REPORT

Z50-1N

MOBIDIC 7A

FINAL DESIGN PLAN

SYLVANIA ELECTRONIC SYSTEMS
A unit of
GENERAL TELEPHONE & ELECTRONICS
GENERAL SYSTEM

GAZZOLA

MOBILE DIGITAL COMPUTER PROGRAM

MOBIDIC 7A

FINAL DESIGN PLAN

Signal Corps
Technical Requirements
SCL 1959A

Contract No. DA-36-039-SC-85117

Approved by: ______________
G. M. Sokol, Manager
MOBIDIC Program

26 August 1960

SYLVANIA ELECTRONIC SYSTEMS
A Division of Sylvania Electric Products Inc.
DATA SYSTEMS OPERATIONS
189 B Street - Needham 94, Massachusetts

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

Section Page

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont.)

LIST OF ILLUSTRATIONS (Cont.)

LIST OF TABLES

SECTION I

INTRODUCTION

MOBIDIC 7A is a general-purpose, high-speed, mobile computer intended for field use by the United States Army. It has been built by the Data Systems Operations of Sylvania Electronic Systems, a Division of Sylvania Electric Products, Inc., under the terms of contract number DA-36-039-SC-85117. MOBIDIC 7A conforms to Signal Corps Technical Requirements SCL 1959A, dated 14 October 1959. The computer provides a reliable mobile computing facility for use by military commanders in the field for combat support and combat control data processing and for combat computation. It can also serve as a vital element in MASS, the modern army supply system.

The computer is housed in two semi-trailer vans. A third semi-trailer van serves as a maintenance van. Tables 1-1, 1-2, and 1-3 list the Sylvania-manufactured, Sylvania-purchased, and government-furnished equipment included in the MOBIDIC 7A computer. The organization of the system is such that equipment in van 2 can be adapted for operation independent of equipment in van 1. The arrangement of the MOBIDIC 7A equipment in vans 1 and 2 is shown in Figures 1-1 and 1-2, respectively.

MOBIDIC 7A is essentially similar to other MOBIDIC computers. The principal difference is the addition of an off-line control system. MOBIDIC 7A includes both a real-time system and a program-interrupt feature.

This final design plan includes information additional to that included in the Final Design Plans of MOBIDICs A, B, C, and D and the Addendum to the Final Design Plan for MOBIDIC D. Reference should be made to those documents for a complete understanding of the MOBIDIC 7A design. This design plan includes the following sections:

I Introduction

II Off-Line Control System

III Line Printer and Line Printer Buffer

IV Card Reader-Punch and Card Reader-Punch Buffer

V Logic and Mechanization Designations

VI Mechanization Lists

VII Van Layout

VIII Element Tester

TABLE 1-1. MOBIDIC 7A EQUIPMENT LIST
SYLVANIA-MANUFACTURED EQUIPMENT

Unit	Description	Quantity
S101-2	Central Processor, with Program Interrupt	1
S201	Console, A	1
S307	Power Supply Set (Van 2)	1
S308	Power Supply Set (Van 2)	1
S310-2	Power Supply Set (Van 1)	1
S311-2	Power Supply Set (Van 1)	1
S312-2	Power Supply Set (Van 1)	1
S353	AC Control and Distribution (Van 1)	1
S354	AC Control and Distribution (Van 2)	1
S401	Core Memory (4, 096 words)	2
S601-2	Converter	2
S602A	Off-Line Control Unit	1
S602B	Off-Line Control Unit	1
S702	Paper Tape Reader and Punch (8-channel), Device Switching Units (See Note 1), and Buffers (See Note 2)	1
S703A-2	Line Printer Buffer	1
S703B-2	Line Printer Control	1
S704-2	Card Reader-Punch Buffer	2
S706	Magnetic Tape Units (Militarized Magnetic Tape Transports)	6
S707	Paper Tape Reader and Punch (5-channel) and Buffer (See Note 3)	1
S709	Flexowriter Table	1

TABLE 1-1. MOBIDIC 7A EQUIPMENT LIST SYLVANIA-MANUFACTURED EQUIPMENT (Cont.)

Unit	Description	Quantity
S720-2	Magnetic Tape Unit Device Switching Units	1
S721	Off-Line Control System Device Switching Units	1
S901	Element Tester	1
S904	Teletype Equipment Stand	1

Note 1: For the 8- and 5-channel paper tape readers and punches and for the Flexowriter.

Note 2: For the 8- and 5-channel paper tape readers and punches (except for part of the 5-channel paper tape punch buffer) and the Flexowriter.

Note 3: Balance of 5-channel paper tape punch buffer.

TABLE 1-2. MOBIDIC 7A EQUIPMENT LIST, SYLVANIA-PURCHASED EQUIPMENT

Description	Mounting	Quantity
Line Printer	In Van 2	1
Card Reader-Punch	In Van 2	2
Militarized Magnetic Tape Transport	See Note 1	11
Flexowriter	In Van 1, in S709	1
Paper Tape Reader	See Note 2	3
Paper Tape Punch		
8-channel	See Note 3	2
5-channel	In Van 1, in S707	1
Paper Tape Spooler	See Note 2	3

Note 1: 8 in Van 1, 2 in Van 2, 1 in Van 3.

Note 2: 2 in Van 1 (1 in S702, 1 in S707); 1 in Van 2, in S721.

Note 3: 1 in Van 1, in S702; 1 in Van 2, in S721.

