appropriate registers and their associated circuits perform computations and logical operations as directed by the program. The instructions, or operations as they are also called, lend themselves to added versatility by making several modes of operation available. Further information on instruction modes and modifiers is given below.

To characterize the function of the Central Processor relating to speed, instruction mode versatility, and register complement, the following paragraphs are given to exemplify typical functions.

A memory access requires 6 µsec for a complete read-write cycle. A register transfer, a full 28-bit addition, subtraction, or shift of one place requires 2 µsec.

A normal mode instruction requires two accesses to memory; one to access the instruction word, and another to access the operand. The execution time is 12 µsec.

In replace, exchange, and branch operations, operands are not used and therefore a second memory access is not required. These operations require only 6 µsec since the operation occurs simultaneously with accessing the instruction. Shifts do not require a second access to memory but do require an additional two µsec if no places are shifted, and 2 + 2n additional µsec where n is the number of places shifted.

Multiply and divide instructions require variable length execution times and a divide instruction requires 12 µsec more than a corresponding multiply. A maximum execution time of 68 and 80 µsec is required for Multiply and Divide respectively.

Search instructions require 6 µsec to access the instruction and an additional 6 µsec for each cell searched. A compare table instruction requires 6 µsec plus two memory cycles for each pair of values compared.

In the index mode, indexing the operand address does not require additional time since decrementing occurs as the address is loaded into the R register.

The indirect mode requires an additional 6 µsec. This memory cycle access is the operand whose address was accessed in the second memory cycle.

In the immediate mode, the lower 14 bits of the instruction are used as the immediate operand. This mode requires only 6 µsec since no memory access is needed to access the operand.

The B mode allows the contents of the B register to be used instead of a cell in memory. The execution time required for this mode is 6 µsec since a true memory access is not required.

The above paragraphs do not describe all modes. Further details related to modes of operation are covered in the 340 Programming Manual.

MASTER I/O CONTROL AND INTERRUPT UNIT

This unit functions as a controller in that it controls and directs the internal computer operations related to input-output, core memory communi-

cations, and priority interrupt handling. To perform these functions six Master I/O Unit electronic registers and their associated circuits are employed. (Refer to Table 7-1 for the complement and use of Master I/O Control and Interrupt Unit registers.)

In order to communicate with I/O functional elements (contact input, contact output, and peripheral devices), a simple set of instructions are programmed. These instructions are classified as contact input, contact output and enable operations. They provide the necessary circuit interconnections to accomplish I/O communications.

This set of instructions performs the following:

A. Specifies the element or device

B. Specifies memory control word locations

C. Specifies memory output or input data table locations

D. Specifies the number of words to be input, output, and/or compared.

The interrupt handling capabilities accommodated by this unit are partially hardware designed and partially programmed. Interrupts are responded to automatically through hardware design but the order by which response is handled is programmed through the M and Q registers. This priority feature, although physically a part of the Master I/O Control and Interrupt Unit, will be covered in the section entitled Priority Interrupt Subsystem.

The most important feature included in this unit is the ability to drive many I/O functions at their maximum rate. This is accomplished through interleaving I/O operations. While one device is outputting or inputting data, the Master I/O Unit examines the status of other I/O devices to determine if any others are capable of accepting or outputting new data. If another device can be utilized, an output or input operation may be initiated, thereby producing interleaved I/O operation of multiple I/O devices. This time-sharing method improves I/O device utilization and enhances system capabilities.

CORE MEMORY UNIT

The Basic Computer includes a basic core memory unit consisting of 4,096 words. This unit may be expanded in increments of 4,096 words up to a maximum of 32,768 words by addition of supplementary core units 34-105. A typical 340 system usually includes the basic 4,096 word core unit plus three additional 34-105 supplementary units for a total of 16,384 words of working core. For purposes of simplifying memory subsystem explanations, the contents of this manual refer to a typical 340 system including core storage capabilities for 16,384 words, or 4 blocks of 4,096 words each. Reasons for this configuration will become evident during the description of core memory addressing and core memory organization which follows in the section entitled Memory Subsystem.

The core memory unit is the working memory of the 340 system. That is, all operations involving direct access to memory are associated with the core. The drum memory unit is used as auxiliary storage only.

I/O AND DC POWER SUPPLIES

Associated with the Central Processor and Master I/O Unit are six dc power supplies. These power supplies deliver the following voltages to internal computer circuits:

A. Central Processor and Master I/O Unit

The Central Processor contains the following supplies:

1. Two +13.5, -13.5, and 4.0 volt dc supplies;
2. One -50.0, -20.0, and +60.0 volt dc supply;
3. One -105.0 and -27.0 volt dc supply.

B. Typical I/O System

The Typical I/O System contains the following supplies:

1. One -13.5, +13.5, and 4.0 volt dc supply;
2. One -50.0, +60.0, and -27.0 volt dc supply.

BASIC COMPUTER REGISTER COMPLEMENT

The following table contains data related to all of the electronic registers contained in the 340 computer.

Table 7-1. 340 Computer Register Data

Register	Length (bits)	Location	Programmable	Function and/or Use
A or Accumulator	28	Central Processor	yes	Major arithmetic working register Use: Arithmetic Logical Branch Extract Merge (A) →core Core →(A) (A) ⟷other registers
B	28	Central Processor	yes	Secondary arithmetic; extension to A register Use: Multiply (holds least significant bits of the product) Divide (holds the remainder) Shift (left or right with A) Temporary Storage Mask (B) →core Core →(B) (B) ⟷(A)
C	28	Central Processor	yes	Temporary storage register Use: Temporary storage Multiply (multiplicand →C) Divide (divisor →C) Replace, search, and compare (C) →core Core →(C) (C) ⟷(A)
I1 I2 I3	14 14 14 (all unsigned)	Central Processor	yes yes yes	Indexing registers Use: Index instructions (modifies operand field) (I_i) →core Core →(I_i) (I_i) ⟷(A) Decremented by any amount $16,384)_{10}$ -- $37777)_8$.
X	14 unsigned	Central Processor	yes	Control counter; count down register Use: Shift (n places shifted) Multiply; (X) = n multiplier bits used Divide; (X) = n quotient bits generated Compare; (X) = n items compared (X) →core Core →(X) (X) ⟷(A)
N	14 unsigned	Central Processor	no	Control register; Next Instruction Address, NIA Register Use: Automatically generates NIA where NIA = CIA + 1 in most operations
R	14	Central Processor	no	Core address register Use: Holds core address in all core information transfers
F	28	Central Processor	no	Intermediate storage register Use: Holds all information being transferred to or from core

Register	Length (bits)	Location	Program-able	Function and/or Use
FC	14	Central Processor	no	Instruction code register Use: Holds operation code of the current instruction address
V	14	Central Processor	no	Temporary storage register Use: Holds starting core address in core-to-drum and drum-to-core transfer operations
Z	14	Central Processor	no	Temporary storage register Use: (Z) = number of words transferred in all drum-to-core and core-to-drum transfer operations
Y	14	Central Processor	no	Intermediate storage register Use: Intermediate storage during compare table operations
M	up to 28	Master I/O Control and Interrupt Unit	yes	Priority interrupt masking registers Use: Mask for up to 28 priority interrupts (M)→core Core→(M) (M)⟷(A)
Q	up to 28	Master I/O Control and Interrupt Unit	yes	Momentary interrupt holding registers Use: Changes up to 10 type 2 momentary interrupts to type 1 continuous interrupts. Controls core extension operations. Q_{4-7}.
J	28 bits	Master I/O Control and Interrupt Unit	no	Control word register Use: Holds the contents of each control word
K	28 bits	Master I/O Control and Interrupt Unit	no	Temporary storage register Use: Holds data being outputted Holds group select table address on inputs Holds results of comparisons on inputs
L	7 bits	Master I/O Control and Interrupt Unit	no	Count register Use: Steps through the 32 different device or function code numbers to determine if any device of function is enabled. (Stops on the function that is ready.) When device is finished, contents continue to count or (search)
LS	7 bits	Master I/O Control and Interrupt Unit	no	I/O device register - temporary storage Use: Holds the code for the I/O device selected
G	14 unsigned	Drum-Core Transfer Unit	yes	Core-drum, drum-core transfer register Use: Modifies (OPA) in drum/core operations (G) ⟷(A)
TS	7	Drum-Core Transfer Unit	no	Temporary storage register Use: Hold drum track-sector address in drum/core operations In d to c operations; (TS) = TTT-SSS pickup address In c to d operations; (TS) = TTT-SSS store address

Register	Length (bits)	Location	Program-able	Function and/or Use
U	28	Drum-Core Transfer Unit	no	Intermediate storage register Use: Holds all words transferred from drum to core

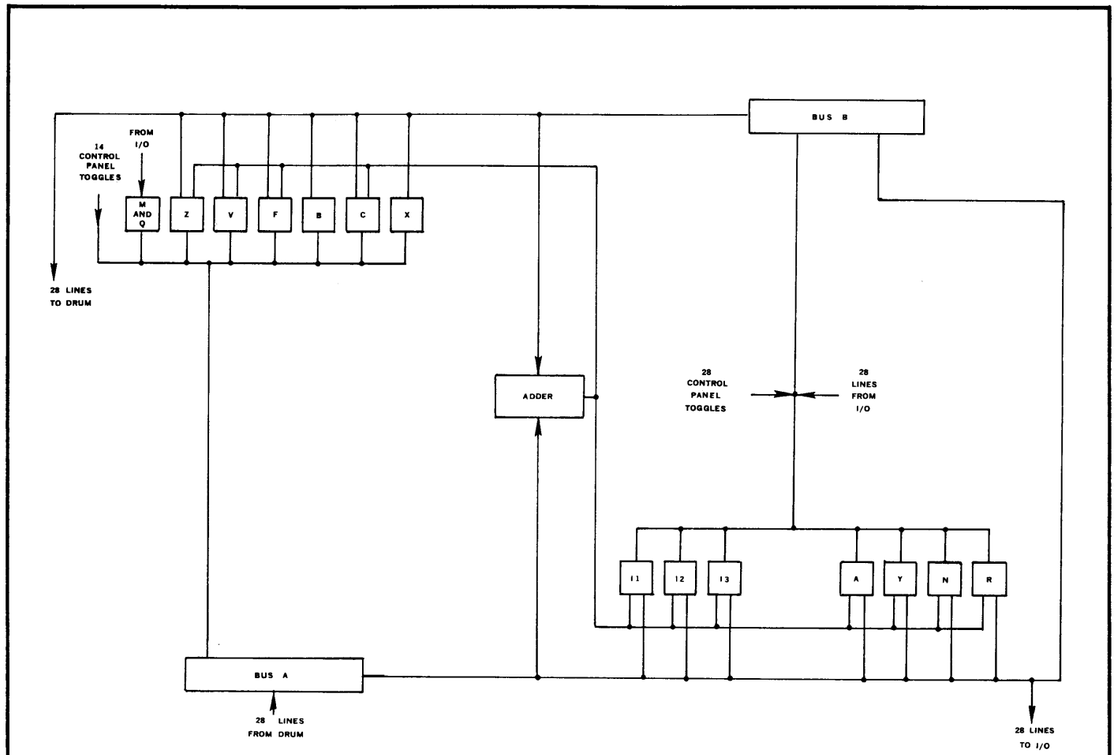


Figure 7-1. 340 Register Data Flow Diagram

MAINTENANCE PANEL

The Maintenance Panel, shown in figure 7-2, is located on the front of the Central Processor cabinet. This panel serves as a mounting for several controls and indicators which are used by programming and maintenance personnel during program checking and equipment maintenance. Major functions provided by this panel include:

A. Data entry

B. Off-line program checkout

C. Circuit troubleshooting and maintenance

D. Program alteration

In addition to the Maintenance Panel, an optional Programming Panel may be included in any 340 system. This panel, usually mounted on the Operator's Console, provides the same types of program-checking features available to the programmer through the use of the Maintenance Panel.

The Maintenance Panel controls and indicators illustrated in figure 7-2 function as follows:

Item 1: These neon indicators display the entire contents of registers selected by the REGISTER SELECTION switch 9.

Item 2: The 14 neon indicators display the 14 most significant bits of the instruction word. These bits contain the computer operation code.

Item 3: The ENTER LOADER pushbutton switch enables the transfer of data from the first 32 words in drum memory to the first 32 words in core. These 32 words make up a bootstrap program which enable the loading of additional data through the use of other panel switches.

Item 4: The 28 DATA and 14 CORE ADDRESS toggle switches within the shaded area permit direct access to core memory. A data word is set into the 28 DATA toggles by placing the toggles in the "up" position, and the address in core is set into the CORE ADDRESS toggles. Then the NON STORE-STORE switch is set to STORE and the CYCLE pushbutton is depressed. The contents of the core address may be displayed on indicators item 1 if the STORE-NON STORE switch is set to NON-STORE, the REGISTER SELECTION switch item 9 is set to F, and the CYCLE button is pressed.

Item 5: The five toggles and three pushbuttons in the shaded area provide: (a) stopping the computer program; (b) stepping through program by instruction, memory cycle, or clock time; (c) restarting computer at origin, at next instruction address, or auxiliary (emergency) origin; and (d) stepping through program from one Branch Unconditional instruction to the next. Provision is made to stop the computer when a core parity error is generated. The programmer may also set the address of an instruction at which to stop on the CORE ADDRESS toggles. By operating the R = CAS switch, the program runs until the instruction address is the same as the setting of the CORE ADDRESS toggles, and then the program stops.

Item 6: The MEMORY CYCLE neon indicators and INCREMENT "N" YES-NO switch are associated with the stepping mode mentioned in item 5 above. The MEMORY CYCLE indicators specify which memory cycle has just been completed. The INCREMENT "N" switch is a maintenance feature which allows repeated execution of the same instruction.

Item 7: The two INTERRUPT toggle switches permit; a) continuous interruption of the computer, b) no interruption at all, or c) normal interrupt operation.

Item 8: This section consists of four neon indicators to display the setting of PARITY ERROR detection flip-flops, two neon indicators which display the states of the O/F (Overflow) and CARRY flip-flops, and a PARITY RESET button which sets all parity flip-flops to zero.

Item 9: The setting of the REGISTER SELECTION switch specifies the register to be displayed on neon indicators 1.

NOTE: The 24 test points at the bottom of the panel — labeled TIMING PERIODS, VOLTAGES, and SYNC — are used only during maintenance procedures for monitoring hardware operation.

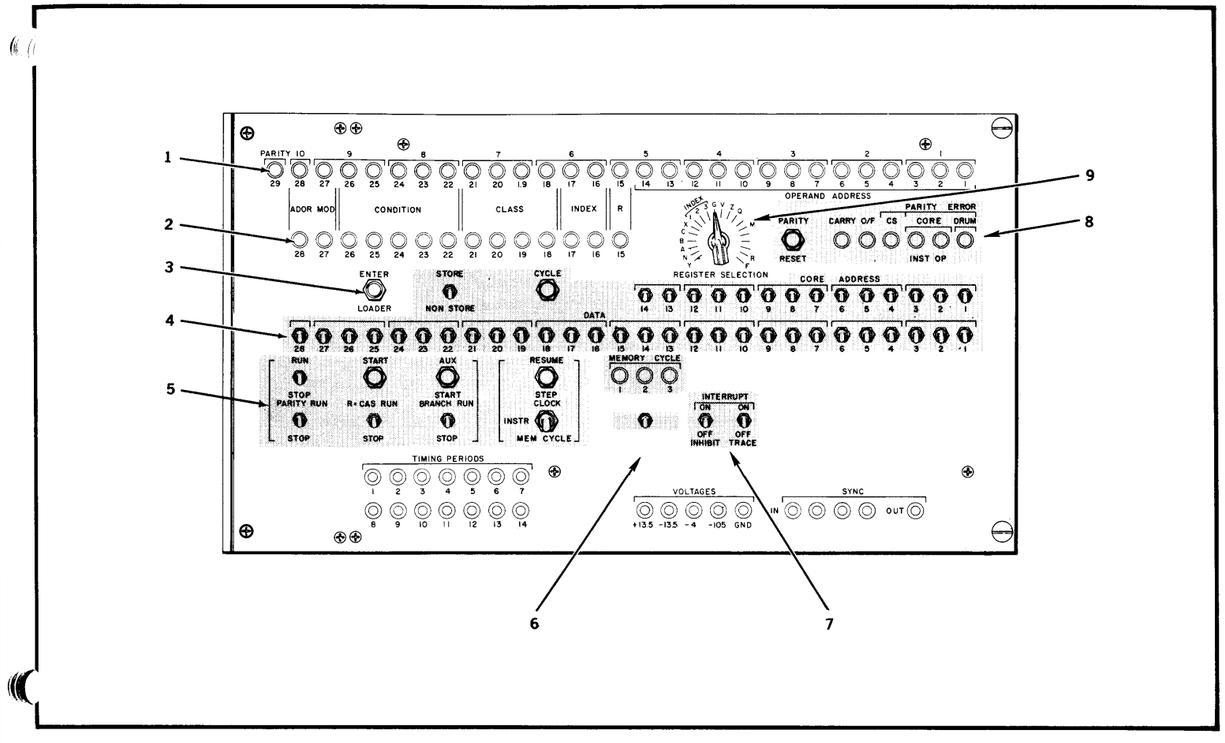


Figure 7-2. 340 Maintenance Panel

SECTION VIII MEMORY SUBSYSTEM

GENERAL

The 340 Memory Subsystem employs both core and drum storage techniques. The core is the direct-access primary working memory; the drum is used as auxiliary memory only.

Normally, both instructions and data are initially placed in drum memory; then transferred from drum to core. Once loaded in core, instructions are executed. Then, under program control, new blocks of drum stored data are transferred to core. All instructions and data which must be preserved for future program use are likewise transferred from working core to the drum. Through this means of data transfer, effective use can be made of minimum size core units and computer time.

The size of the magnetic drum is important only from the standpoint of the amount of information that must be retained. Therefore, specific applications call for specific drum size requirements.

DRUM MEMORY UNIT

Two magnetic drums are available for use with the 340 Control Computer System. A standard drum, 34-107, is designed for a maximum of 384 general storage tracks providing a total of 98,304 words of memory. This drum is normally equipped with 16,384 words of memory. Whatever the size of the initial installation, the standard drum may be field-expanded to maximum capacity in increments of 4,096 words.