TABLE 1-3. MOBIDIC 7A EQUIPMENT LIST, GOVERNMENT-FURNISHED EQUIPMENT

Description	Mounting	Quantity
Air Conditioner	2 in each Van	6
Space Heater	1 in each Van	3
Field Telephone	See Note 1	2
Teletype Printer	In Van 1, on S904	1
Teletype Reperforator	In Van 1, on S904	1
Power Unit (100 KW)	- - - - - - - - - - -	2
M393A2 Van	- - - - - - - - - - -	3

Note 1: 1 in Van 1, 1 in Van 2

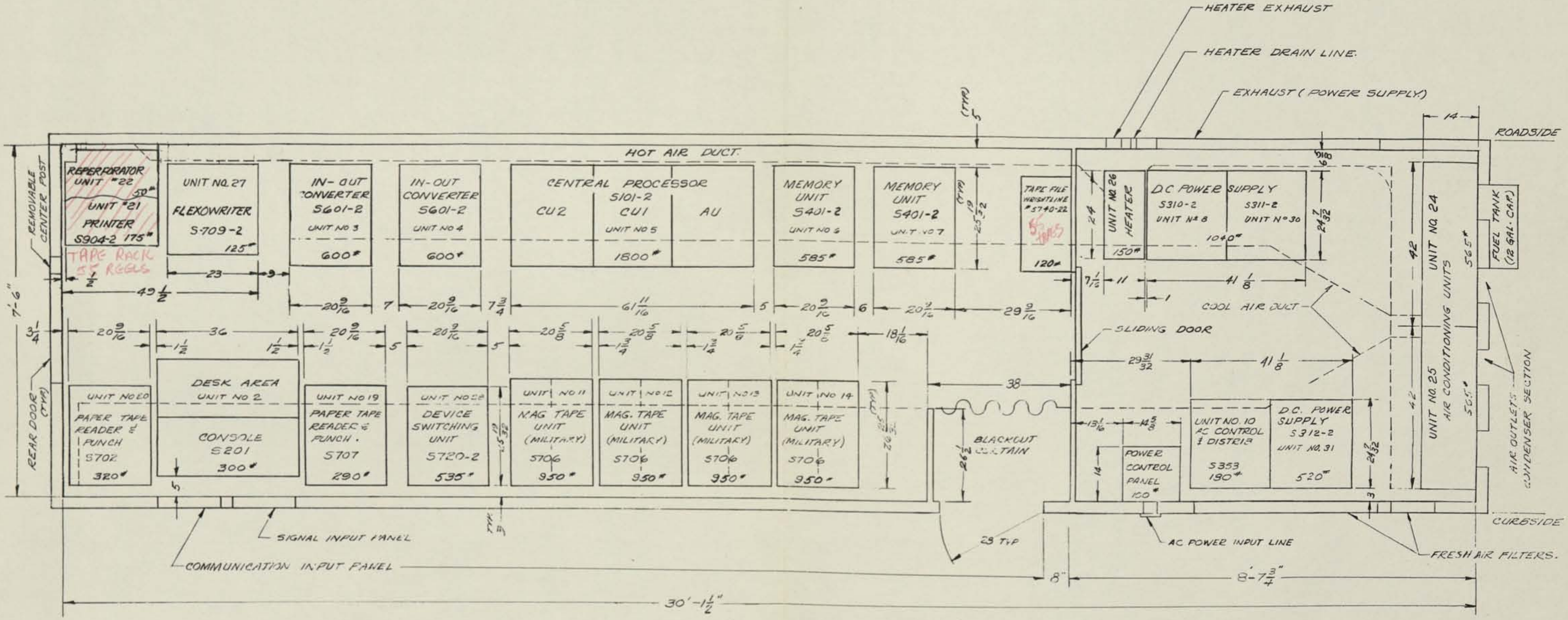

HEATER EXHAUST
HEATER DRAIN LINE
EXHAUST (POWER SUPPLY)
ROADSIDE
CURBSIDE
HOT AIR DUCT
REMOVABLE CENTER POST
REAR DOOR (TYP)
REPERFORATOR UNIT #22 50#
UNIT #21 PRINTER S904-2 175#
TAPE RACK 55 REELS
UNIT NO. 27 FLEXOWRITER S-709-2 125#
IN-OUT CONVERTER S601-2 UNIT NO 3 600#
IN-OUT CONVERTER S601-2 UNIT NO 4 600#
CENTRAL PROCESSOR S101-2
CU2
CU1
AU
UNIT NO 5 1800#
MEMORY UNIT S401-2 UNIT NO 6 585#
MEMORY UNIT S401-2 UNIT NO 7 585#
TAPE FILE WEIGHTLINE #S740-22 120#
UNIT NO. 26 HEATER 150#
D.C POWER SUPPLY
S310-2 UNIT Nº 8
S311-2 UNIT Nº 30
1040#
UNIT NO. 24
UNIT NO. 25 AIR CONDITIONING UNITS
565#
FUEL TANK (12 GAL. CAP.)
AIR OUTLETS CONDENSER SECTION
COOL AIR DUCT
SLIDING DOOR
UNIT NO 20 PAPER TAPE READER & PUNCH S702 320#
DESK AREA UNIT NO 2
CONSOLE S201 300#
UNIT NO 19 PAPER TAPE READER & PUNCH. S707 290#
UNIT NO 28 DEVICE SWITCHING UNIT S720-2 535#
UNIT NO 11 MAG TAPE UNIT (MILITARY) S706 950#
UNIT NO 12 MAG. TAPE UNIT (MILITARY) S706 950#
UNIT NO 13 MAG. TAPE UNIT (MILITARY) S706 950#
UNIT NO 14 MAG. TAPE UNIT (MILITARY) S706 950#
BLACKOUT CURTAIN
UNIT NO. 10 AC CONTROL & DISTRIB S353 190#
D.C. POWER SUPPLY S312-2 UNIT NO. 31 520#
POWER CONTROL PANEL 100#
SIGNAL INPUT PANEL
COMMUNICATION INPUT PANEL
AC POWER INPUT LINE
FRESH AIR FILTERS
23 TYP
7'-6"
30'-1½"
8"
8'-7¾"
49½
23
36
38
61 11/16
29 31/32
41 1/8
14
42

Figure 1-1. MOBIDIC 7A Van 1 Layout

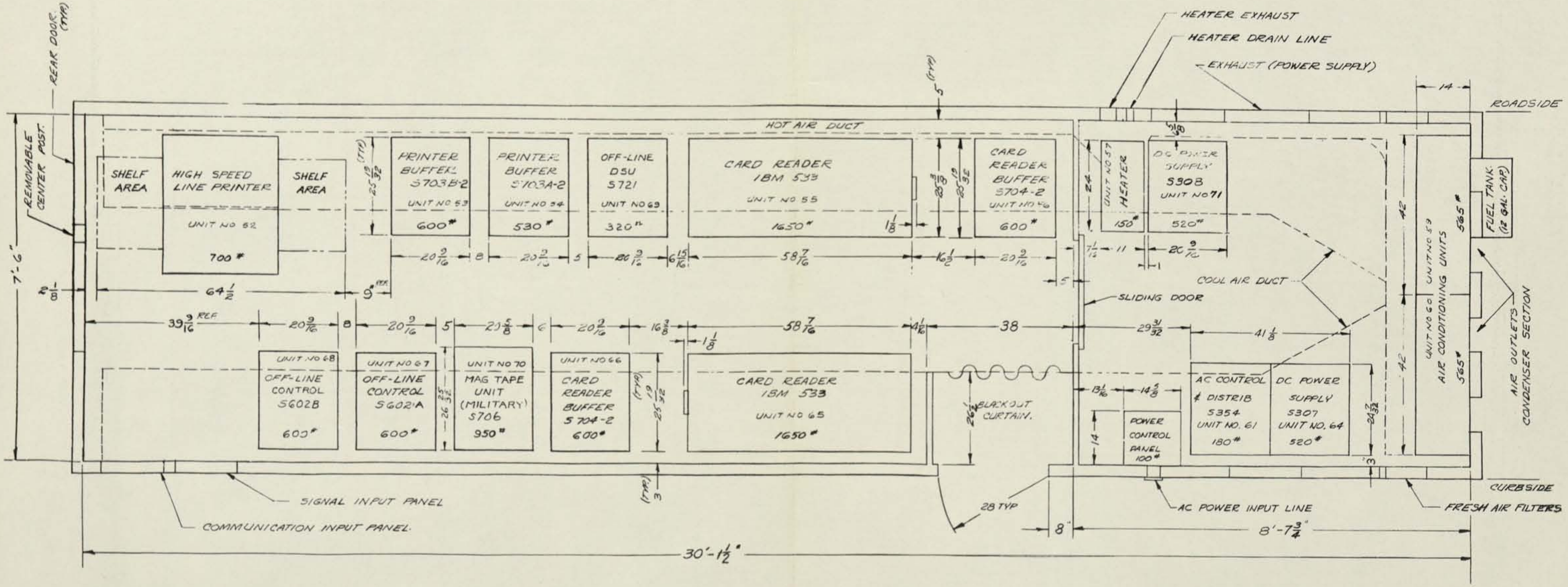


Figure 1-2. MOBIDIC 7A Van 2 Layout

SECTION II

OFF-LINE CONTROL SYSTEM

2.1 GENERAL

One of the features of MOBIDIC 7A is the off-line control system. This system permits off-line data processing without interference with computer data processing. The off-line control system is described in detail in this section.

2.2 PURPOSE AND USE

The MOBIDIC Off-Line Control Unit (OLCU) has been designed to increase the effectiveness of MOBIDIC data-processing systems. The OLCU enables transfer of data between MOBIDIC in-out devices, the transfer taking place independent of central computer control and despite differences in device operating rates. The OLCU makes it possible, for example, to transfer data between magnetic tape and in-out media having lower operating rates. This is often desirable since substantial savings in central computer operating time can be realized by taking advantage of the relatively high rate of data transfer which can be achieved by using magnetic tape as the in-out media during execution of in-out instructions. The OLCU enables the format of data to be changed. Data stored on paper tape, for example, can be punched on cards for further processing. The OLCU is highly useful in applications which require processing of a large volume of data which either has been prepared off-line or is to be used off-line. The in-out devices included in the MOBIDIC 7A off-line control system are listed in Table 2-1. The OLCU is useful in any Fieldata application that requires off-line transfer or conversion of data since the formats used by MOBIDIC computers are those used for Fieldata. In general, the term conversion will be used in this section to describe either transfer of data between in-out media or conversion of data from one format to another. Both transfer and conversion can be performed simultaneously by the OLCU. The relationship between the MOBIDIC 7A in-out and off-line control systems is shown, in block diagram form, on Figure 2-1.

TABLE 2-1. MOBIDIC 7A OFF-LINE CONTROL SYSTEM IN-OUT DEVICES

Unit	Device	Quantity
S706	Militarized Magnetic Tape Transports	2 *
S721	Paper Tape Punch, 8-channel	1
	Paper Tape Reader, 8-channel	1
- - -	Card Reader-Punch	2 *
- - -	Line Printer	1 *

* = Capable of being used either on-line or off-line.

For a description of the on-line, off-line interlocks, see 2.10.