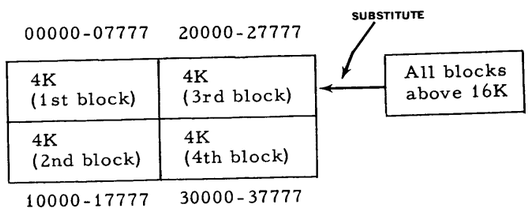
CORE MEMORY ORGANIZATION

Basic organization of the core memory unit begins with the magnetic ferrite core. Individual cores represent individual bits within the word structure and are wired into a core plane configuration shown in figure 8-1A. This core plane consists of an X and a Y axis represented by rows of cores connected in a 64 x 64 matrix. Core planes are constructed into core blocks, shown in figure 8-1B. Core blocks consist of 4,096 twenty-nine bit words of core memory. Note that in figure 8-1C corresponding X and Y bits within each core plane form a 29 bit word (28 bits plus 1 parity bit). Note also that through this matrix configuration each of the 4,096 words may be specified by the X-Y lines chosen. Figure 8-1 depicts X1 and Y1 selecting octal address 00000. Any core cell from address 00000 through 07777 may be chosen or selected to provide 4,096 words of core memory under matrix select control.

Having briefly described the selection characteristics and physical configuration of the basic 4K core memory unit, it is evident that for systems employing more than 4K of core, a more elaborate method of cell selection is required. This is done by assigning a series of octal numbers to each block of 4K core such that the following core assignments are designated for up to 16K.

00000-07777	1st block	4,096 words
10000-17777	2nd block	8,192 words
20000-27777	3rd block	12,288 words
30000-37777	4th block	16,384 words

In systems employing over 16K of core, each 4K block above 16K is substituted for the third block (20000-27777). Selection of the specific block to be used is made under program control. This substitution is known as core extension and the following simple diagram and table 8-1 illustrates the method used.



RESERVED CORE MEMORY (DEDICATED CORE)

Certain core locations are reserved (or dedicated) for specific use in the system and are not available to the programmer for general storage. These dedicated locations are required by the Priority Interrupt System, the Master I/O Control Unit, and as Operation Extension Addresses.

The dedicated Operation Extension Addresses are always the same regardless of the size core involved. However, the Interrupt and Master I/O

Table 8-1. Core Block Address and Q-Bit Extension Designators

Active Block	Q-Bit Core Extension Designators	Location Addresses	
4K	0000	00000-07777	
8K	0000	01000-17777	
12K	0000	20000-27777	
16K	0000	30000-37777	Always substitute
20K	0001	20000-27777	into 3rd block
24K	0010	20000-27777	
28K	0011	20000-27777	
32K	0100	20000-27777	

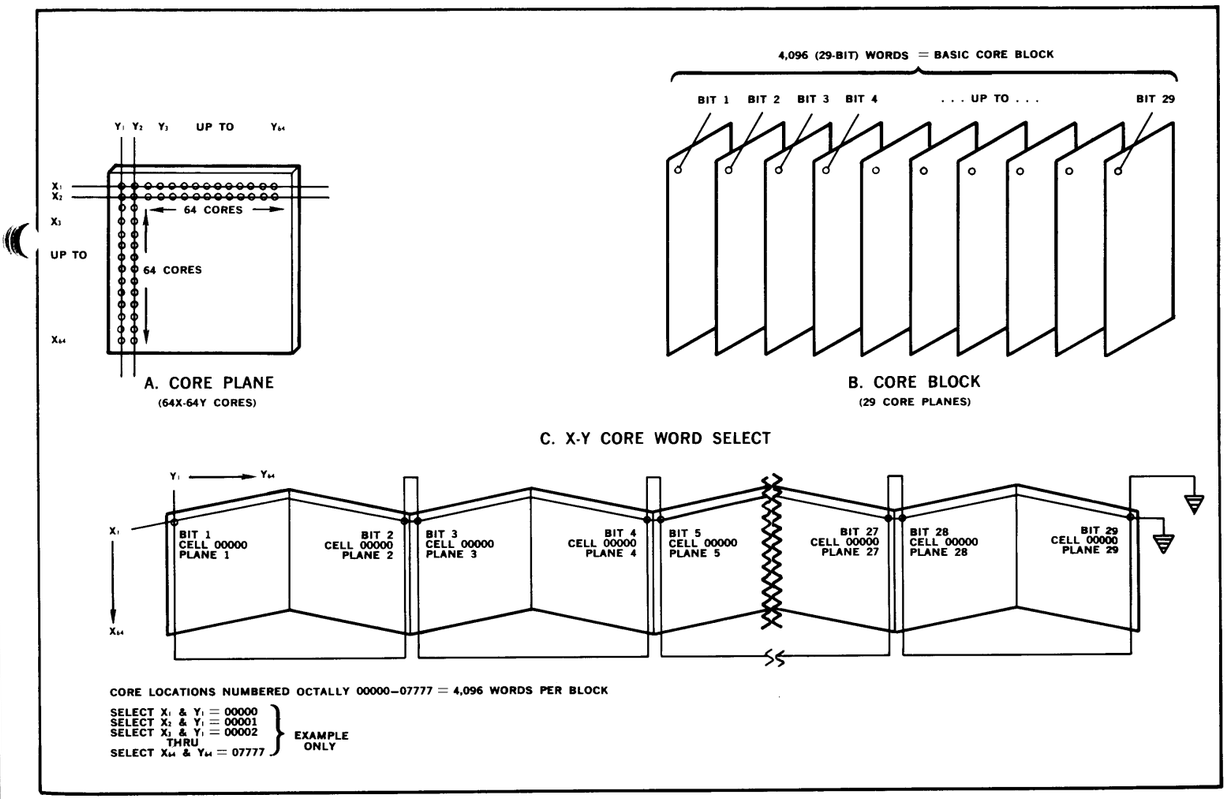


Figure 8-1. Core Memory Organization

dedicated locations "float" as the system expands so they are always positioned in the same relative addresses at the upper end of the memory. Base dedicated locations are given below for Interrupt and Master I/O dedicated cores; for each block above 4K in the system, octal 10000 is added to the base address. Thus if the dedicated location is 7200_8 for a 4K memory, it is 17200 for 8K, 27200 for 12K, 37200 for 16K, and so on.

A. Operation Extension Addresses

$ADDRESS_8$	FUNCTION
200-376	Group 1
400-576	Group 2
600-776	Group 3
1000-1176	Group 4
1200-1376	Group 5

B. Master I/O Addresses

$ADDRESS_8$	FUNCTION
37200	Contact Input Group
37201	Contact Input Compare
37202	Contact Input Data
37203	Contact Output, Single
37204	Contact Output, Multiple, Group
37205	Contact Output, Multiple, Data
37206	Immed. Contact Output, Single
37207	Analog, Relay Select
37210	Analog, Hi-Lo Limit Scan
37211	Analog, Data (Converted)
37212	Time of Day
37213	Core Timer (Elapsed Time Interrupt)
37214	Output Typewriter No. 1
37215	Output Typewriter No. 2
37216	Keyboard No. 1 Input
37217	Card Reader
37220-37377	Other I/O Devices
37470	Contact Input Results (not zero)
37471	Contact Input Group Address
37472	Device Finished

C. Interrupt Addresses

LINE	RE-ENTRY $ADDRESS_8$ (EVEN)	RESPONSE $ADDRESS_8$ (ODD)
1 through 28	37400 through 37466	37401 through 37467

CORE MEMORY POWER FAILURE PROTECTION

The 340 core memory is protected against power failure. Basically, this feature detects a degradation of computer power and stops the computer.

When power returns, the automatic restart of the system program is under control of the POWER ON MODE switch on the maintenance panel.

DRUM MEMORY ORGANIZATION

The drum is divided into basic units called tracks. Each track is divided into either 128 or 256 sectors. By assigning each track a unique identification number, and numbering the sectors within a track from 000 to 177 (octal), any 28-bit word in drum memory may be referenced by giving its track and sector number address. In referring to memory addresses it is customary to list the track first followed by the sector. Thus the memory address 57-132 identifies track 57, sector 132. Drum memory addressing is illustrated in figure 8-2 below.

Selection of the desired track is accomplished through the use of magnetic head matrix selection circuits. The desired sector is selected as a function of certain reserved timing tracks and associated timing circuitry.

RESERVED DRUM MEMORY

One track of the drum is reserved for use by the load program, and therefore, is unavailable for general storage. This track is normally numbered track 000.

DRUM MEMORY PROTECTION

The primary programming and operating considerations for the drum memory subsystem are related to the memory protect feature. As previously explained, blocks of memory may be guarded, thus preventing erasure of previously written data. The drum memory unit employs two-position toggle switches which provide the following guard control functions:

A. Position 1; the area of memory associated with the switch is unguarded and writeable.

B. Position 2; the area of memory associated with the switch is guarded by the switch.

WORD FORMATS

All memory stored data within the 340 computer may be classified as either data or instructions;

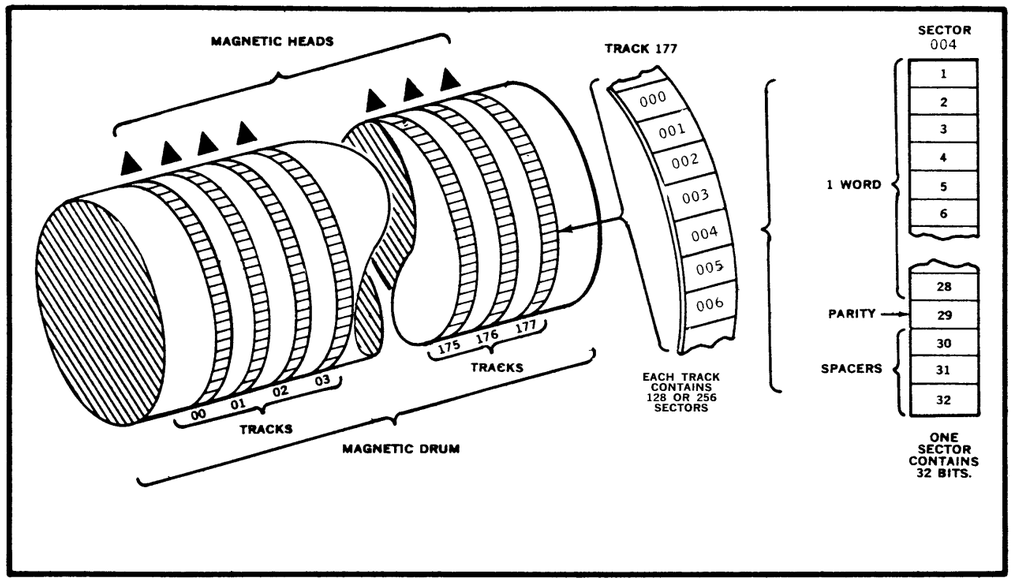


Figure 8-2. Drum Memory Organization

therefore, two separate word formats specify these major classes and are described below.

A. Data Word

The data word, figure 8-3, consists of the magnitude of the number represented in bits 1 through 27, while bit 28 serves as the sign and bit 29 specifies parity.

A zero in the sign position denotes a positive number and one bits are generated in bit 29 to maintain an odd parity configuration.

B. Instruction Words

The 340 instruction word format, figure 8-4, is divided into two basic fields of 14 bits each. Bits 1 through 14 are called the Operand Address Field and bits 15 through 28 are called the Operation Code Field.

B. Instruction Words

The TRW-340 instruction word format, figure 8-4, is divided into two basic fields of 14 bits each. Bits 1 through 14 are called the Operand Address Field and bits 15 through 28 are called the Operation Code Field.

The Operand Address Field (OPA) specifies the core address of an operand or, may also specify the operand itself. The largest operand which can be contained in this field is $37777)_8$. When the octal contents are read from this field, binary bits are grouped as shown in figure 8-4. Bits 13 and 14 specifying the most significant octal digit.

The Operation Code Field, shown below, consists of a relative transfer modifier, an index designator, a basic instruction code, an instruction code modifier, and a mode designator. The relative transfer modifier is used in data transfers between drum and core. The Index designator specifies one of three index registers to be used in modifying the contents of the OPA field. The basic instruction code, as the name implies, specifies a specific type of computer operation, e.g., loads and stores. The instruction modifiers are used to

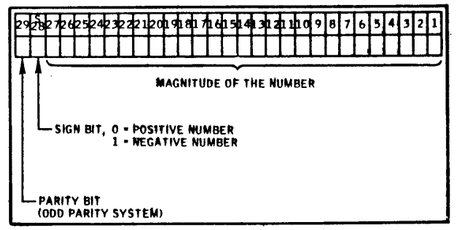


Figure 8-3. Data Word Format

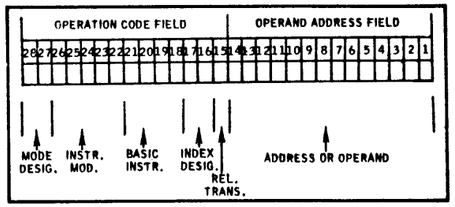


Figure 8-4. Instruction Word Format

modify the basic instruction specified in the instruction code. Finally, the mode designators provide a final variation of the basic or modified basic instruction. The following paragraphs briefly describe the effects of computer operation in each of the available modes.

1. Normal Address Mode (N)

   The normal mode is the conventional direct accessing method for executing instructions. In the normal mode, the Operand Field of the instruction word specifies the base address of the memory location from which the Operand is obtained (or stored). This base address is the Operand Address. If indexing is specified, this base address is modified by the contents of an index register to give the Effective Operand Address (EOA).

2. Indirect Address Mode (I)

   In the indirect mode the Operand Field of the instruction specifies the base memory location which contains the actual Operand Address to be used by the instruction. If the indirect mode is indexed, the base address is modified before the memory access is made.

3. Immediate Address Lower Half Mode (L)

   In this mode of Arithmetic, Logical and Load instructions, the Operand Field is interpreted as the Effective Operand of the instruction. In executing a lower half mode instruction, the 14 bits of the Operand Field operate on the lower 14 bits of the specified register or memory location. (In store instructions, the upper 14 bits are unchanged. In Store Data Toggles, the toggle modes affect the upper and lower portions of the EOA.)

4. Immediate Address Upper Half Mode (U)

   This mode operates like the lower half mode, except that the 14 bits of the Operand Field operate on the upper 14 bits of the specified register or memory location. (In store instructions, the upper 14 bits are unchanged. In Store Data Toggles, the toggle modes affect the upper and lower portions of the EOA.)

5. B-Register Mode (B)

   In the B-register mode the operand is taken from or stored into the B Register. Because no memory access cycle is needed, this mode requires less time to execute than the normal mode.

6. X-Register Mode (X)

   The X-register mode is used with Branch instructions to flag the instruction to pick up the branch address from the 14-bit X-register rather than from the Operand Field.

7. H-Mode (H)

   The H-mode is available only on the Divide instruction to indicate that the contents of the combined A and B registers are divided by the Effective Operand.

8. Branch Mode (Br)

   The branch mode is available on the Replace and Exchange instructions. In the branch mode, the next instruction address is taken from the Operand Field.

9. Mask Modifier (M)

   A mask modifier bit is provided in many operations in order to cause the Effective Operand or the designated register transfer to be masked by the contents of the B register. Masking is a bit-by-bit logical operation in which the only bits affected by the operation are those where the corresponding bit in the B register is 1.

10. Zero Test Modifier (Z)

    The zero test modifier operates to cause a conditional branch to be executed based on the results of the operation. When the result of the operation is zero, the Next Instruction Address (NIA) is the Current Instruction Address (CIA) plus 2. When the condition is not met, the NIA is the CIA plus 1.

11. Indexing (I1, I2, I3)

    Three 14-bit program controlled Index registers are provided for address modification. When an instruction is tagged to use one of the I registers, the Effective Operand Address is the base Operand Address as specified in the Operand Field less the contents of the designated I register.

SECTION IX ANALOG INPUT SUBSYSTEM

GENERAL

The 340 Analog Input Subsystem automatically selects and converts analog quantities (continuously varying voltages) from process-connected measuring instruments into digital form. These digital quantities representing analog voltages are then either: 1) stored in core memory, 2) compared against pre-stored high-low limits, or 3) both compared and stored.

The basic analog input equipment configuration allows sampling of up to 2,048 analog signals at rates up to 100 points per second. This sampling speed is defined as the maximum 340 relay input sampling rate. Slower sampling rates may be specified. Optional hardware is available to permit time integral noise rejection for the input signals. This mode is under program control and limits the sampling speed to 30 or 40 points per second. Optional hardware is available to permit sampling at the rate of 8,000 points per second; however, solid state input circuitry must be substituted for the conventional mercury-wetted type input relays supplied with the basic subsystem. Optional hardware is also available to provide a high speed (8000 outputs/sec) output capability which would be suitable for communicating with solid state D-A units.

SUBSYSTEM HARDWARE

Figure 3-1 (1 of 3) is included to aid in defining subsystem hardware configurations. Frequent reference to this diagram will be necessary in the brief description of the hardware below.

The Analog Control Unit 34-330 communicates with the Master I/O Control and Interrupt Unit within the Basic Computer 34-104. The control unit functions to: 1) select one of 2,048 input relays within the respective Input Selection Units 34-301 and 34-311 through J-Box 34-314, 2) select one of four gain settings available with Amplifier 34-320, and 3) select integration or normal mode (if option is included). The control unit is also capable of selecting a fourth high-speed input sampling rate in systems employing solid state input circuitry. In high-speed input systems (up to 8,000 inputs per second), a High-Speed Multiplexer Unit 34-313 is required. For high-speed output systems, optional units 34-332, 34-333, and 34-334, must be specified.

The 34-320 A to D Converter is a Redcor designed 12-bit-plus sign bipolar amplifier/converter. This device employs gains of 50, 200, 500, and 1000. Gain control is a function of programming and is exercised through the analog control unit. The A to D Converter accepts input signals in the ranges shown below from the Thermal Reference Plane and Signal Conditioner 34-303 and converts these quantities into their digital counterparts to within the system accuracies.

System accuracy is as follows:

Input Signal Range	Sampling Speed	Accuracy (% of Full Scale)
0 to +200 mv	10-100 pts./sec.	+0.05%
0 to ± 50 mv	10-100 pts./sec.	±0.075%
0 to ± 20 mv	10-100 pts./sec.	±0.1%
0 to ± 10 mv	10- 50 pts./sec.	±0.1%
0 to ± 10 mv	50-100 pts./sec.	±0.15%

Higher level inputs may be tied into the system but will be scaled down to the zero to ±200 mv range before being presented to the A-D unit.

An optional Integrator Unit 34-331 may be employed to provide averaging and noise rejection for analog signals as follows:

A. At 30 pts./sec. — 256 conversions each sample.

B. At 40 pts./sec. — 16 conversions each sample.

Integration techniques are not employed at the normal 100 pt./sec. input sampling rate.

The Thermal Reference and Signal Conditioner 34-303 contains compensating thermocouple cold junctions for each type of thermocouple used in the system. Thermistor input bridges measure the ambient temperature of the plane and generate compensating voltages in series with the input voltage. This provides the necessary correction to a standard reference temperature.

Thermocouple Reference Power Supplies 34-302 provide excitation for the cold junction compensating bridges. Each supply is confidence checked by programming special inputs. Signal conditioning supplies 34-312 provide necessary excitation for other inputs such as RTD's, slidewires, strain gauges, etc. These supplies may be confidence checked by using unit 34-304. An extra analog in-

put point is required to confidence-check each of these power supplies.