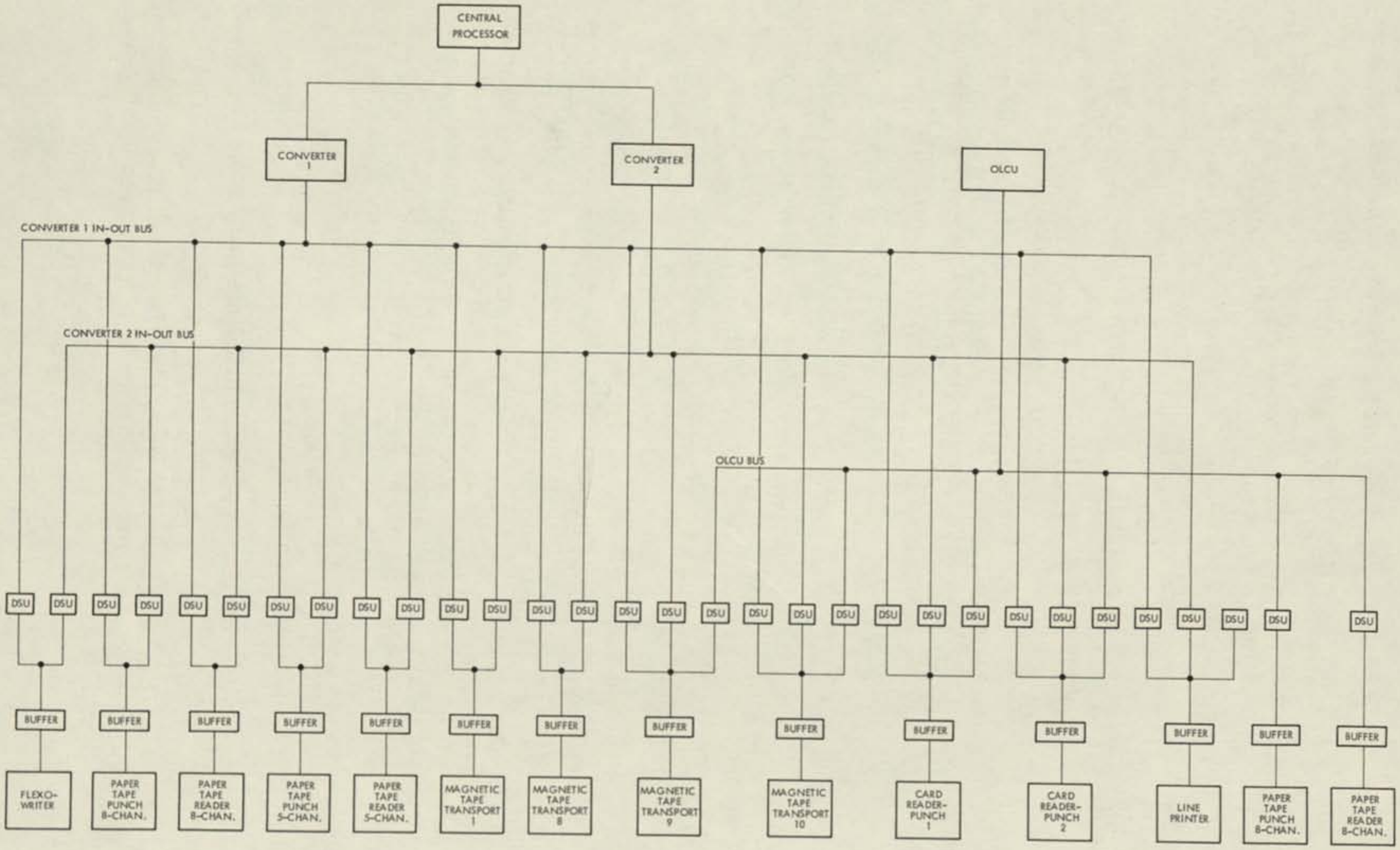


Figure 2-1. MOBIDIC 7A In-Out and Off-Line Control Systems, Block Diagram

Table 2-2 lists the devices, data rates, and formats used. The rates listed are those of the in-out devices presently used in MOBIDIC in-out systems. Conversion may be performed between any input and output device listed. Any necessary changes in format, including conversions between Hollerith and Fieldata code, are performed by the OLCU. The conversions actually carried out depend upon the control settings on the operator's panel.

The OLCU consists essentially of a memory, several high-speed buffer registers, controls for operating the in-out devices and OLCU, and associated circuits. The memory is of the magnetic-core type. It is capable of storing 672 Fieldata characters or 336 Hollerith characters. The memory and registers are described in 2.6 and 2.7. The controls are described in 2.6 and 2.9. The circuits are described in 2.7. OLCU equipment is housed in two MOBIDIC-type enclosures. These enclosures, the S602A and S602B units, are described in 2.8.

Transfer operations, also called conversion operations, are carried out one block of information at a time. A typical conversion cycle includes the following sequence of operations:

1. Clear the OLCU and check for operator errors.
2. Start the selected input device.
3. Transfer information from the selected input device to the OLCU memory. The transfer continues either until the OLCU memory is filled, or until an end-of-block or end-of-file mark is sensed, or until the specified number of cards (4 maximum) have been read. A stop code on paper tape input is considered to be equivalent to end-of-file mark. The card reader-punch generates an end-of-file signal if the last cards in the deck being run are fed through the reader with the card hopper empty.
4. Stop the selected input device.
5. Start the selected output device.
6. Transfer information from the OLCU memory to the selected output device.
7. If end-of-file mark (or the equivalent) was not reached during step 3, repeat the entire sequence of operations starting at step 2.

TABLE 2-2. OLCU DEVICE RATES AND FORMATS

Input Devices		
Media	Rate *	Formats
Magnetic Tape	45, 000 char/sec	Standard Fieldata
Paper Tape	270 char/sec	Standard Fieldata
Punched Cards	200 cards/min	Row Fieldata Column Fieldata Hollerith
Output Devices		
Media	Rate	Formats
Magnetic Tape	45, 000 char/sec	Standard Fieldata
Paper Tape	100 char/sec	Standard Fieldata
Punched Cards	100 cards/min	Row Fieldata Column Fieldata Hollerith
Line Printer	600 lines/min	Same as when used on-line

* = Rates listed are those of presently-used devices in the MOBIDIC 7A.

8. If an end-of-file mark (or equivalent) was reached during step 3, stop the cycle of operations after step 6 and halt both the OLCU and the selected output device.

Since the OLCU completes a one-character cycle in ten microseconds, it can handle data more rapidly than can any presently-known and available MOBIDIC in-out device. It can, for example, operate with magnetic tapes having an operating rate of 90, 000 characters per second. The off-line control system data handling rates are device-limited, rather than OLCU-limited.

The OLCU can be used in a blockette mode of operation when using either paper tape or magnetic tape input. When operating in this mode not all of the information contained in a block is converted. Instead, only that contained in part of the block is converted. Each selected portion within the block is called a blockette. Use of the blockette mode provides specialized searching capacity and permits on-line preparation of magnetic tapes on which are stored blocks of information longer than those capable of conversion during a single cycle of OLCU operation.

2.3 OPERATIONAL CONSIDERATIONS

2.3.1 Compatibility with MOBIDIC In-Out Devices

The design of the OLCU is such that it is operationally compatible with all in-out devices (and their associated buffers) previously incorporated in MOBIDIC systems. One of the primary design objectives was to make the transfer and conversion capabilities of the OLCU as universal as feasible. This requirement has been met. It is possible to convert or transfer data from any MOBIDIC input media to any MOBIDIC output media with but one exception. The only exception is that card to card transfers cannot be performed. The data formats used in off-line control operation are exactly the same as those of the various in-out media incorporated in MOBIDIC. Magnetic and paper tapes can be handled in either the interpret sign or the non-interpret sign mode. Punched cards of the type used in the MOBIDIC card reader-punch may also be handled in either of these modes. Because of the widespread use of Hollerith code in electronic data processing (EDP), provision has been made in the design of the OLCU for automatic conversion between

Hollerith and Fieldata codes when such is desirable. Because information in the OLCU memory is stored in column format, provision has been made for automatic row to column conversion during card input and for automatic column to row conversion during card output. When operating with punched cards, selection of the interpret sign mode, by operation of the BINARY mode switch, will cause the OLCU to read or punch the cards as row cards (information punched on the cards in horizontal rows, as distinct from column cards on which information is punched in vertical columns). The format of the row cards is the same as that of cards read or punched during normal MOBIDIC system on-line operation of the card reader-punch. Selection of the non-interpret sign mode, by operation of the ALPHANUM (alphanumeric) mode switch, will cause the OLCU to perform an automatic row to column or column to row conversion, as necessary. Every column on the card is then read or punched as if it consisted of two Fieldata characters (6 data bits each). The Hollerith mode may also be selected, by operation of the HOLL (Hollerith) mode switch, during OLCU operation when using punched cards. In that mode the OLCU automatically performs the required column to row or row to column conversion. Each column is treated as if it consisted of one Hollerith character. Conversions between the Hollerith and Fieldata codes is carried out automaticlly by the OLCU when operating in the Hollerith mode.

2.3.2 Format Control

A minimum of format control has been included in the design of the OLCU. Such a design approach offers the following three advantages:

1. Simplicity.

2. Ready adaptability for future use with in-out devices or device formats not presently employed in MOBIDIC systems.

3. Compatibility and similarity of programming for on-line and off-line operation of the in-out equipment.

2.3.3 Padding

Provision has been made in the OLCU design for processing padding characters or bits in a manner appropriate to the in-out devices or formats used. The following examples indicate the use of padding characters by the OLCU.

Cards handled by the card reader-punch during on-line operation are read or punched under computer control as if using an 84-column format. Such a format is compatible with the standard six or seven character MOBIDIC word length. During off-line operation cards are also handled as if they had a similar 84-column format. The four columns additional to the 80 columns on the standard card are always generated or deleted by equipment external to the OLCU. The extra, or padding, columns are carried through the conversion process in the OLCU although they are not actually punched or read. When converting information from magnetic tape to cards punched in the Hollerith code 84 characters must be physically present on the tape for each complete card punched. Four of these characters are the padding characters. The padding characters are not punched and, therefore, never appear on the punched card. They are, however, processed through the OLCU.

77770505

At times there may not be an integral number of six-character groups on paper tape. An integral number of such groups must, however, appear on any magnetic tapes to be read by the computer (if an integral number of such groups is not read a magnetic tape reading error signal is generated). The OLCU has been so designed that when converting information from paper tape to magnetic tape in the non-interpret sign mode sufficient padding characters are automatically generated by the OLCU to assure that the next larger integral number of six-character groups will be present on the magnetic tape.

2.3.4 Set-Up and Error Control

An optimum compromise between simplicity and flexibility, within the broad general requirement of over-all OLCU design simplicity, has been achieved in the design, arrangement, and function of the controls and indicators on the operator's panel. Except for actual loading of the in-out devices to be used during OLCU operation, any conversion can be completely set up and flexibly controlled by appropriate operation of the panel-mounted controls. OLCU operational and error control procedures are such that the operator is required to perform only simple actions. These actions are essentially similar for each of the in-out devices with which the OLCU is employed.

A complete error-checking capability, extending to all in-out devices and formats used, has been incorporated in the OLCU design. This has been done to assure detection of all errors that might occur during conversion. It is imperative that such errors be detected at that time since in many cases the converted information is not used until some time later. Provision has been made not only for controlling the various error-checking and correcting features but also for ignoring errors when such is desirable. For example, error indications can be ignored when the urgent need for a particular conversion is more important than the general accuracy level of that conversion.

2.3.5 Blockette Mode

Capability for operating in the blockette mode has been included in the design of the OLCU. This capability permits more efficient computer operation and increases OLCU flexibility. Computer operation is more efficient since long blocks of information can be written on magnetic tape at relatively high speed for subsequent off-line processing. The advantages of increased OLCU flexibility are particularly evident when magnetic or paper tape is used as the input media. When paper tape input is used blocks of information of variable length can be written on magnetic tape or cards can be punched with a variable number of characters per card. When magnetic tape input is used specific information within a block can be selected for conversion.