Units 34-321, 34-322, 34-323, and 34-324 are optionally available equipment which may be used in conjunction with the A to D Converter 34-320. These options provide: 13-bit conversion, sample and hold circuits, back-up amplification techniques, and additional and/or nonstandard gain settings.

The Analog Input Overvoltage Protection Unit 34-309 is also optionally available to provide protection against equipment damage due to the presence of excessively high voltages or currents on analog input lines. Protection is provided against AC overvoltages to 250 RMS and to DC overvoltages to 200 volts. Voltage and current limiting techniques allow short duration transients to be limited without input disconnect. Fusing provides complete disconnect for the presence of continuous overvoltages.

NOTE: One analog input should be reserved for each motor-driven potentiometer analog output.

SIMPLIFIED SUBSYSTEM OPERATION

The following paragraphs provide a simplified operational description of the Analog Input Subsystem. As previously mentioned, the Analog Input Subsystem provides three separate functional features. These are:

1. Sampling specified inputs and storing the desired inputs in core memory.
2. Sampling specified inputs and comparing the sampled inputs against pre-stored limits.
3. Sampling specified inputs, comparing the sampled data against pre-stored limits, and storing the sampled data.

In order that these three functions can be accommodated, a special instruction and combined function code is reserved for each. The instruction involved is the ENABLE instruction (ENA) and the code representing this instruction appears in bits 15-28 of the instruction word shown in Figure 9-1. Bits 1 through 14 of the instruction word contain the function code which specifies one of the three functions listed above. Refer to Figure 9-1 and note that an octal code of 007 in bits 1-14 indicates that the analog input(s) are to be sampled, stored, and compared. A code of 010 in bits 1-14 specifies that the analog input(s) are to be sampled and compared only. Finally, an octal code of 011 specifies that the analog input(s) are to be sampled and stored only.

After the computer detects an enable instruction for either of the three analog input functions, the computer examines the contents of two or three core locations for further information. These core locations are known as dedicated core cells and are specific core addresses. The dedicated core locations for each function are listed below.

ENA 007 — dedicated core
37207, 37210, and 37211

ENA 010 — dedicated core
37207 and 37210

ENA 011 — dedicated core
37207 and 37211

For example, if the ENA instruction specifies a 007 function, the computer examines cells 37207, 37210, and 37211 since an 007 function specifies a sample, store, and compare operation. If the ENA function is 010, the computer examines only 37207 and 37210 since an 010 function specifies a sample and compare only operation. The 011 function involves cells 37207 and 37211 since an 011 function specifies a sample and store only operation.

All parameters relating to input addresses, store table addresses, number of inputs to be sampled, compare table addresses, etc., are defined in Figures 9-2, 9-3, and 9-4. These illustrations show the type of data contained in each of the three dedicated core cells as well as the associated data tables. Dedicated core locations and data tables must be loaded and/or set up prior to execution of the ENA operation.

To summarize Analog Input Subsystem operation, refer to Figure 9-5. The diagram depicts analog digitized process input routing for proper signal conditioning, relay selection, compensation, amplication, and signal conversion. Note that (1) control of signal conditioning and relay selection is handled through communications between the Analog Control Unit and the Master I/O Controller block, and (2) the Master I/O is activated by the ENA instruction executed in the Central Processor.

When the Analog Input Subsystem is operating in the Compare With Limits Mode and an alarm condition is found, the following takes place:

1. An alarm interrupt (type 2) is generated.
2. The value of the out-of-limits variable is stored in memory.
3. The Analog Input Subsystem is disabled.

After the alarm condition is attended to, the system is re-enabled by the response routine for the analog alarm interrupt.

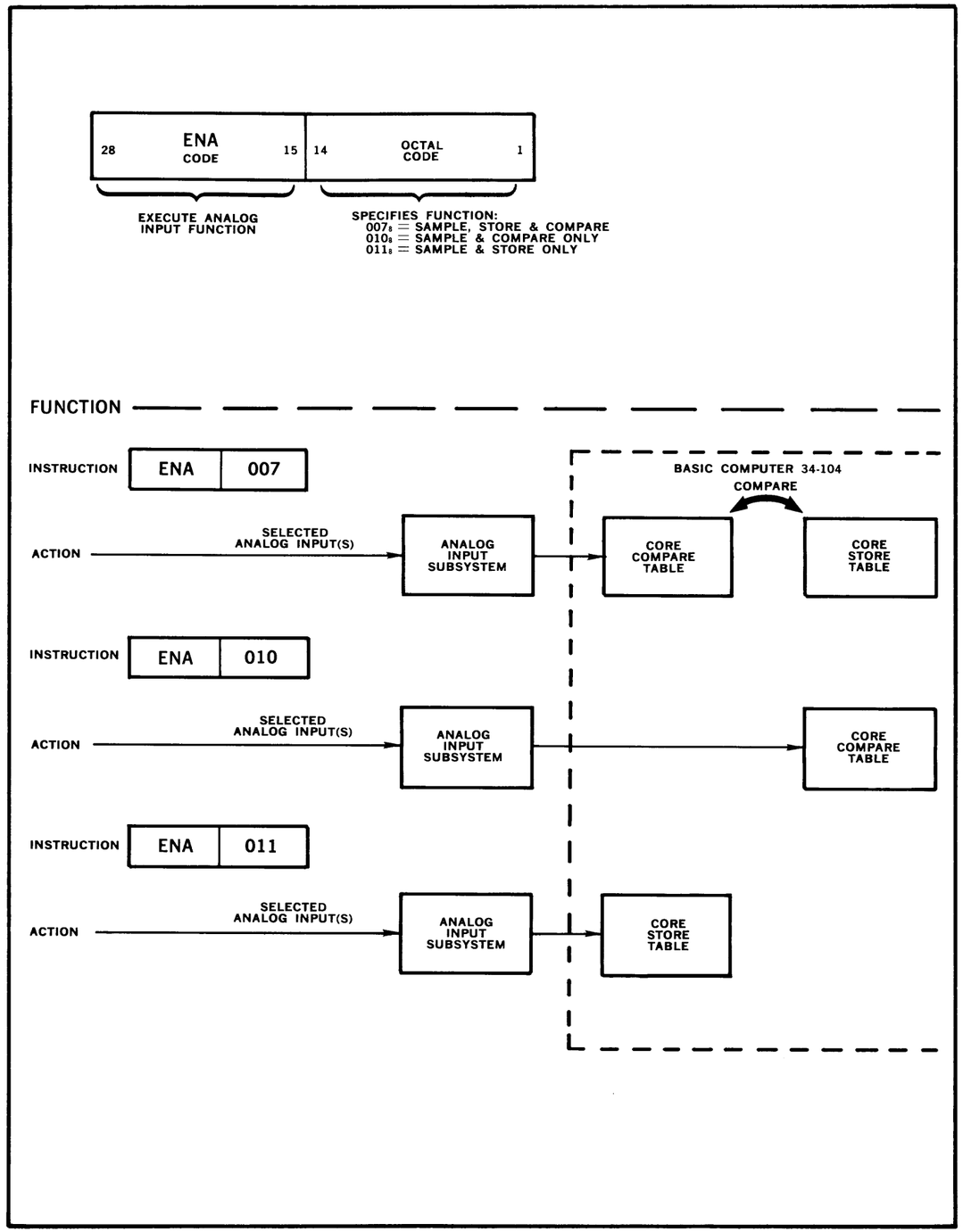


Figure 9-1. Analog Input Instruction Words and Simplified Functional Diagram

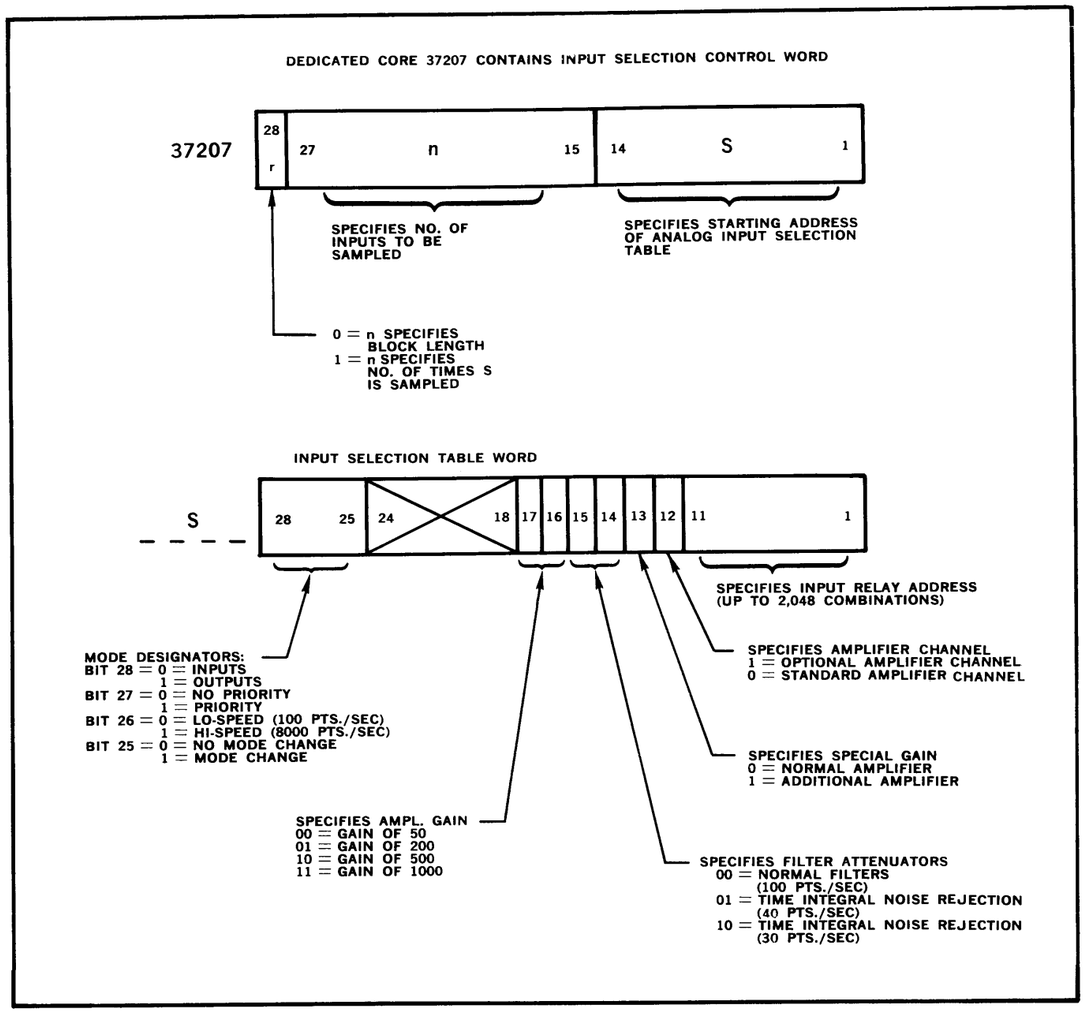


Figure 9-2. Analog Input Selection Control and Selection Table Words

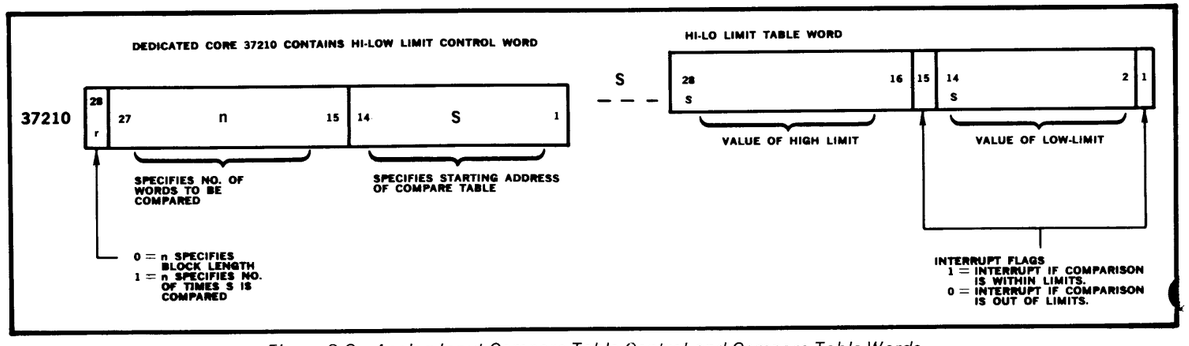


Figure 9-3. Analog Input Compare Table Control and Compare Table Words

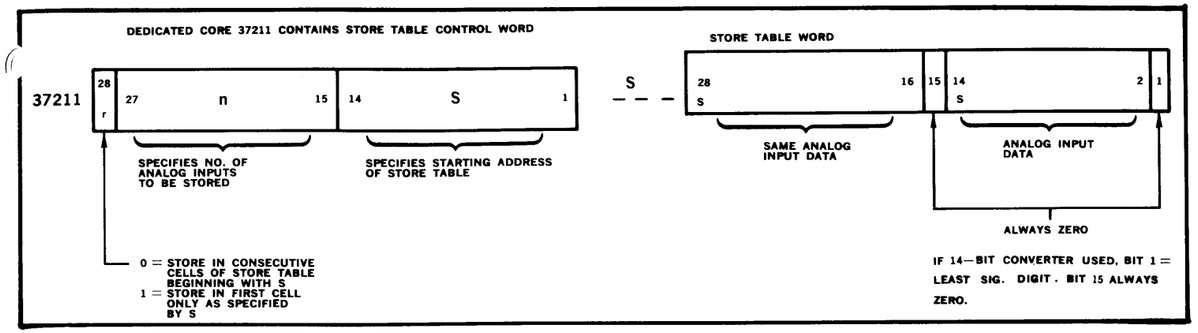


Figure 9-4. Analog Input Store Table Control and Store Table Words

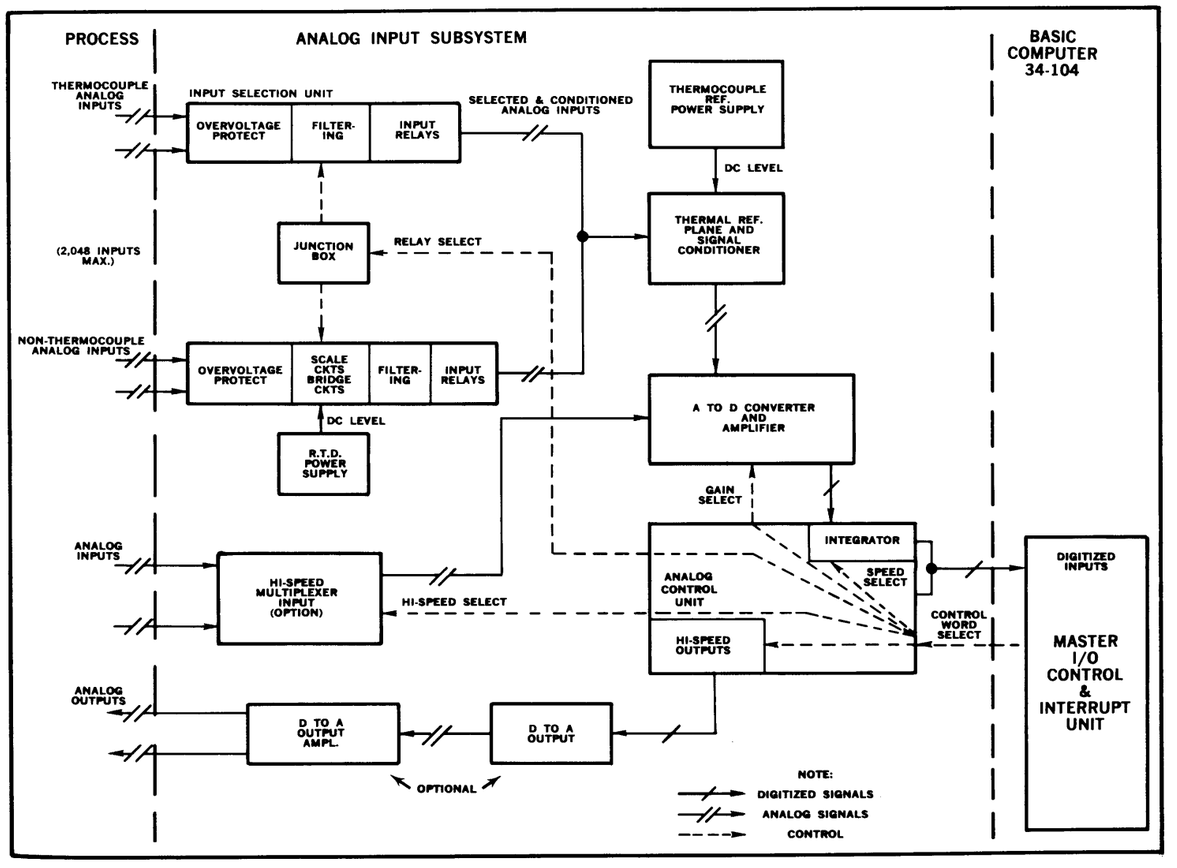


Figure 9-5. Analog Input Subsystem, Simplified Functional Diagram

SECTION X ANALOG OUTPUT SUBSYSTEM

GENERAL

The Analog Output Subsystem, Figure 10-1, converts computer-generated digital quantities into variable resistances, voltages, or currents for use in the control of process-connected analog controllers. All analog outputs provided by this subsystem are one of two types; resistance-divider outputs, or motor-driven potentiometer outputs. Motor-driven potentiometer outputs are provided in increments of seven up to a total of 128. Resistance-divider outputs are provided in increments of four up to a total of 128. The total number of analog outputs provided by the 340 system is 256.

The process controllers are under operator control and the Analog Output Subsystem is disabled and remains disabled until the operator presses the CONTROL OUTPUTS button located on the Operator's Control Console. In response to this action, the computer connects analog outputs to process controllers and assumes control of the process.

During computer control of analog outputs, the Fail-Safe Subsystem (para.below) continuously monitors primary power. Upon detection of a power failure, the Fail-Safe Subsystem maintains all analog outputs at their last settings. In addition, the operator may press the FREEZE OUTPUTS button, and in so doing, all computer-controlled analog outputs are held constant. An important part of the overall fail-safe approach in the design of the system is the inclusion of independent power sources for both the computer and the Analog Output Subsystem. Even though the computer may be disabled by a power failure, the analog output subsystem sustains all outputs at their last reasonable settings until the operator resumes manual control.

SUBSYSTEM HARDWARE

Figure 3-1 (1 of 3 and 2 of 3) is included to aid in defining subsystem hardware configurations. Frequent reference to these two diagrams will be necessary in the brief description of the hardware below.