2.3.6 On-Line and Off-Line Operation

As noted previously, the OLCU has been designed to operate with the in-out devices and their buffers which are already in use in MOBIDIC systems. Certain of these devices can be used either on-line or off-line. The following example indicates the utility of this device sharing arrangement. Magnetic tapes can be prepared on-line for off-line processing and the off-line processing carried out without physical handling of the tape reel. Also, such tapes can be prepared off-line and then used, without tape reel handling, for on-line input to the computer.

2.4 DESIGN CONSIDERATIONS

The MOBIDIC off-line control system was designed to meet several general requirements. It was necessary that the design:

1. Allow continuous operation of the complete MOBIDIC on-line system without interference by the off-line control system.

2. Prevent improper use of in-out devices. The design is such that attempts to do so can occur only as a result of an operator's error in using the off-line control equipment.

3. Indicate attempts to use an in-out device improperly.

More detailed design considerations are described in paragraphs 2.4.1 through 2.4.5.

2.4.1 System Compatibility

The off-line control system has been designed for compatibility with existing MOBIDIC in-out systems. Not only was it considered necessary that the OLCU be capable of operating with the same types of in-out devices and buffers as are used in such systems but it was furthermore considered desirable to so design the off-line control system that several such devices could be used either on-line or off-line. Accordingly, the OLCU bus and device switching units (DSUs) as well as the OLCU have been designed to meet these objectives. The same DSU can be used for either on-line or off-line operation in the case of devices capable of such operation. During on-line operation the DSU is used with the MOBIDIC converter; during off-line operation it is used with the OLCU. Such an arrangement is required, for example, when four converters and an OLCU are included in a particular MOBIDIC in-out system. In such a system three of the four DSUs associated with each shared (i.e., on-line, off-line) device are used during on-line operation. The fourth DSU can be connected, by appropriate manual procedures, for either on-line or off-line operation.

2.4.2 Converter - OLCU Similarity

The design of the OLCU has been based to a great extent upon the experienced gained on the standard MOBIDIC converter. This has been possible because of the similarities between the two units. Control of in-out device operation by the OLCU is essentially similar to that exercised by the converter. Substantial savings in the number of transistors required have been realized by simplifying the converter design wherever feasible. The control area, for example, is one in which the simpler design has been employed to advantage. The OLCU does not need to modify orders and it is not required to regulate execution of rewrite instructions. Elimination of these functions has allowed use of a simpler design, resulting in a need for fewer transistors. Use of fewer transistors has also simplified trouble-shooting and maintenance procedures.

2.4.3 Design Approach

The OLCU design approach was determined after careful analysis of the operational requirements. Although the OLCU in general has been designed as a wired program machine, a variety of design techniques, each suited to the requirements of a particular area of the OLCU, have been employed. The desire for simplicity of design and ease of maintenance dictated division of operational control into three general sections. These are the internal, external, and master control sections. The external control section regulates in-out device operation. All of the various operations, or instructions, which must be performed within the OLCU to carry out a particular conversion are, in effect, mechanized within the internal control section. The master control section, at times referred to as the program control section, governs the sequence in which the OLCU performs the selected conversion.

2.4.4 Memory Capacity

The capacity of the magnetic-core OLCU memory was chosen as a result of a design compromise. Use of a small capacity memory would contribute toward OLCU component cost reduction. Use of a larger capacity memory, however, would permit writing lengthy blocks of information on magnetic tape during a single conversion operation. This would have the effect of decreasing the operational cost of reading or writing information on magnetic tape. The size of the memory chosen, 672 Fieldata characters, is small enough to hold down component costs, yet it is sufficiently large to provide storage for relatively long blocks of

112 words.

information. It is large enough, for example, to store as much information as can be read from or written onto four punched cards. It should be noted, however, that memory storage capacity equal to that required to store information for one card is required for conversion checking purposes. Therefore, only as much information as can be stored on three cards can be converted during any single conversion when the card checking feature is being used.

2.4.5 Device Switching Unit Interlocks

Interlocks have been provided in the design of those device switching units (DSUs) associated with in-out devices which are capable of either on-line or off-line operation. The interlocks prevent improper use of such devices. The design is such that a device actually being used off-line cannot be used on-line by the MOBIDIC system and a device actually being used on-line cannot be used off-line.

2.4.6 Verify Mode

The OLCU design provides for operation of the equipment in a verify mode. This mode of operation permits off-line checking of paper tape punched under OLCU control. In the verify mode the output cycle is inhibited and input is checked for parity errors. The OLCU imposes no limit on block length when tape input is used during operation in the verify mode.

2.5 DESIGN DESCRIPTION

The design of the off-line control system and its equipment is described in the three major paragraphs which follow. System design is detailed in paragraph 2.6, circuit design in paragraph 2.7, and mechanical design in paragraph 2.8.

2.6 SYSTEM DESIGN

The overall relationship between various items in the MOBIDIC computer and the off-line control system is shown in block diagram form on Figure 2-1.

2.6.1 OLCU Block Diagram

The OLCU is shown in block diagram form on Figure 2-2. A brief listing of the basic function of each portion of the OLCU represented by a block on the diagram follows. More complete descriptions of various aspects of the OLCU are given in paragraphs 2.6.2 through 2.6.7

1. Memory - This is a magnetic-core type memory which provides storage capacity for up to 672 Fieldata characters (or 336 Hollerith characters).

2. MA Counter - The MA (memory address) counter (MAC) used to address the memory location into which information is to be stored or from which information is to be extracted.

3. MBR - The MBR (memory buffer register) is a 16-bit register used to store information being transferred to or from memory.

4. IBR - The IBR (input buffer register) is an 8-bit register which accepts information entering the OLCU from the selected in-out device.

5. OBR - The OBR (output buffer register) is an 8-bit register used to hold the information going from the OLCU to the selected in-out device.

6. FH Code Converter - The FH (Fieldata to Hollerith) code converter is used when information is to be converted from Fieldata code form to Hollerith code form. Appropriate control and parity bits are generated in this converter. The FH code converter accepts information from the IBR and sends it to the MBR.