In analog output systems employing Set Point Stations, a Multiple Contact Output Control Unit 34-220 is also required. This unit drives the Set Point Stations. A feedback line should be connected from each Set Point Station output to the analog input subsystem.

In analog output subsystems employing resistance-divider type outputs, a Multiple Contact Output Control Unit 34-220 is required. Resistance-Divider Unit 34-230 contains the divider select relays which are driven by the Multiple Contact Output Control Unit 34-220. Output terminal boards 34-232 provide interface connections for 16 resistance-divider analog outputs.

An optional Resistance-Divider Power Supply 34-231 is also available.

SIMPLIFIED OPERATION OF SETPOINT STATION OUTPUTS

The Computer Control Set Station is an instrument that indicates the setting of a given computer output signal and translates the signal as necessary for the output device the signal governs. The output signal from the computer drives either a pneumatic or an electronic transmitter in the Set Station. Outputs from the Set Station may be used as set points for secondary controllers or as direct signals to position the final control elements.

An indicating pointer on a dial on the face of the Set Station indicates the computer's control signal setting. This dial also shows the settings of hi-lo limit stops that limit the range of the output signal. These limit stops can be set from the front of the instrument.

To change the set point of a process controller or the position of a final control element, the computer signals the Computer Control Set Station to increase or decrease its output. A motor in the Set Station drives the pointer and the transmitter. White UPSCALE and DOWNSCALE lamps indicate that the computer is increasing or decreasing the output signal. Simultaneously, a feedback signal to the computer indicates the exact output position of the Set Station; the feedback signal is checked by the computer before a new set point is demanded.

A yellow COMPUTER lamp indicates when the Set Station is electrically connected to the computer. This lamp lights when a COMPUTER ON-OFF switch is in the ON position and a LIMIT BYPASS lever is in the LIMITS position.

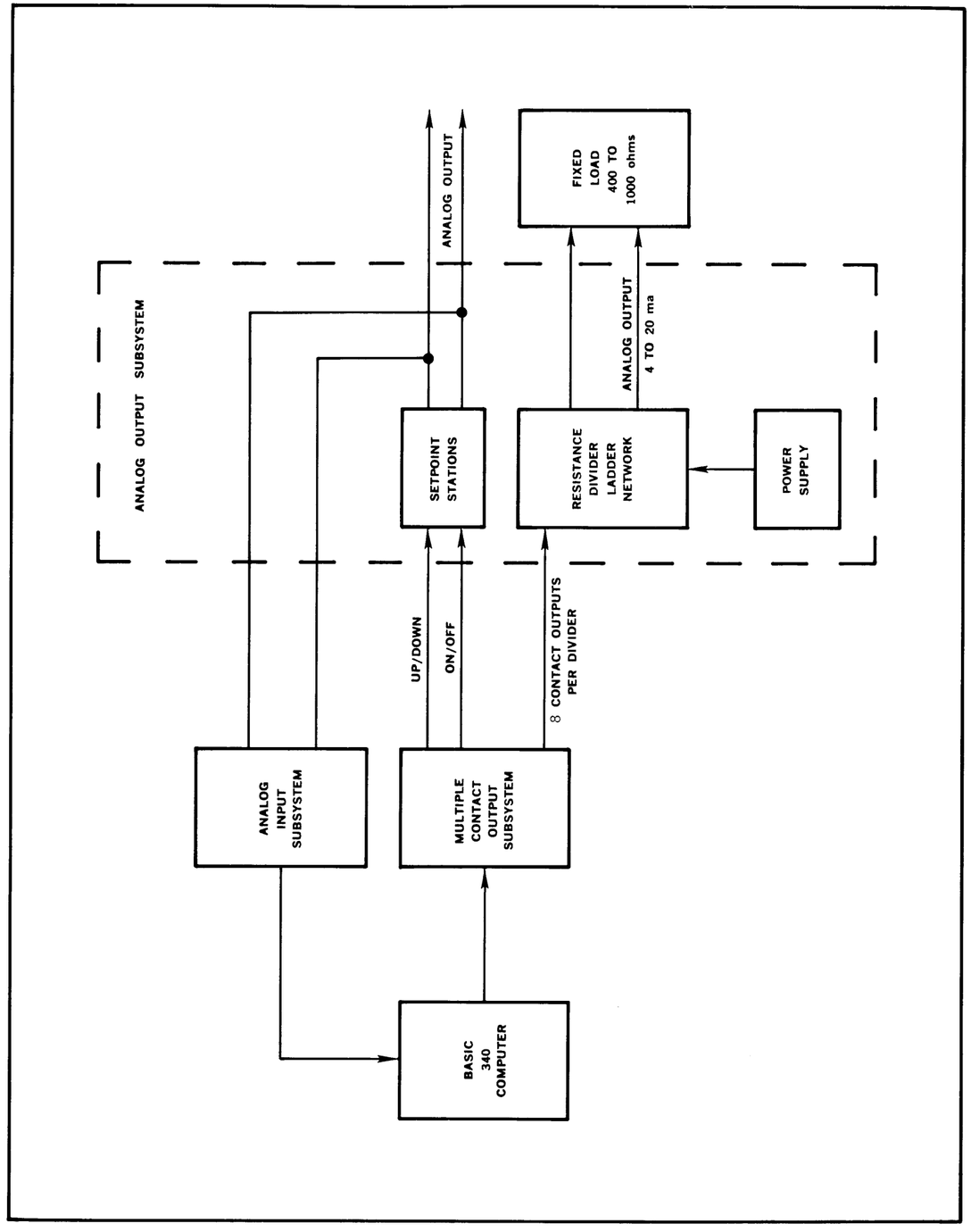


Figure 10-1. Analog Output Subsystem, Simplified Diagram

When the COMPUTER ON-OFF switch is set to the OFF position, the Set Station output may be adjusted manually, using an Operator's Control Knob, within the limit stops. With the LIMIT BYPASS lever in the LIMITS BYPASSED position, the output may be adjusted manually over the entire output range without disturbing the limit settings. This mode is useful during startup. The Set Station is connected to the controller in such a way that a "bumpless" transfer between the computer and manual operation is achieved.

A red AT LIMIT lamp indicates whenever a limit is reached or exceeded.

SIMPLIFIED OPERATION OF RESISTANCE-DIVIDER OUTPUTS

A single resistance-divider network includes eight relays. These relays are magnetically latching type devices which remain set until reset. The use of eight relays, each using two sets of contacts, allows every divider output to be stepped through the resistance range in 256 discrete settings between 4 and 20 milliamps into a fixed load of from 400 to 1000 ohms.

Each resistance-divider network is assigned a specific group address. When it becomes necessary to change an output from any one of the 128 resistance-divider networks, the computer, under program control, executes a contact output from data stored in a core table or core location. Only eight bit positions of the core stored data affect the divider output since only eight relays are involved. A one bit sets its associated relay and a zero resets the relay. Upon execution of the contact output, relays are set or reset and the output is altered through normal divider network action.

Resistance-divider analog outputs provide stepped output signals. This type of analog output allows instantaneous switching from any one setting to any other setting.

To implement resistance-divider analog outputs, the overall system requirements must include a Contact Output Subsystem.

Excitation voltages for resistance-divider outputs are not normally furnished by the manufacturer.

SECTION XI CONTACT INPUT SUBSYSTEM

GENERAL

The 340 Contact Input Subsystem provides communications between the operator and computer, and the process and computer. Communications between these points are handled by means of contact (switch or relay) closures which are representative of the state of various functions or devices. States may be interpreted as on, off, true, false, yes, or no; and through this method of communications, the computer is periodically contacted and data is updated or changed.

The contact input operations incorporated in the 340 repertoire of instructions make communications and information transfers simple and straightforward. Typical input devices include operator-controlled console switches, process relays, and process switches.

All contact inputs enter the 340 in groups of 28 (or less) input lines. Each group is addressed by an appropriate input enable code and the binary states of input lines are transferred to the computer. This information is then handled in one of the following ways: 1) stored in core memory; 2) compared against prestored data; or 3) both stored and compared.

Refer to Figure 11-1 and note the connections to process-connected contact inputs. These input lines are accepted by the computer as either open or negative voltage inputs. Negative inputs are recognized as true inputs and open lines are recognized as false inputs. A true signal is represented by -125 volts and a false signal is represented by 0 volt.

For purposes of explanation, each line of any input group is assigned a separate bit position in the computer word. Figure 11-1 illustrates a seven-line input group and indicates how bit positions in the computer word are affected by the input. (True inputs set the respective computer word bit.)

The basic Contact Input Subsystem is designed to provide input handling capabilities for up to 1,792 input lines. These lines are grouped into 64 octally numbered groups of 28 input lines per

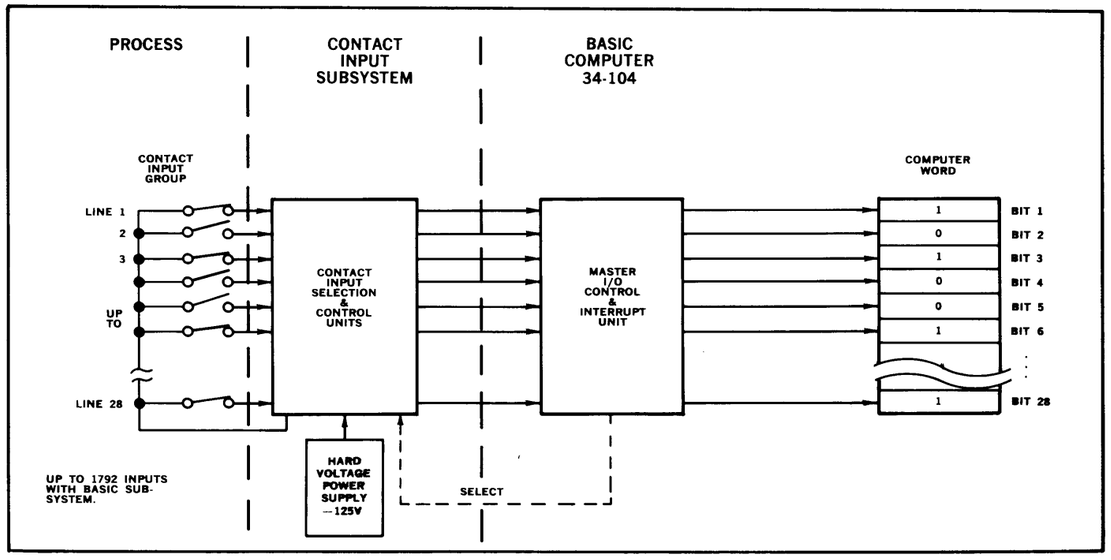


Figure 11-1. Basic Contact Input Subsystem and Group Input Handling

group. Input lines are sampled at the rate of 3 microseconds each input. Contact input group assignments are specified on an individual system basis and input group selection is discussed below.

SUBSYSTEM HARDWARE

Figure 3-1 (2 of 3) may be referenced in defining subsystem hardware configurations. The Contact Input Subsystem consists basically of one or more Terminal Board Units 34-200, a Contact Input Selection Unit 34-202, and a Contact Input Control Unit 34-201. The Control Unit 34-201 communicates with the Master I/O Control and Interrupt Unit within the Basic Computer 34-104. This unit controls selection of up to 64 input groups as specified by the Master I/O Unit and exercises this control over up to eight 34-202 selection units. The Input Terminal Board 34-200 provides an input interface for up to 28 contact inputs. A Contact Input Hard Voltage Unit 34-204 is optionally available to provide a hard voltage input capability for up to 8 input groups.

SIMPLIFIED SUBSYSTEM OPERATION

The following paragraphs provide a simplified operational description of the Contact Input Subsystem. As previously mentioned, the Contact Input Subsystem provides three separate functional features. These are:

1. Sampling a specified input group and storing these data in core memory.

2. Sampling a specified input group and comparing the sampled inputs against prestored data.

3. Sampling a specified input group, storing the sampled inputs, and comparing the stored sample against prestored data.

To accomplish the above functions, two major operations may be programmed depending upon the function desired. One method of inputting a group of input lines is known as a direct contact input. This instruction specifies the operation code (CIN) in bits 18-28 of the instruction, and the octal group address in bits 1-7. Bits 8-17 are not applicable. This instruction causes the selected group of 28 lines to be read into the A register. Although this instruction is useful in contact input programming, a second (store A) instruction is required to store the data in core. The second method, known as the contact input group enable, is more widely used and provides the flexibility of sampling, storing, comparing, or all three. The enable (ENA) code is represented in bits 18-28 of the instruction as shown in Figure 11-2. Bits 1-7 of the instruction word contain the function code which specifies one of the three functions listed above. Refer to Figure 11-2 and note that an octal code of 000 in bits 1-7 indicates that the contact input group is to be sampled, stored, and compared. A code of 001 in bits 1-7 specifies that the contact input group is to be sampled and compared only. Finally, an octal code of 002 specifies that the contact input group is to be sampled and stored only.

After the computer detects an enable instruction for either of the three contact input functions, the computer examines the contents of two or three core locations for further information. These core cells are known as dedicated core cells and are assigned a specific core address. The dedicated core locations for each contact input enable function are listed below:

ENA 000 --- dedicated core 37200, 37201, and 37202.

ENA 001 --- dedicated core 37200 and 37201.

ENA 002 --- dedicated core 37200 and 37202.

For example, if the ENA instruction specifies a 000 function, the computer examines all three cells since an 000 function specifies a sample, store, and compare operation. If the ENA function is 001, the computer examines only 37200 and 37201 since an 001 function specifies a sample and compare only operation. The 002 function involves cells 37200 and 37202 since this function specifies a sample and store-only operation.

All parameters relating to group input addresses, store table address, number of groups to be sampled, compare table addresses, etc., are defined in Figures 11-3, 11-4, and 11-5. These illustrations provide the information relative to the type of data contained in each of the three dedicated core cells as well as the associated data tables. Dedicated core locations and data tables must be set up prior to execution of the ENA operation.

In summarizing the Contact Input Subsystem operation, it should be noted that control of input group selection is handled through communications between the Contact Input Control Unit 34-201 and the Master I/O Control and Interrupt Unit within Basic Computer 34-104. The Master I/O is in turn activated by the ENA instruction executed in the Central Processor circuits.

When the Contact Input Subsystem is operating in the comparison mode and a non-compare occurs, the subsystem performs the following:

1. The results of the comparison are stored in location 37470 and the group address is stored in location 37471.

2. A non-compare alarm interrupt is generated (type 2).

3. The Contact Input Subsystem is disabled.

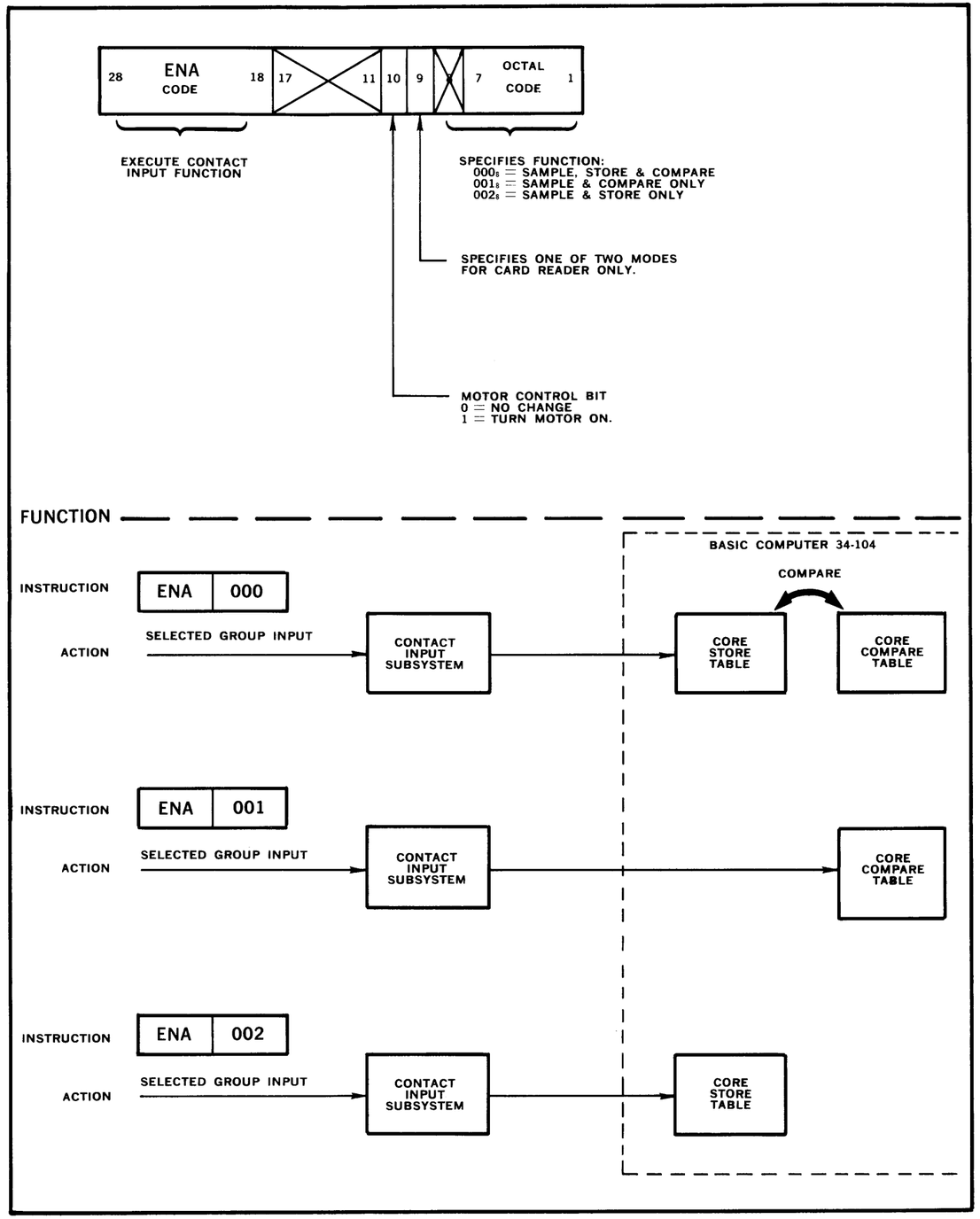


Figure 11-2. Contact Input Instruction Word and Simplified Functional Diagram

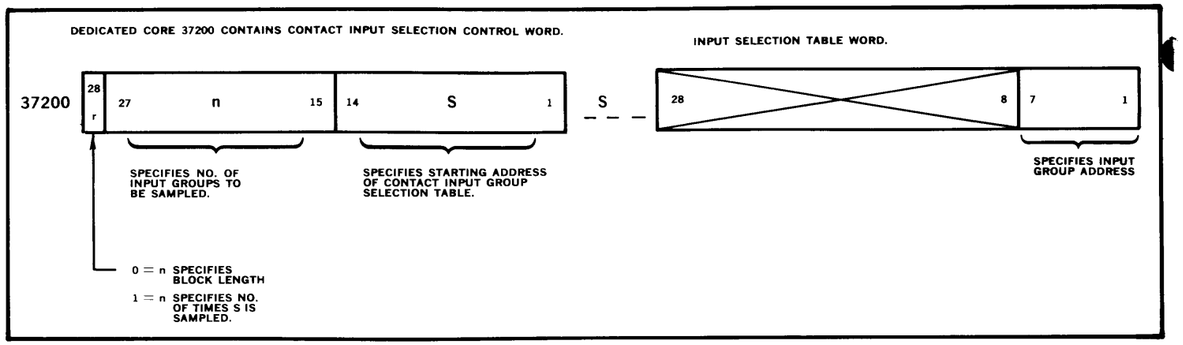


Figure 11-3. Contact Input Selection Control and Selection Table Words

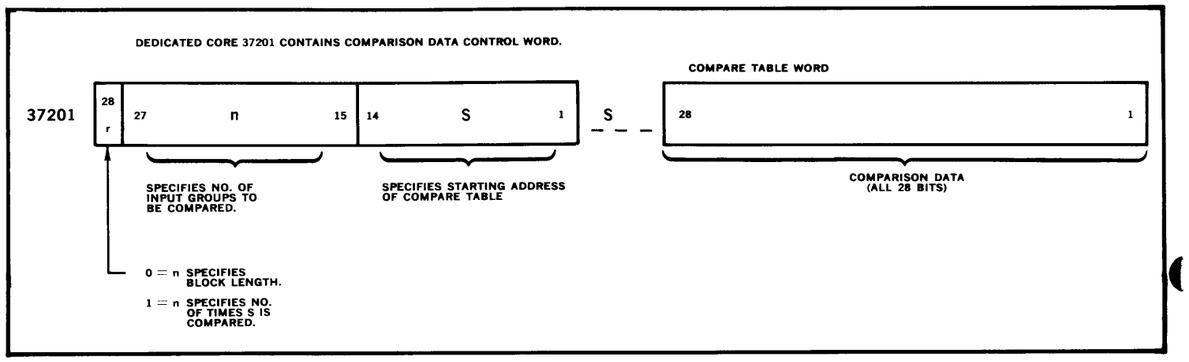


Figure 11-4. Contact Input Compare Table Control and Compare Table Words

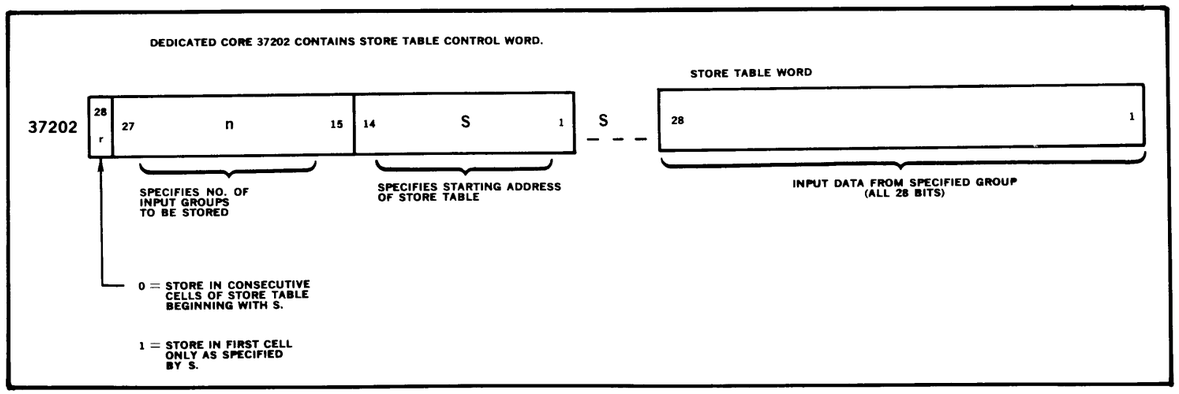


Figure 11-5. Contact Input Store Table Control and Store Table Words

SECTION XII CONTACT OUTPUT SUBSYSTEM

GENERAL

The Contact Output Subsystem allows the computer to communicate with process and operators. While process requirements vary considerably, the 340 Contact Output System makes special systems adaptability possible.

The 340 repertoire of instructions incorporates contact output operations that make communications and information transfers simple and straightforward. Typical output devices include console visual displays, alarm annunciators, and relays or switches for process use.

The Contact Output Subsystem communicates with output devices or process contacts by means of contact closure configurations. That is, the output from the computer is a digital quantity represented by the state of an output line or a group of output lines.

All contact outputs originate within the computer and are outputted to the process functional element in groups of 28 output lines. Each output group is assigned a specific code and the binary configuration of output lines are represented by individual bit positions of the contents of memory. Refer to Figure 12-1.

The basic Contact Output Subsystem is designed to provide 1792 multiple contact output lines. Contact output group assignments are specified on an individual system basis and output group selection is discussed below.

SUBSYSTEM HARDWARE

Figure 3-1 (2 of 3) is included as an aid in defining subsystem hardware requirements and may be referenced throughout the following.

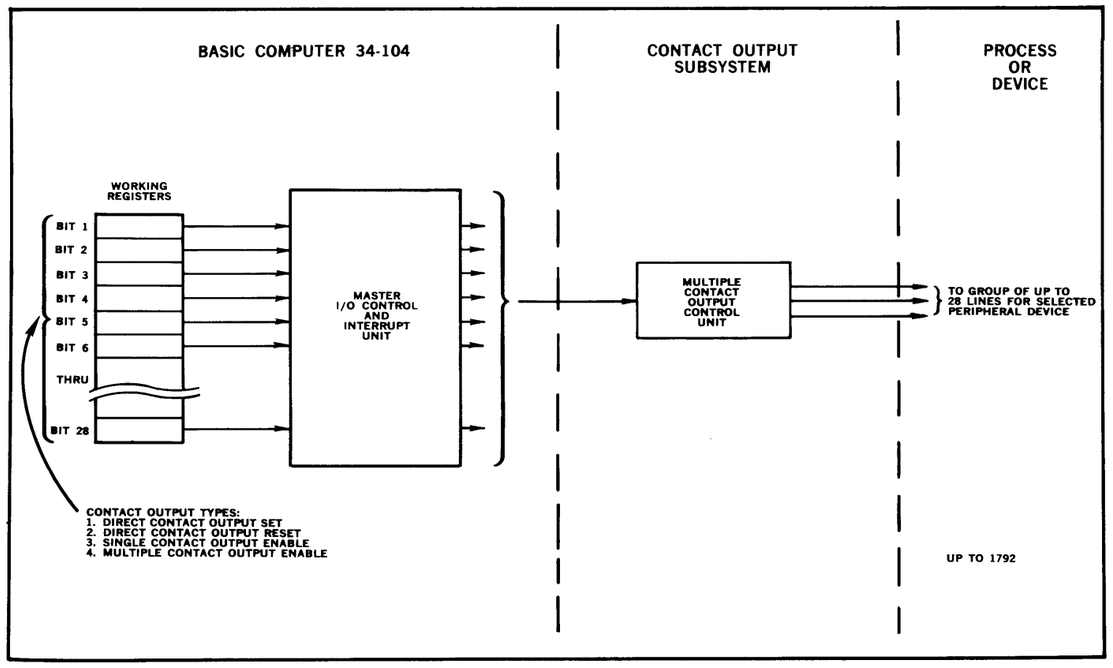


Figure 12-1. Basic Contact Output Subsystem and Group Output Handling

Required multiple contact output hardware includes a Multiple Contact Output Control Unit 34-220, Multiple Contact Output Unit 34-221, and necessary Terminal Board Units 34-223. The 34-220 unit controls the selection of relays within output unit 34-221 to provide outputs to 64 groups of 28 lines per group. To enable 1792 outputs from one control unit, 32 output units (34-221) are required. A supplementary control unit may be optionally acquired to double the control capacity in which case the number of output units is also doubled. A single terminal board assembly provides output interface for one group of 28 multiple contact outputs. Relays within the output units, whether 110 VA or 250 VA type, may be mercury-wetted contact self-latching or momentary-type relay output. In systems configurations requiring 1) contact output override capabilities, 2) contact output resistance-divider analog output capabilities, or 3) peripheral plotting equipment; the total number of available contact outputs are reduced. For further information relative to these types of outputs, contact the Bunker-Ramo regional field office in your area.

SIMPLIFIED SUBSYSTEM OPERATION

The multiple contact output subsystem sets one or all 28 process connected relays within any one of 128 relay groups.

Multiple contact outputs affect all relays within a group. Outputs of this type are used for driving set point stations, resistance-divider analog outputs, and outputs to peripheral plotting equipment as well as for process outputs.

To accomplish multiple contact outputting, the Master I/O Controller communicates with the multiple contact output control units in the same manner as described for single contact outputs except that the enable operation used is slightly different. The instruction involved is, as in single contact outputs, the ENABLE instruction (ENA) and the code representing this instruction appears in bits 15-28 of the instruction word shown in Figure 12-4. Bits 1-14 contain the multiple contact output function code. This code is always an octal 004.

When the computer detects an ENA 004 instruction, the computer examines two dedicated cells in core memory in sequence. These cells are numbered 37204 and 37205. These locations contain the information relating to output group select and group data locations as shown in Figure 12-4. Dedicated core locations and data tables must be set up prior to execution of all contact output instructions.

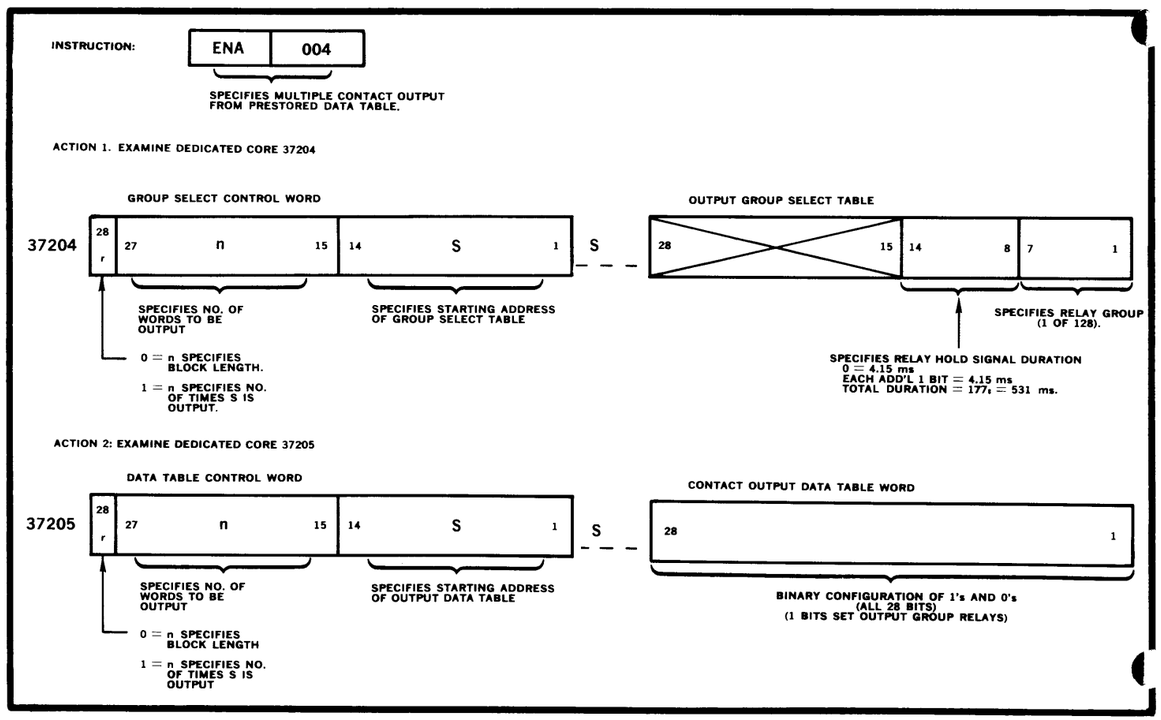


Figure 12-2. Multiple Contact Output Device Enable Control and Data Table Words

SECTION XIII INTERRUPT SUBSYSTEM

GENERAL

The 340 Interrupt Subsystem serves as a rapid means of altering computer operations in response to detected changes in critical process or computer operating conditions. Through the Interrupt Subsystem, the computer program in progress may be automatically interrupted (stopped) in order to: 1) respond to and correct process emergencies, 2) accept or output data and/or alarm messages, and 3) perform calculations and/or adjustments as necessary.

The basic 340 is designed to accept up to 28 interrupt signals. Five are reserved for interrupt functions originating within the computer proper.

The most important single feature of this subsystem is the ability of assigning up to 28 distinct levels of priority to incoming interrupt signals. This feature enables programmed control over initial priority assignment or a change in assignments as required.

SIMPLIFIED SUBSYSTEM OPERATION

The 340 Interrupt hardware continuously monitors all interrupt lines for detection of a change in state from false to true. Detection of a true signal, either from process connected sensors and limit switches or manually actuated operator switches, causes the computer to react to the associated interrupt. Refer to Figure 13-1 and note that the lo-level sensor shown produces an interrupt input when the tank liquid goes below a prescribed level. At the instant the lo-level sensor is closed the computer recognizes a change of state on the line. The computer finishes the instruction in progress and upon completion, the computer is forced to branch to an interrupt subroutine associated with the particular interrupt condition. In the illustration, the subroutine activates a pump which refills the tank, checks the level, and when the level is reached, the pump is turned off. The last instruction in the subroutine is a branch to an address which causes the computer to resume at the place in the program at which the interrupt occurred.

Under the priority system, also shown in Figure 13-1, if a higher priority situation occurs during a lower priority interrupt subroutine, the lower priority subroutine is abandoned until the higher priority has been satisfied. Upon completion of the higher priority subroutine, the computer is returned to the lower priority subroutine and finally to the original point in the working program at which the first interrupt occurred.

The 340 detects changes in lines from two types of interrupts. These interrupts are known as Type 1: those signals which are continuous, and Type 2: those signals which are momentary in nature but which must be at least 12 μseconds in duration. Since inputs of the momentary type could conceivably be lost (or forgotten) before the computer could react, a Q register is included in the Interrupt Subsystem to prevent such a loss. As an example, consider that interrupt 5 in Figure 13-1 is a momentary type, and that at the occurrence of interrupt #5 the computer happened to be involved in subroutine #4. Considerably more than 12 μsec could elapse before the finish of routine #4 and resumption of routine #5. The Q register is used to solve this problem by changing a momentary signal to a continuous signal, thereby providing a means of remembering occurrences of Type 2 interrupts.

Another register, the M register, is used as a mask register for all interrupts, Type 1 and Type 2. The M register serves as the means for establishing priority and may be loaded or changed under program control as necessary. Figure 13-2 illustrates how priority is assigned and how interrupt types are intermixed. Figure 13-2 also illustrates the various configurations of M and Q registers available as follows:

A continuous block of core (56 words) is reserved for the interrupt system. Of these, 28 words are used for recording the next program instruction (re-entry address) and the other 28 words are used for the first word of the interrupt response routine (the response address). These two groups of core words are interlaced such that each interrupt line has an associated "pair" of core addresses, starting with core address 37400. Core address 37400 holds the program next instruction address after the computer has been interrupted by interrupt line 1 and, core address 37401 contains the first instruction of the associated interrupt response routine for line 1.

In all interrupt configurations the five reserved interrupts are used as outlined in Table 13-1. Other interval interrupt signals are available for use at the discretion of the programmer. These are shown in Table 13-2.

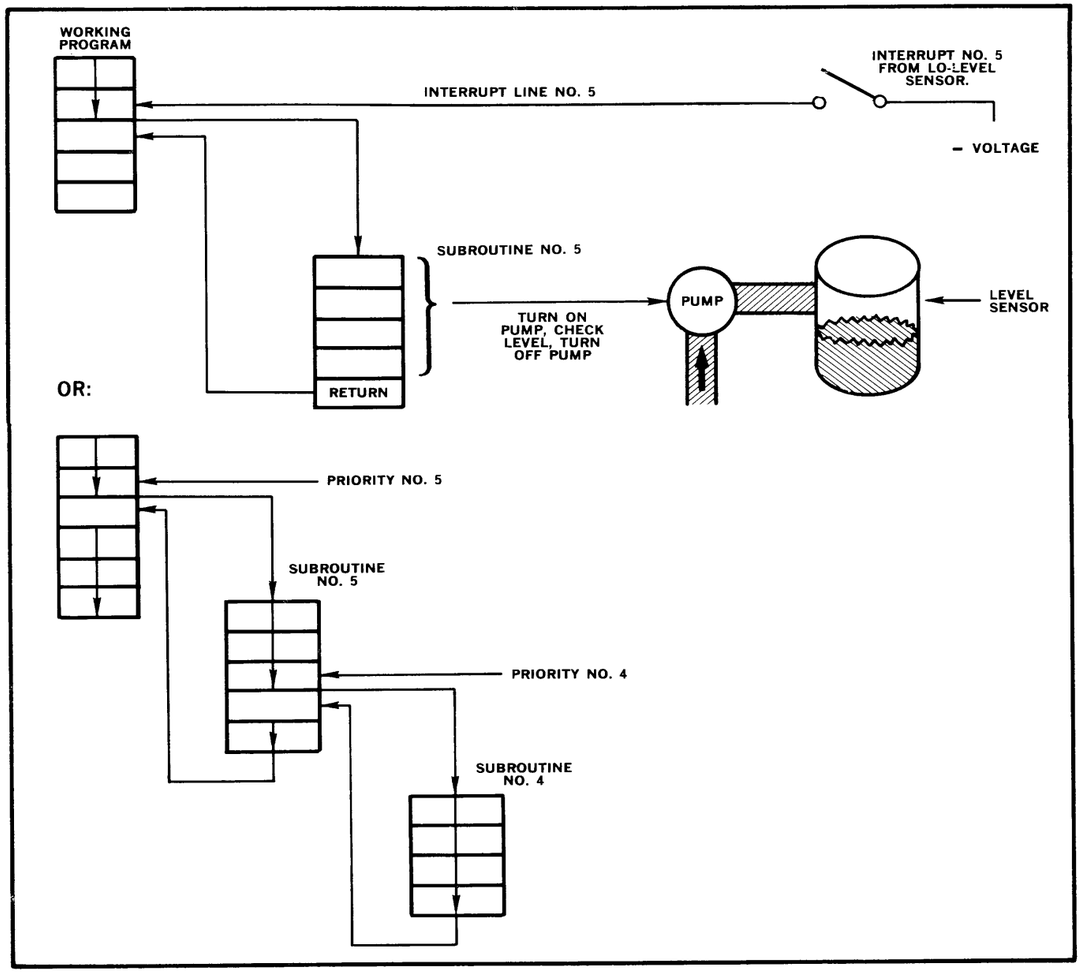


Figure 13-1. Simplified Interrupt Action

After completing an interrupt routine, the computer must return to the program that was interrupted. The next instruction address of the program that was being executed when the interrupt occurred was automatically merged with a BRANCH operation code, then stored into the first of the "pair" of addresses associated with the interrupt line. Thus, to exit from an interrupt routine, the response program goes to this re-entry address and program control branches back to the interrupted program.