7. HF Code Converter - The HF (Hollerith to Fieldata) code converter is used when information is to be converted from Hollerith code form to Fieldata code form. The HF code converter accepts information from the MBR and sends it to the OBR.

8. Internal Control - The internal control section regulates transfers of information between the OBR, IBR, MBR, and memory.

9. External Control - The external control section regulates transfers between the IBR, OBR, and in-out devices.

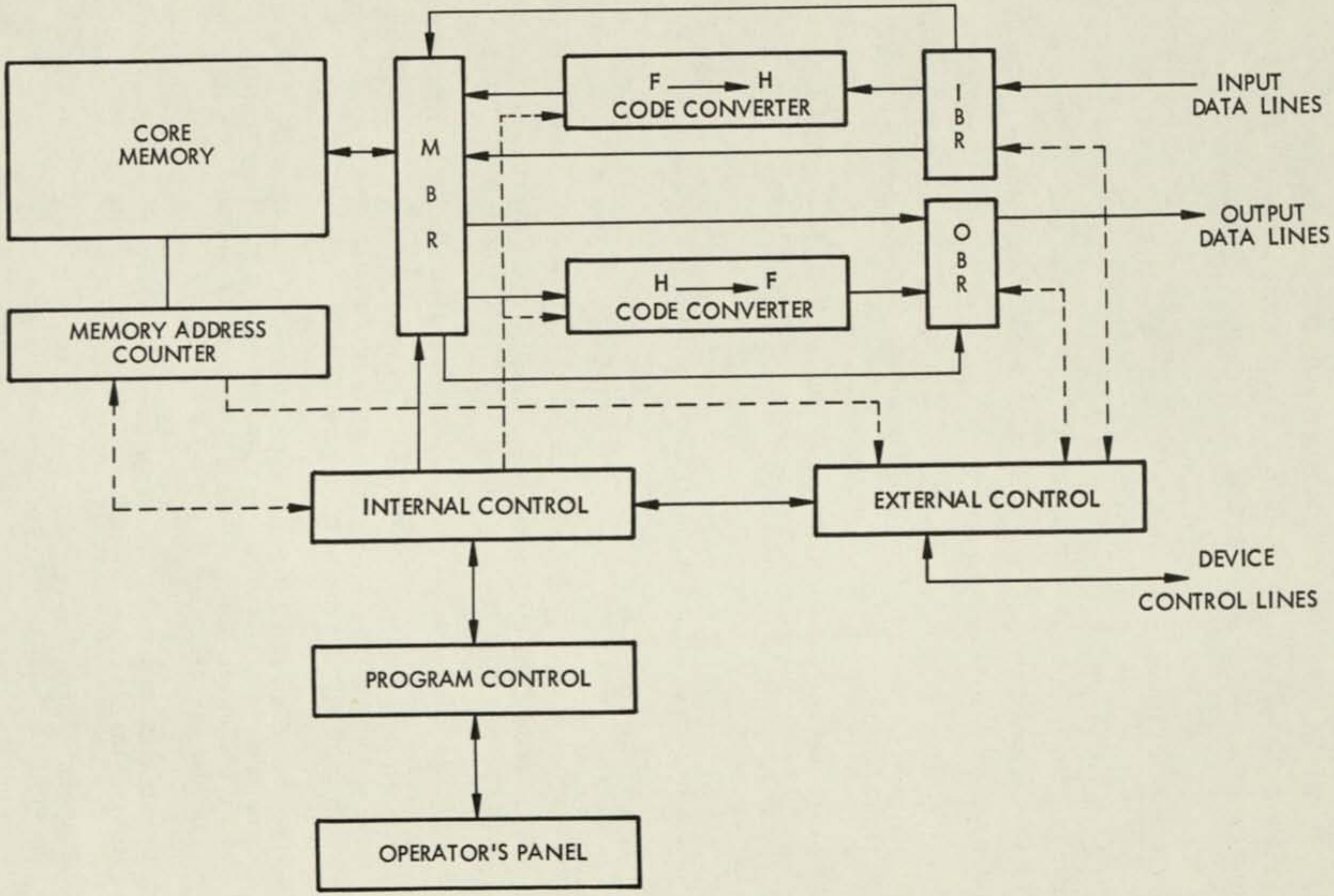


Figure 2-2. Off-Line Control Unit Block Diagram

10. Program Control - The program control section provides all of the status and instruction signals necessary for OLCU operation and regulates the sequence of OLCU operations on the basis of the set-up on the operator's panel.

11. Operator's Panel - The operator's panel contains the controls and indicators required for manual control of OLCU operation.