Each interrupt line also has a unique response address assigned by the 340 Interrupt Subsystem. When an interrupt occurs, the computer automatically stores the next program instruction address in the assigned re-entry address, then transfers to the first address of the associated interrupt response routine which is stored in the second core location of the pair. Refer to Figure 13-3 for an example of re-entry and response address techniques associated with dedicated core locations. Table 13-3 lists all addresses associated with the interrupt subsystem.

The timing considerations for the Interrupt Subsystem may be divided into two parts. The time required to scan the interrupt lines and the time required to begin the response to interrupt once an interrupt is recognized. Scanning is done in 18 μsec for the 28-line interrupt system. The time to respond to the interrupt signal depends on how long it takes to complete the instruction in progress.

Table 13-1. Reserved Interrupts

Interrupt Source	Type	Function
Elapsed Time	2	Signals run-down of elapsed time counter
End Drum-Core Transfer	2	Signals end of information transfer between drum and core
I/O Transfer Complete	1	Computer/Device information transfer completed
Drum Parity Error	1	Signals odd-bit parity error in a drum word
Instruction Parity Error	1	Signals odd-bit parity error in a core word

Table 13-2. Available Internal Interrupts

Interrupt Source	Type	Function
Operand Parity Error	1	Core parity error detected when accessing operand
I/O Access Parity Error	1	Core parity error detected when transferring data to Master I/O Controller.
Analog Input Out-of-Limits	2	Generated when analog input exceeds limits. This interrupt is required by the standard analog scan programs.
Contact Input Non-Comparison	2	Generated when contact input does not compare. This interrupt is not required by the standard contact scan programs.
Operator's Console Request	1	
Operator's Console Cancel	1	
Operator's Console Unassigned	1	For optional use in system program.
Operator's Console Unassigned	1	
Operator's Console Connect Outputs	1	
Operator's Console Freeze Outputs	1	
Card Punch Alarm	1	Equipment malfunction.

Table 13-2. Available Internal Interrupts (Cont.)

Interrupt Source	Type	Function
Card Reader Alarm	1	Equipment malfunction.
Paper Tape Punch, Low Paper	1	Signals low paper level.
Paper Tape Reader, Parity Error	1	Signals parity error on tape.
Once-A-Second Pulse (64/60)	2	Not used in standard programming system.

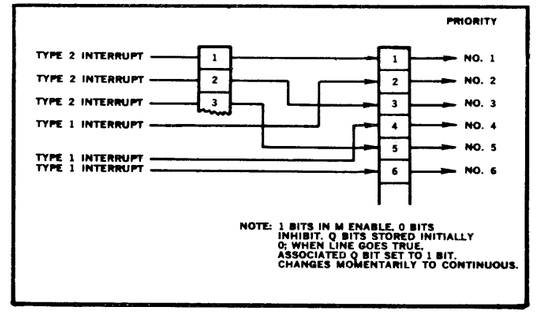


Figure 13-2. Interrupt Register Configurations and Simplified Priority Establishment

Interrupt Line No.	Re-Entry Address	Response Address
1	37400	37401
2	02	03
3	04	05
4	06	07
5	10	11
6	12	13
7	14	15
10	16	17
11	20	21
12	22	23
13	24	25
14	26	27
15	30	31
16	32	33
17	34	35
20	36	37
21	40	41
22	42	43
23	44	45
24	46	47
25	50	51
26	52	53
27	54	55
30	56	57
31	60	61
32	62	63
33	64	65
34	37466	37467
41	37500	37501

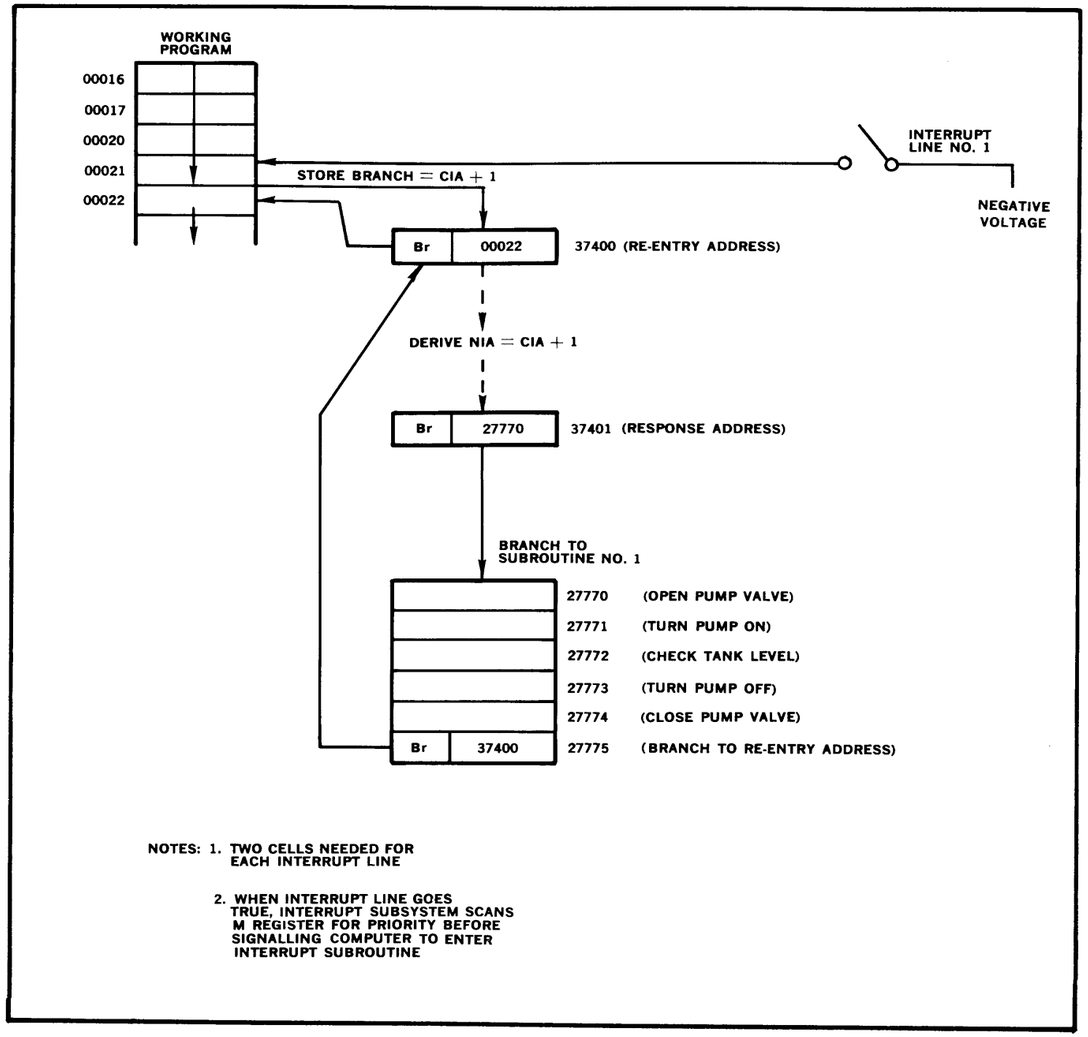


Figure 13-3. Interrupt Re-Entry Response Address Techniques

SECTION XIV
OPERATOR'S CONSOLE AND FAIL-SAFE OPERATION

GENERAL

An Operator's Console, provided with most 340 Control Computer Systems, serves as a central point from which the operator communicates with the system. Most consoles supplied are of the standard console design, as shown in the illustration; however, custom consoles are provided to meet the needs of specific applications. The standard console provides working desk space and an operator's control panel from which operational communications are maintained.

OPERATING CONTROLS AND INDICATORS

The operating controls and indicators located on the Operator's Control Panel are described below and consist of the following: (1) a six-character Data Display, (2) a 28-position Function Matrix, (3) two Function Selector switches, (4) 12 Manual Entry decimal input switches, and (5) seven Control pushbutton switches. Refer to Figure 14-1 for an illustration of these controls and indicators.

A. Six-Character Data Display

The standard display consists of six projection-type (visual) display indicators. Each indicator displays one of twelve characters. Indicators D_1 through D_4 display numerals 0 through 9, plus a green and a red dot. D_6 displays the sign of the number. Indicator D_5 displays the following designation words:

VOLTS
TIME
TEMPERATURE
PRESSURE
FLOW
PERCENT
INSTRUMENT
EXEC-ROUTINE
MEMORY VALUE
LOWER LIMIT
UPPER LIMIT
RED DOT

Indicator D_5 identifies the type of information being displayed on indicators D_1 through D_4. Note that in Figure 14-1, the DATA DISPLAY presents TIME (time of day).

The information presented on the DATA DISPLAY indicators results from a computer multiple contact output as requested through specific contact inputs from other panel selectors. For example, before a TIME contact output is displayed, a contact input request for that functional display must be made through the use of the FUNCTION SELECTOR and CONTROL switches. These switches are more thoroughly discussed below.

B. Function Matrix

The Function Matrix is a translucent material divided into 28 squares (4 x 7 matrix). Each square contains an engraved code number in the upper-left corner and a function designation. A lamp mounted behind each square lights when the operator selects a function. Typical functions include:

Trend Display
Trend Log
Trend Log Point
Set Point Lineup
Time Display
Demand Log
Alarm Summary
Memory Display
Memory Print
Enter Data

Figure 14-1 shows a Function Matrix containing the standard functions. Each Function Matrix is engraved to include special functions for specific systems.

The FUNCTION MATRIX, through the use of 28 lamps, visually parallels the FUNCTION SELECTOR switch settings. As the FUNCTION SELECTOR switches are positioned through their setting range, the matrix displays a readout which is equivalent to the matrix code selected. An example of the interrelationship between the FUNCTION SELECTOR switches and the FUNCTION MATRIX is illustrated in Figure 14-2. Refer to the diagram and note that the heavy line connecting switches S11 and S12 through lamp 22 provide a complete circuit for the lamp, thereby providing a direct readout of the selected function. In this example, the function corresponding to code 22 specifies a DEMAND LOG.

C. Function Selector Switches

Two thumbwheel input switches, labeled FUNCTION SELECTOR S11 and S12, allow the operator to set up the desired function select code. As illustrated above, switches S11 and S12 must be set to read the number designated in the upper-left corner of the appropriate FUNCTION MATRIX square. Once switches S11 and S12 have been set to a code and a request made, the computer executes a contact input from these switches to determine which coded function is desired. Refer to Figure 14-2 and note that with switches S11 and S12 set as illustrated, switch banks A and B supply grounded and ungrounded conditions on six BCD output lines. These lines, when interrogated by a computer contact input instruction from a specific address, produce the desired binary coded decimal line configuration. In Figure 14-2, switches S11 and S12 are shown positioned for a DEMAND LOG function, as represented by the 010 010 binary output corresponding to a function code of 22.

Of 64 possible codes which may be set on the FUNCTION SELECTOR switches, only 28 are normally available to the operator. The remaining pre-assigned coded functions are used by programmers and maintenance personnel.

D. Manual Entry Switches

Two sets of six thumbwheel input switches are connected to the computer as contact input lines. These switches are labeled MANUAL ENTRY switches S21-S26 and S31-S36 as shown in Figure 14-1. When the computer executes a contact input command from a specific address, these switches enter decimal numbers (in 4-bit BCD form) into the computer. Although the switches may be interpreted differently by different programs, standard programs generally use six of the switches (S31-S36) for addresses (or point numbers) and the remaining six (S21-S26) for data, scale, format selection, etc.

Switches S21 through S36 are wired similarly to S11 and S12. The primary difference is a 4-line BCD output from each switch instead of three lines as shown in Figure 14-2.

E. Control Switches

Seven momentary-contact pushbutton switches are provided on the standard operator panel. Working in conjunction with computer inputs, outputs, priority interrupts, and the fail-safe circuits, these back-lighted pushbuttons provide the operator with direct control over the computer program and hardware. Numbered from right to left, S1 through S7, the pushbuttons normally perform the functions described below.

1. REQUEST Switch (S1)

This pushbutton initiates the action requested by the operator through the FUNCTION SELECTOR switches, S11 and S12. Pressing the REQUEST pushbutton causes the button to glow. When the computer responds to the request, the light is extinguished (by the program).

2. CANCEL Switch (S2)

This pushbutton cancels any action previously requested. For example, to stop a "trend log" request, the operator first selects 13 on FUNCTION SELECTOR switches S11 and S12. The operator then presses the CANCEL pushbutton which produces an interrupt. The computer responds to the interrupt, stops the trend log printout, and turns off the light behind the CANCEL pushbutton.

In systems employing an Interrupt Subsystem, the CANCEL pushbutton is usually connected as an interrupt. In systems without interrupt capabilities, the CANCEL pushbutton is connected to the contact input/output subsystems.

3. Switches S3 and S4

These pushbutton functions are left unassigned and are available for customer assignment. The contacts of relays set by these pushbutton switches (or the contacts of the pushbutton) may be connected as contact inputs or outputs, or priority interrupt lines, or can be used as direct control over devices external to the computer control system. In most control computer installations, these pushbuttons are assigned to connect and disconnect computer outputs. Under these conditions, the pushbuttons are interlocked with fail-safe circuits to ensure that computer outputs are connected only after set-point lineup has been accomplished, and that computer operation is satisfactory.

4. ALARM Switch (S5)

When an alarm condition is detected, the computer executes a contact output command which causes the ALARM pushbutton to flash on and off. At the same time an audible alarm (chime or buzzer) is initiated. The operator then presses the ALARM pushbutton. When the ALARM button is pressed the alarm is silenced, the button glows steady red, and an "acknowledge" signal is sent to the computer. The ALARM button stays illuminated until the alarm condition is corrected and a computer response is received.

5. FREEZE OUTPUTS and CONNECT OUTPUTS Switches (S6 and S7)

These two pushbuttons are integrated with the fail-safe circuits. When the operator presses the FREEZE OUTPUTS pushbutton, the computer is prevented from changing any output currently being applied to the process. Also, fail-safe circuits cause an output freeze condition independent of operator or computer action. Only the operator may initiate output control (by pressing the CONNECT OUTPUTS pushbutton), and such action is effective only if the computer is operating satisfactorily, and, if a set-point line-up has been performed.

NOTE: The standard operator console functions are normally accommodated through 84 contact inputs, 28 multiple contact outputs, and up to seven interrupt lines.

FAIL-SAFE OPERATIONS

A power control chassis, located within the Operator's Console, contains a timer and associated power interlock circuits which function to automatically freeze contact outputs upon detection of power failure or computer malfunction. Refer to Figure 14-3 and note that a pre-set timer is kept energized by a reset signal applied under program control. The reset signal may be set to any duration from 0.1 to 10 seconds. As long as the timer is not allowed to run down, power to contact output control units is sustained.

If the program fails, however, the timer runs down and all contact outputs from the computer are disabled since the -14 volt dc line to the contact output control unit is broken. Note also that the console operated FREEZE OUTPUTS pushbutton described above has the same effect on contact outputs as does the timer. The CONNECT OUTPUTS switch places all contact outputs under the control of the computer.

NOTE: The reset signal shown in Figure 14-3 may be of variable duration to satisfy specific needs.

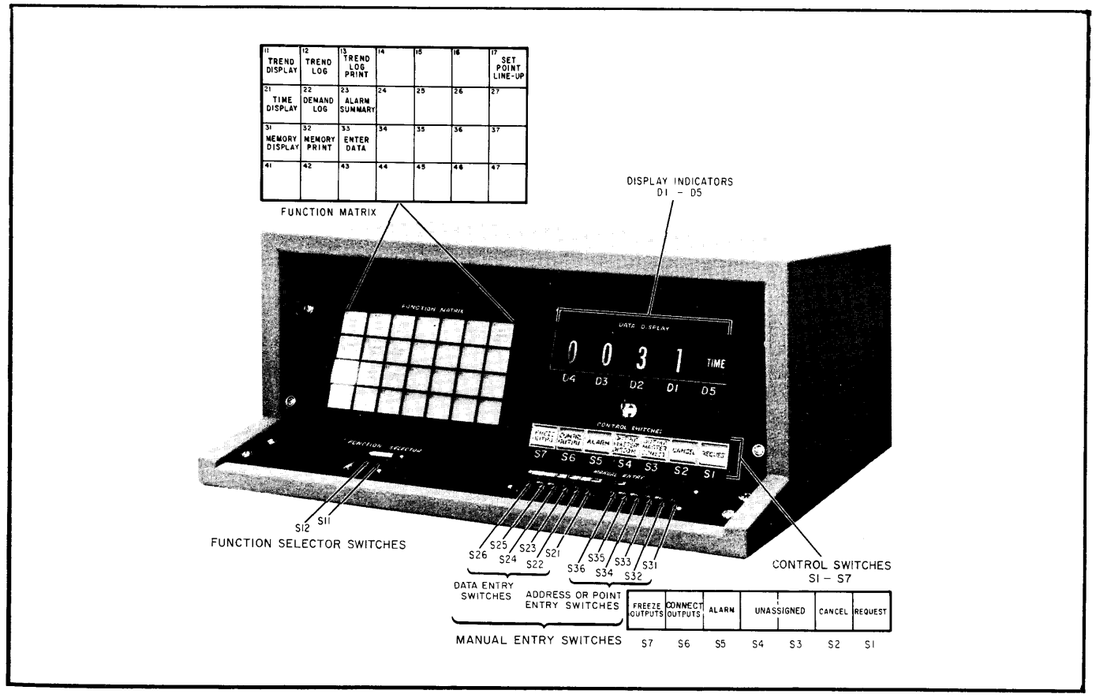


Figure 14-1. Typical TRW-340 Operator's Control Panel

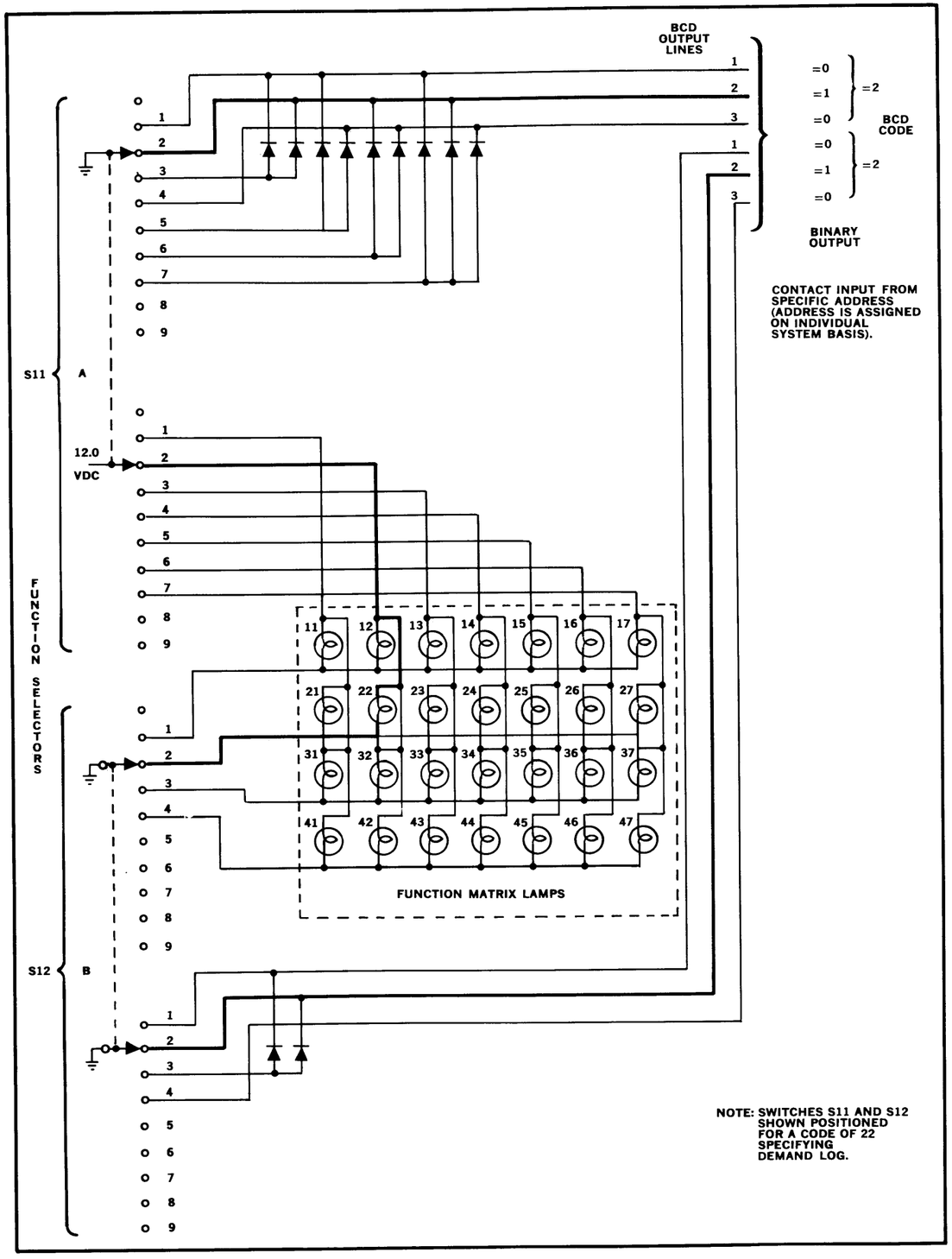


Figure 14-2. Simplified Schematic Diagram of Function Selector-Matrix Operation

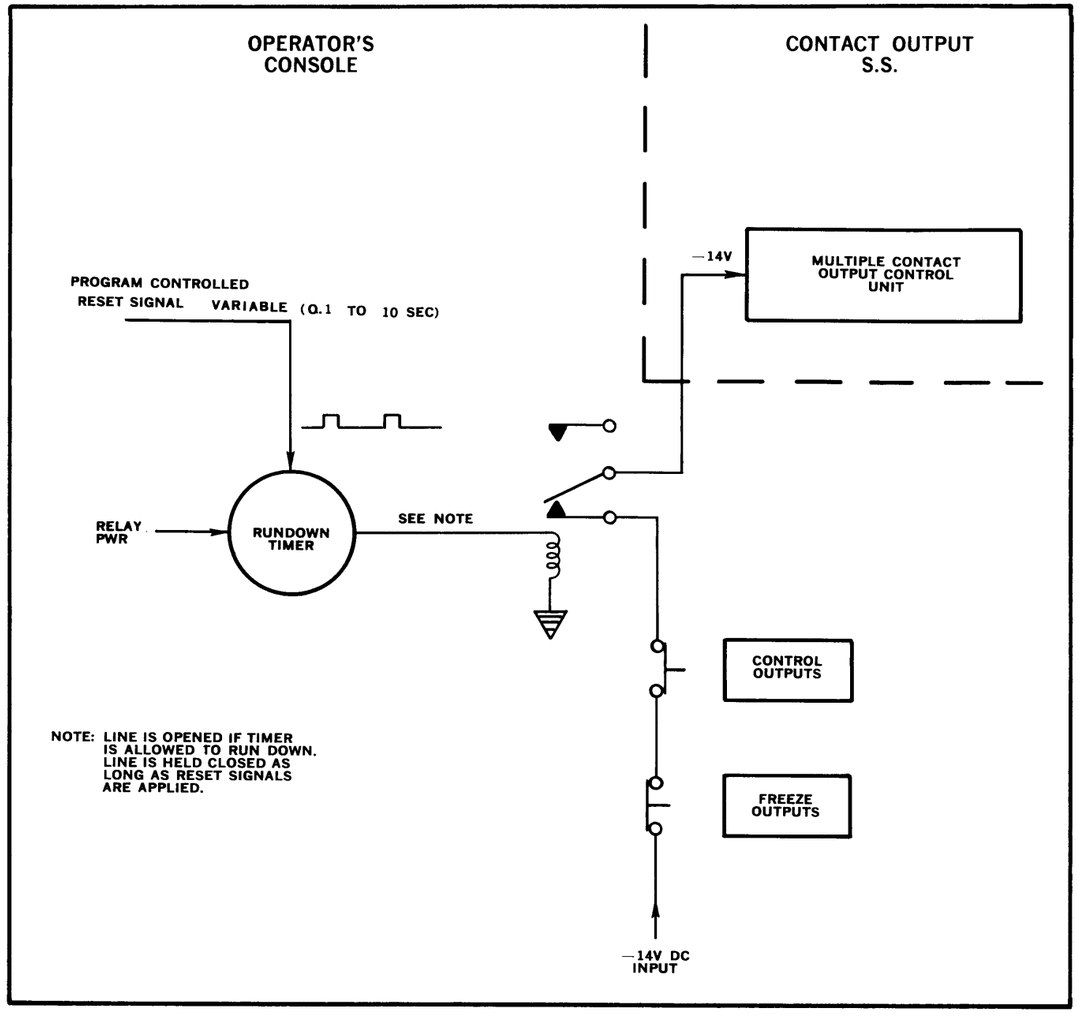


Figure 14-3. Simplified Fail-Safe and Reset Timer Operation

SECTION XV INPUT/OUTPUT PERIPHERAL DEVICES

GENERAL

Presently, eleven input/output peripheral devices are considered standard 340 system options. As system applications grow, it is anticipated that additional peripheral devices will be added to this list. The devices to be covered in this section include the following:

A. Output Writer 34-416 or -424 (16 or 30 inch carriage)

B. Teletype Printer 34-408

C. Card Reader 34-402

D. Card Punch 34-404

E. Paper Tape Reader 34-410

F. Paper Tape Punch 34-412

G. Input Keyboard 34-417

H. High-Speed Line Printer 34-425

I. Teletype Input 34-406

J. Magnetic Tape Unit 34-431

K. Plotter Models 34-426, -427, -428, or -429

HARDWARE REQUIREMENTS

Refer to Figure 3-1 (3 of 3) to aid in defining hardware requirements for each of the peripheral devices listed above.

Two Output Writers; 34-416 or 34-424, or a combination of the two, are controlled by one Output Writer and Keyboard Control Unit 34-415. Each Output Writer however, must be accompanied by a separate Decoder Adapter Unit 34-418. The Output Writer and Keyboard Control Unit 34-415, in addition to controlling two Output Writers, is also capable of accommodating inputs from one Input Keyboard 34-417. Individual control units must be provided with the following:

Device	Required Control Unit
Card Reader 34-402	34-401
Card Punch 34-404	34-403
Teletype Printer 34-408	34-407 and/or 34-405
Paper Tape Reader 34-410	34-409
Paper Tape Punch 34-412	34-411
Teletype Input 34-406	34-405
High Speed Line Printer 34-425	34-419

Four Plotters are also available with the system. Model numbers for these units are: 34-426, 34-427, 34-428, and 34-429 respectively. In systems employing either of these plotters, a Multiple Contact Output Control Unit 34-220 and a Multiple Contact Output Unit 34-221 is required. This unit serves as the plotter controller.

OPERATING FEATURES

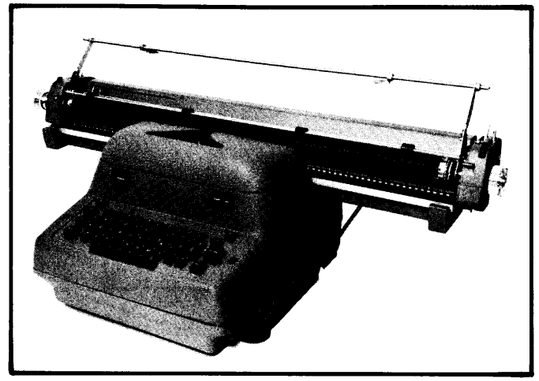
The following paragraphs briefly describe the operating features of 340 peripheral devices.

A. Output Writer 34-416 and 34-424

The 34-416 is an electric typewriter equipped with a 16-inch carriage while 34-424 is the same unit equipped with a 30-inch carriage. These units are modified to operate with the 340. Both devices operate at a maximum rate of 10 characters per second from memory stored data packed four characters to the word. (This device will also operate from unpacked data.) Each character is six bits long. The typeface provided with the output writers is Micro Gothic style, single case, six lines and 14 characters per inch.

B. Teletype Printer 34-408

The 34-408 may be a Teletype Model 28 or a Model 32. The Model 28 is an output device while Model 32 serves as an input/output device. These units operate at a maximum rate of 10 characters per second from memory stored data packed four characters to the word. (This device will also operate from unpacked data.) Type size permits typed outputs at six lines and 10 characters per inch.

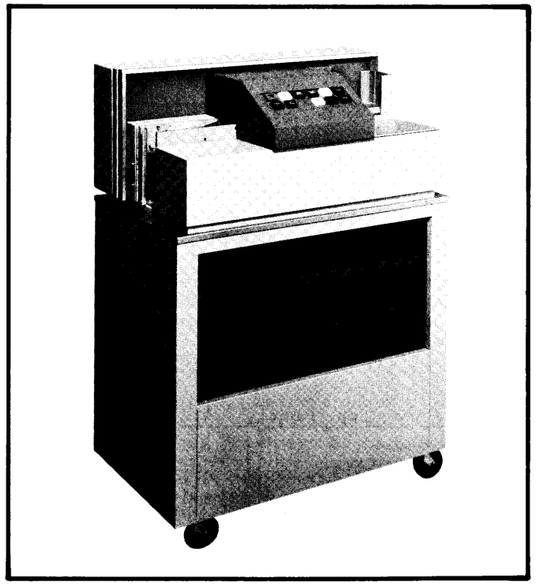


Output Writer

Teletype Printer

C. Card Reader 34-402

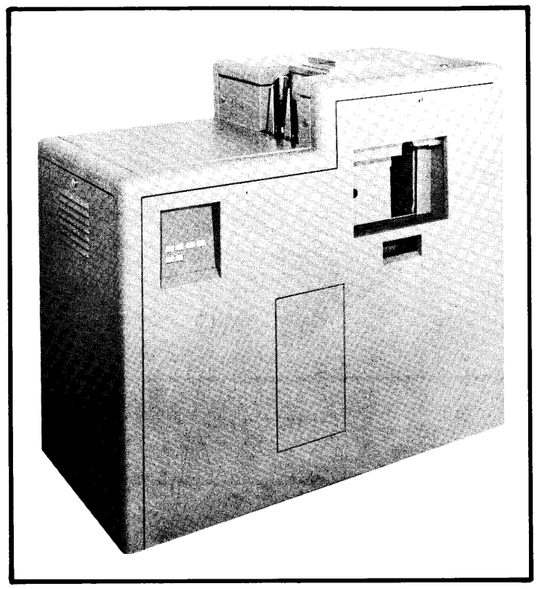
The card reader is capable of interpreting 80-column punched paper cards at a maximum rate of 200 cards per minute. This device will interpret either binary or alpha information.



Card Reader

D. Card Punch 34-404

The card punch is capable of punching 80-column, 12-row punched paper cards at a maximum rate of 100 cards per minute.



Card Punch

E. Paper Tape Reader 34-410

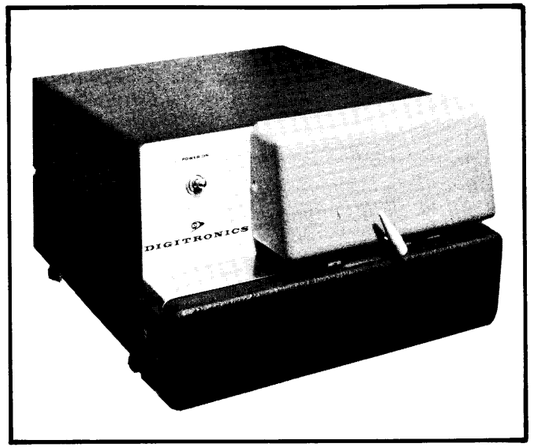
This device is capable of reading data from seven-level paper tape into memory at a maximum rate of 300 characters per second. The seventh level on the tape is for odd parity checking of each character read.

F. Paper Tape Punch 34-412

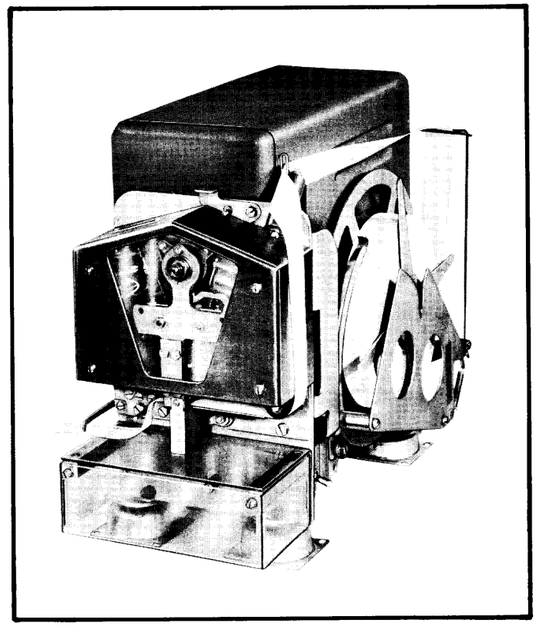
The 34-412 is capable of punching seven level paper tape at the rate of 110 characters per second. The seventh level on the tape is an odd parity check position.

G. Input Keyboard 34-417

The 34-417 enables the entering of data into memory via the 34-415 Output Writer and Keyboard Control Unit. Data is entered in blocks of six characters each and input data is not packed from this device.



Paper Tape Reader

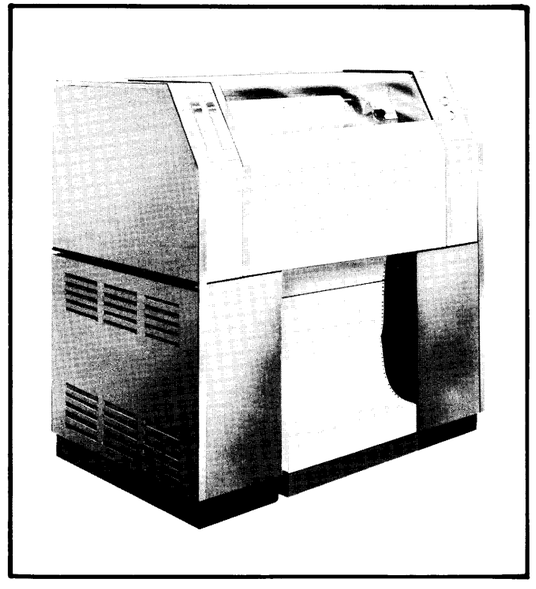


Paper Tape Punch

Input Keyboard

H. High Speed Line Printer 34-425

The 34-425 is a high-speed line printer capable of printing output messages at the rate of 300 lines per minute. Each line is 120 columns long and each column accommodates one of 64 available characters. Memory stored outputs are packed four characters to the word and each character is six bits long. Vertical spacing is six lines per inch.



High Speed Line Printer

I. Magnetic Tape Unit 34-431

The 34-431 Magnetic Tape Unit stores data on standard 1/2-inch wide magnetic tape in the form of 6-bit characters plus parity. Data is transferred to or from tape at rates of either 41.7 kc or 15 kc depending upon whether the unit is operated in the high or low density mode. In the high density mode, characters are packed 556 per inch. In the low density mode, characters are packed 200 per inch. In either mode, the tape is traversed at the rate of 75 inches per second. Data is always stored in blocks of at least 10 characters and an end-of-block even parity check is made of the longitudinal count. As each character is read from or written on tape, character parity is checked. Character parity may be either odd or even as determined by the control word.

In the high density mode, the computer makes characters available for writing every 24 μsec. Characters are read at the same rate. In the low density mode, characters are written every 66 μsec.

Up to eight Magnetic Tape Units may be driven by a single 34-430 Magnetic Tape Control Unit. Outputs to, or inputs from the tape unit are controlled by the 34-430 unit as directed by the Master I/O Control and Interrupt Unit. The tape control unit is capable of controlling the reading or writing of only one tape unit at a time. The only exception is made during rewinding of one unit at which time a second unit can be operated.

J. Plotters 34-426, 34-427, 34-428, 34-429

Four separate digital incremental plotters are available for use with 340 Control Computer Systems. When any of the four plotters is employed, the system must include a Multiple Contact Output Control Unit 34-220 and a Multiple Contact Output Unit 34-221.

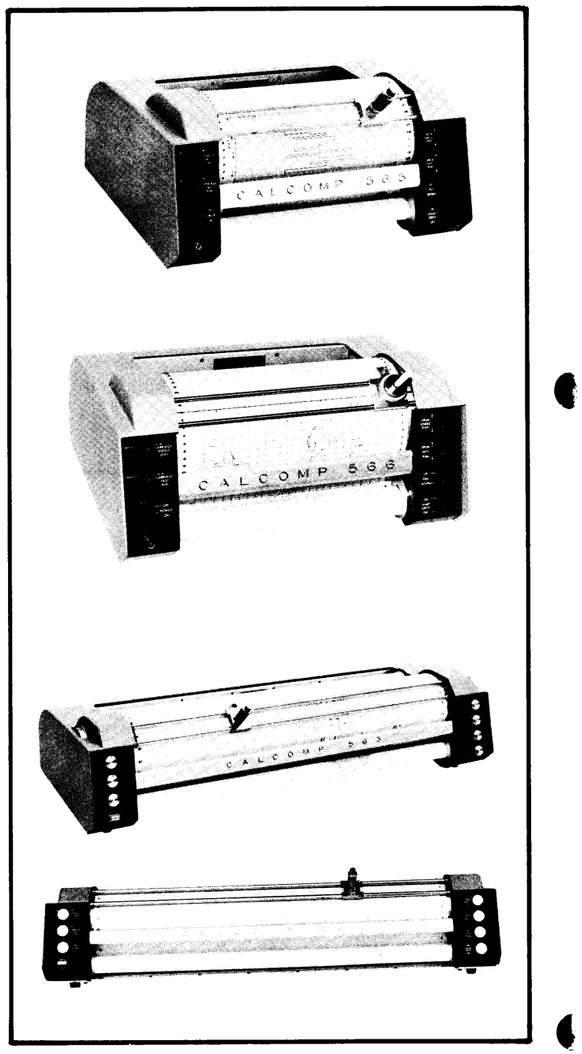
Magnetic Tape Unit

The plotter provides systems users with hard copy graphical presentations of analog quantities from 6-bit data stored (unpacked) in core memory.

Multiple contact outputs from the computer control six relays within the Contact Output Subsystem. These relays in turn control the following plotter functions: +X direction, -X direction, +Y direction, -Y direction, pen up, and pen down.

Each output causes pen movement in increments of 0.01 or 0.005 inch depending upon which plotter model is used. The maximum rate at which data can be output from the multiple contact output is 4.15 milliseconds per point and the plotter is capable of completing movement in a specified direction in 3 milliseconds. An exception to plotter reaction time is evidenced by the command to raise or lower the pen. In this operation, a delay of 100 milliseconds is required.

Plotters operate at either 18,000 or 12,000 points per minute, depending upon the model used, with pen-up or pen-down changes at 600 operations per minute. Plotter Models 563 and 565 provide a 29-1/2-inch plotting surface width while models 562 and 564 provide an 11-inch plotting width.



Digital Incremental Plotters

SECTION XVI TYPICAL OPERATING INSTRUCTIONS

GENERAL

The 340 Control Computer System is designed such that all operating procedures are made as simple and straightforward as possible. A simple set of operating procedures enable an operator to fully utilize the computer hardware to the best advantage in a minimum amount of time.

Although most system installations require different hardware, operating techniques, and procedures, there are those procedures which remain nearly the same in any system. Some typical procedures of this type are given here to illustrate operational simplicity using the 340 Operator's Console.

NOTE: All systems users are provided with applicable operating documentation at the time of equipment delivery.

TYPICAL OPERATING PROCEDURES

A. Display Time

This function causes the time of day to be displayed visually in DISPLAY INDICATORS D1-D4.

1. Set the FUNCTION SELECTOR switches S11 and S12 to read 21. Note that the FUNCTION MATRIX TIME DISPLAY indicator is illuminated.

2. Press the REQUEST button. The time of day (in Navy units) is displayed in indicators D1 through D4. Indicator D5 displays the word TIME. (The time is updated each minute.)

   NOTE: In some systems, the hardware and software is such that the time of day is displayed in the DISPLAY INDICATORS at all times except during periods when other display functions are specifically requested.

B. Trend Display

To execute the Trend Display function, perform the following steps:

1. Set FUNCTION SELECTOR switches S11 and S12 to read 11. Note that the FUNCTION MATRIX-TREND DISPLAY indicator is illuminated.

2. Set MANUAL ENTRY switches S31 through S36 to the address of the word to be displayed.

3. Set switch S21 to 1, 2, or 3, depending upon the display mode desired. See typical display modes below:

   (a) Display Mode 1 (Switch S21 set to 1).

   When switch S21 is set to 1, the contents of the address set on switches S31 through S36 are displayed as a four digit integral number on DISPLAY indicators D1 through D4. D6 displays the sign. Display indicator D5 indicates one of the following:

   (1) Temperature

   (2) Pressure

   (3) Flow

   (4) Instrument

   (5) Memory Value

   NOTE: An analog value not satisfying any of the first three categories above is displayed in D5 as INSTRUMENT.

   If the magnitude of the data is greater than 9999, the data is divided by 10 and is displayed in D1 through D4. In this case, indicator D5 displays a "Red Ball" to denote that the magnitude has been divided by 10.

   (b) Display Mode 2 (Switch S21 set to 2).

   When switch S21 is set to 2, indicators D1 through D5 display the information in the same manner as described for Mode 1 above.

The only exception is that the variable displayed is an instantaneous converted value of the process variable.

(c) Display Mode 3 (Switch S21 set to 3).

This display mode is reserved for direct display of an analog input. When switch S21 is set to 3, indicators D1 through D4 display the instantaneous voltage of the analog input addressed by the settings of switches S31 through S36. D6 displays the sign. Indicator D5 displays the word VOLTS.

NOTE: An illegal setting of switch S21 or switches S31 through S36 causes the Output Writer to print out the words ILLEGAL CODE (or other such printout).

4. Press the REQUEST switch S1 and note the DISPLAY reading.

C. Trend Log

To execute the Trend Log function, perform the following steps:

1. Set FUNCTION SELECTOR switches S11 and S12 to read 12. Note that the FUNCTION MATRIX-TREND LOG indicator is illuminated.

2. Set up MANUAL ENTRY switches S31 through S36 to the address at which the trend log is to start.

3. Press the REQUEST switch S1. When this switch is pressed, the Output Writer will begin printing out a set of variables beginning with the variable stored at the location selected by switches S31 through S36. A typical printout format is as follows:

Time Variable Designator Value No. 1
Time Variable Designator Value No. 2
Time Variable Designator Value No. 3
etc.

Five minutes (or as programmed) after the first set of variables has been logged, a second set is logged. The same variables are logged at regular intervals for one hour (or as programmed), after which the REQUEST switch must again be pressed to continue logging at regular intervals. If during the course of logging, it is desirable to stop the logging function in order to perform another function, the CANCEL switch S2 may be pressed;

however, this switch must be pressed just as the printout completes a full line and prior to beginning printout of a new line. This action cancels succeeding log printouts.

NOTE: Variables may be added to, or eliminated from, the trend log table by requesting or cancelling the trend log point. This function is described in paragraph D.

D. Trend Log Point

To set, add, or delete trend log points from the trend log table, perform the following steps:

1. Set the FUNCTION SELECTOR switches S11 and S12 to read 13. Note that the FUNCTION MATRIX-TREND LOG POINT indicator is illuminated.

(a) To set the trend log points, perform the following:

(1) Set MANUAL ENTRY switches S31 through S36 to the address from which the trend log is to start.

(2) Set MANUAL ENTRY switches S21 and S22 to the desired number of variables to be logged. (10 is normally the number of variables which are logged at any one time. If fewer than ten are desired, the number zero should be set on S22. If a number larger than 10 is set on these switches, the words NO ENTRY (or other such printout) will be typed out on the Output Writer.

(b) To add a trend log point to the log table, perform the following:

(1) Set MANUAL ENTRY switches S31 through S36 to the address of the variable to be added.

(2) Set MANUAL ENTRY switches S21 and S22 to the number corresponding to the position in the printout that the added variable is to appear.

(3) Press the REQUEST switch.

NOTE: If a variable is added to a log table containing 10 variables, the tenth variable is automatically dropped from the printout.

(c) To delete a trend log point from the log table, perform the following:

(1) Set MANUAL ENTRY switches S31 through S36 to the address of the variable to be deleted.

(2) Set MANUAL ENTRY switches S21 and S22 to the number corresponding to the variable in the printout which is to be deleted.

(3) Press the CANCEL switch.

NOTE: A setting of 00 on switches S21 and S22 causes the entire trend log table to be deleted.

To summarize operations, the operator sets the desired function code on FUNCTION SELECTOR switches S11 and S12 and checks that the proper square is lighted on the FUNCTION MATRIX. If the back-lighted function is correct, the operator presses the REQUEST switch S1. When the pushbutton switch is interrogated by the computer, the program reads the Function Selector and other appropriate console switches and starts the desired operation. The program also turns off the REQUEST light. Through the use of the CANCEL pushbutton the operator may stop the action requested even after the action has been started.

APPENDIX H SUMMARY OF OPERATIONS

Class	Symbolic Code	Normal Numeric Code	Operation	Modes	Modes Indexable	Modes Maskable	Execution Time in μs		
							Normal	Indirect	L U or B
LOAD/STORE	LOD A	000440	Load A	NILUB	NI	NILU	12	18	6
	LOD B	016440	Load B	NILUUc	NI	NILUUc	12	18	6
	LOD C	002440	Load C	NILUB	NI	NILU	12	18	6
	LOD X	006440	Load X	NILB	NI	NIL	12	18	6
	LOD I1	003440	Load Index 1	NILB	NI	NIL	12	18	6
	LOD I2	004440	Load Index 2	NILB	NI	NIL	12	18	6
	LOD I3	005600	Load Index 3	NILB	NI	NIL	12	18	6
	LOD M	007440	Load Interrupt Mask M	NILUB	NI	NILU	12	18	6
	LOD Q	013440	Load Interrupt & Guard Cont. Q	NILUB	NI	NILU	12	18	6
	STR A	000400	Store A	NILUB	NILU	NILU	12	18	12, 6
	STR B	016400	Store B	NILU	NILU	----	12	18	12
	STR C	002400	Store C	NILUB	NILU	NILU	12	18	12, 6
	STR X	006400	Store X	NILB	NIL	NIL	12	18	12, 6
	STR I1	003400	Store Index 1	NILB	NIL	NIL	12	18	12, 6
	STR I2	004400	Store Index 2	NILB	NIL	NIL	12	18	12, 6
	STR 13	005400	Store Index 3	NILB	NIL	NIL	12	18	12, 6
	STR M	007400	Store Interrupt Mask M1	NILUB	NILU	NILU	12	18	12, 6
	STR Q	013400	Store Interrupt & Guard Cont. Q	NILUB	NILU	NILU	12	18	12, 6
	STR T2	015400	Store Data Toggles	NILUB	NILU	NILU	12	18	12, 6
ARITHMETIC	MPY n	000740	Multiply (n bits multiplier)	NILUB	NI	----	14+2n	20+2n	14+2n
	DIV n	000700	Divide (n bits quotient)	NILUBH	NIH	----	26+2n	32+2n	26+2n
	ADD	000640	Add	NILUBZ	NI	NILU	12	18	6
	SUB	002640	Subtract	NILUBZ	NI	NILU	12	18	6
	HWA	001640	Half Word Add	NILBZ	NI	NIL	12	18	6
	ADM	006640	Add to Memory	NIZ	NI	NI	18	24	-
	SBM	007640	Subtract from Memory	NIZ	NI	NI	18	24	-
	DIX I1	003640	Decrement Index I1	NILBZ	NI	NIL	12	18	6
	DIX I2	004640	Decrement Index I2	NILBZ	NI	NIL	12	18	6
	DIX I3	005640	Decrement Index I3	NILBZ	NI	NIL	12	18	6
LOGICAL	EXT	000340	Extract	NILUZ	NI	NILU	12	18	6
	EXM	005340	Extract to Memory	NIZ	NI	NI	18	24	-
	MRG	001340	Merge	NILUBZ	NI	NILU	12	18	6
	MGM	004340	Merge to Memory	NIZ	NI	NI	18	24	-
	XOR	002340	Exclusive or	NILZ	NI	NI	12	18	6
	COM EQ	043340	Compare Equal	N	N	N	12	-	-
	COM NE	143340	Compare Not Equal	N	N	N	12	-	-
	COM GR	003340	Compare Greater Than	N	N	N	12	-	-
	COM LS	103340	Compare Less Than	N	N	N	12	-	-
	SCN EQ	046340	Scan Equal	N	N	N			
	SCN NE	146340	Scan Not Equal	N	N	N	6+6c	where c=	
	SCN GR	006340	Scan Greater Than	N	N	N	words	scanned	
	SCN LS	106340	Scan Less Than	N	N	N			
	CMT EQ	047340	Compare Tables Equal	N	N	N			
	CMT NE	147340	Compare Tables Not Equal	N	N	N	6+12c	where	
	CMT GR	007340	Compare Tables Greater Than	N	N	N	c = words	compared.	
	CMT LS	107340	Compare Tables Less Than	N	N	N			
									(Br)
REPLACE/ EXCHANGE	RAW C	002500	Replace A with C	NBr	-	NBr	6	-	6
	RAW I1	003500	Replace A with Index 1	NBr	-	NBr	6	-	6
	RAW I2	004500	Replace A with Index 2	NBr	-	NBr	6	-	6
	RAW I3	005500	Replace A with Index 3	NBr	-	NBr	6	-	6
	RAW XH	001500	Replace A with x, Hold Upper	NBr	-	NBr	6	-	6
	RAW XC	006500	Replace A with x, Clear Upper	NBr	-	NBr	6	-	6
	RAW M	007500	Replace A with Int. Mask M	NBr	-	NBr	6	-	6
	RAW Q	013500	Replace A with Int. Cont. Q	NBr	-	NBr	6	-	6
	RAW G	016500	Replace A with G	NBr	-	NBr	6	-	6
	RAW T1	017500	Replace A with Address Toggles	NBr	-	NBr	6	-	6
	RAW T2	015500	Replace A with Data Toggles	NBr	-	NBr	6	-	6
	RWA C	042500	Replace C with A	NBr	-	NBr	6	-	6
	RWA I1	043500	Replace I1 with A	NBr	-	NBr	6	-	6
	RWA I2	044500	Replace I2 with A	NBr	-	NBr	6	-	6
	RWA I3	045500	Replace I3 with A	NBr	-	NBr	6	-	6
									(Br)
	RWA X	046500	Replace X with A	NBr		NBr	6	-	6
	RWA M	047500	Replace Int. Mask M with A	NBr	-		12	-	12
	RWA Q	053500	Replace Int. Cont. Q with A	NBr	-		12	-	12
	RWA G	056500	Replace G with A	NBr	-	NBr	6	-	6
	EAW B	016600	Exchange A with B	NBr	-	-	6	-	6
	EAW C	002600	Exchange A with C	NBr	-	NBr	6	-	6
	EAW XH	001600	Exchange A with X, Hold Upper	NBr	-	NBr	6	-	6
	EAW XC	006600	Exchange A with X, Clear Upper	NBr	-	NBr	6	-	6
	EAW I1	003600	Exchange A with Index 1	NBr	-	NBr	12	-	12
	EAW I2	004600	Exchange A with Index 2	NBr	-	NBr	12	-	12
	EAW I3	005600	Exchange A with Index 3	NBr	-	NBr	12	-	12
	EAW M	007600	Exchange A with Int. Mask M	NBr	-	NBr	12	-	12
	EAW Q	0013600	Exchange A with Int. Cont. Q	NBr	-	NBr	12	-	12
								(Condition Met)	

Class	Symbolic Code	Normal Numeric Code	Operation	Modes	Modes Indexable	Modes Maskable	Execution Time in μs		
							Normal	Indirect	L, U or B
BRANCH	BUN	000000	Branch Unconditionally	NIX	NI	-	6	12	6
	BZE	016000	Branch if A Zero	NIX	NI	-	6	12	6
	BZM	001000	Branch if A Masked Zero	NIX	NI	-	6	12	6
	BNZ	012000	Branch if A not Zero	NIX	NI	-	6	12	6
	BNM	002000	Branch if A Masked not Zero	NIX	NI	-	6	12	6
	BPO	013000	Branch if A Positive	NIX	NI	-	6	12	6
	BNG	003000	Branch if A Negative	NIX	NI	-	6	12	6
	BLB	004000	Branch if A Contains Low Bit	NIX	NI	-	6	12	6
	BNXI1	007000	Branch if Index 1 not Zero	NIX	NI	-	6	12	6
	BNXI2	010000	Branch if Index 2 not Zero	NIX	NI	-	6	12	6
	BNXI3	011000	Branch if Index 3 not Zero	NIX	NI	-	6	12	6
	BOF	005000	Branch if Overflow Indicator On	NIX	NI	-	6	12	6
	BCY	015000	Branch if Carry Indicator On	NIX	NI	-	6	12	6
	BCP	006000	Branch if Core Parity Error	NIX	NI	-	6	12	6
	BDP	014000	Branch if Drum or Direct Access Parity Error	NIX	NI	-	6	12	6
SHIFT/ NORMALIZE	ALA	040540	Shift A Left Arithmetic	NX	-	-			
	ALO	041540	Shift A Left Logical Open	NX	-	-			
	ALC	042540	Shift A Left Logical Closed	NX	-	-			
	ARA	043540	Shift A Right Arithmetic	NX	-	-			
	ARO	044540	Shift A Right Logical Open	NX	-	-	8+2n where		
	BLA	045540	Shift A, B Left Arithmetic	NX	-	-	n = places shifted		
	BLL	046540	Shift A, B Left (A Arith, B Log) Open	NX	-	-			
	BLO	047540	Shift A, B Left Logical Open	NX	-	-			
	BLC	050540	Shift A, B Left Logical Closed	NX	-	-			
	BRA	052540	Shift A, B Right Arithmetic	NX	-	-			
	BRL	051540	Shift A, B Right (A Arith, B Log Open	NX	-	-			
	NAA	053540	Normalize A Arithmetic	NX	-	-			
	NAO	054540	Normalize A Logical Open	NX	-	-	8+2c where		
	NAC	055540	Normalize A Logical Continued	NX	-	-	c = number of shift		
	NBA	056540	Normalize A, B Arithmetic	NX	-	-	to normalize		
	NBL	057540	Normalize A, B (A Arith, B Log) Open	NX	-	-			
PROGRAM CONTROL	NOP	017000	No operation	N	-	-	6	6	-
	STP	046600	Stop and Branch	NI	NI	-	6	12	-
	SRB	045600	Set Return and Branch	NI	NI	-	12	18	-
	SET	064600	Set	N	-	-	6	-	-
	RST	044600	Reset	N	-	-	6	-	-
	STA	047600	Set Core Address	N	-	-	6	-	-
INPUT/OUTPUT CONTROL	ENA	040600	Enable	N	-	-	12	-	-
	DIS	060600	Disable	N	-	-	12	-	-
	CIN	041600	Direct Contact Input	N	-	N	18	-	-
DRUM/CORE TRANSFER	SDC	000300	Start Drum-to-Core Transfer	NI	NI	-	18-24	24-30	-
	SCD	040300	Start Core-to-Drum Transfer	NI	NI	-	18-24	24-30	-
	HDT	001300	Halt Drum Transfer	NI	NI	-	24	30	-
OPERATION EXTENSION	OEX	variable	Operation Extension	NI	NI	-	12	18	-