2.6.2 OLCU Control Center

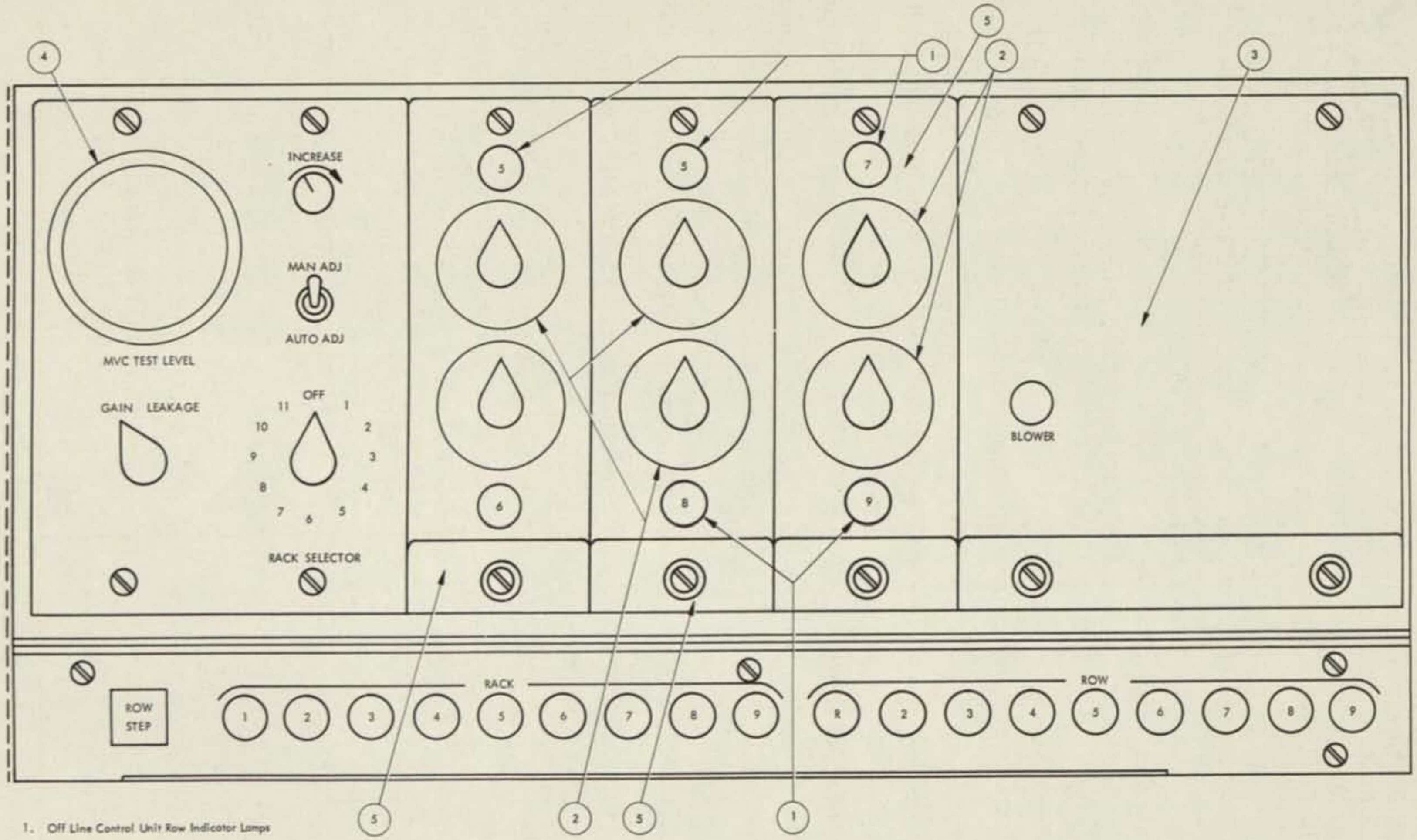
The OLCU control center is located in the S602A enclosure. It occupies the upper four rows (A1, A2, A3, and A4) of that enclosure. The top row (row A1) houses the subassemblies associated with the marginal check voltage (MCV) system. The operator's panel, located in the area normally used for logic-package assemblies in standard MOBIDIC enclosures, is divided into two principal sections. These are the Device and Mode Selection and the Operating and Error Checking sections. The MCV row is provided with a door. This has been done because access to the controls on the MCV subassemblies is not usually required nor is it desirable that such access be easy during normal OLCU operation.

2.6.2.1 Marginal Check Voltage System Subassemblies

The marginal check voltage system subassemblies are similar to corresponding standard MOBIDIC MCV subassemblies. The arrangement of the six MCV subassemblies, in row A1 of the S602A enclosure is shown on Figure 2-3. The various controls, indicators, and associated items used in controlling MCV application are listed on Table 2-3 which also describes the principal function of each item.

2.6.2.2 Device and Mode Selection Section

The upper section of the operator's panel is the device and mode selection section. The arrangement of the controls and indicators are as shown on Figure 2-4. The controls and indicators on the device and mode selection section are listed on Table 2-4. This table also describes the principal function of each item.



1. Off Line Control Unit Row Indicator Lamps
2. Package Selector Switch
3. Stepping Switch Sub-Assembly
4. MCV Control Subassembly
5. Selector Slwtch Subassembly

Figure 2-3. OLCU Marginal Check Voltage Subassemblies

TABLE 2-3. MCV CONTROLS AND INDICATORS

Item	Function
MCV Control Subassembly	
MCV TEST LEVEL Meter	Indicates the MCV level applied to circuits during MCV system operation.
GAIN-LEAKAGE Switch	Used for MCV bias selection. Deterioration in transistor beta is checked when this switch is set at the GAIN position; deterioration in transistor I_{co} is checked when this switch is set at the LEAKAGE position.
Manual Voltage Control	Used to adjust the MCV level when the MAN ADJ-AUTO ADJ switch is set at the MAN ADJ position.
MAN ADJ-AUTO ADJ Switch	Used to select the method of adjusting the MCV level. When this switch is set at the MAN ADJ (manual adjustment) position the MCV level is determined by the setting of the manual voltage control. When this switch is set at the AUTO ADJ (automatic adjustment) position the MCV level is determined by the setting of the level controls in the MCV power supply.
RACK SELECTOR Switch	Use to select the rack to which the MCV is to be applied.
MCV Selector Switch Subassemblies	
Row Indicators	Light to indicate the OLCU row to which the MCV is applied.
Selector Switches	Used to select the logic-package assembly to which the MCV is to be applied.
Stepping Switch Subassembly	
- - -	The stepping switch in this subassembly is used to control automatically the sequential application of the MCV to the logic-package assemblies in a selected row in the OLCU.

TABLE 2-3. MCV CONTROLS AND INDICATORS (Cont.)

Item	Function
BLOWER FAILURE Switch	Lights if a blower failure occurs.
Indicator Panel Subassembly	
Rack and Row Indicators	Light to indicate the rack and row to which the MCV is applied.
ROW STEP Switch	Used to apply the MCV to each row (one at a time) in a selected rack.

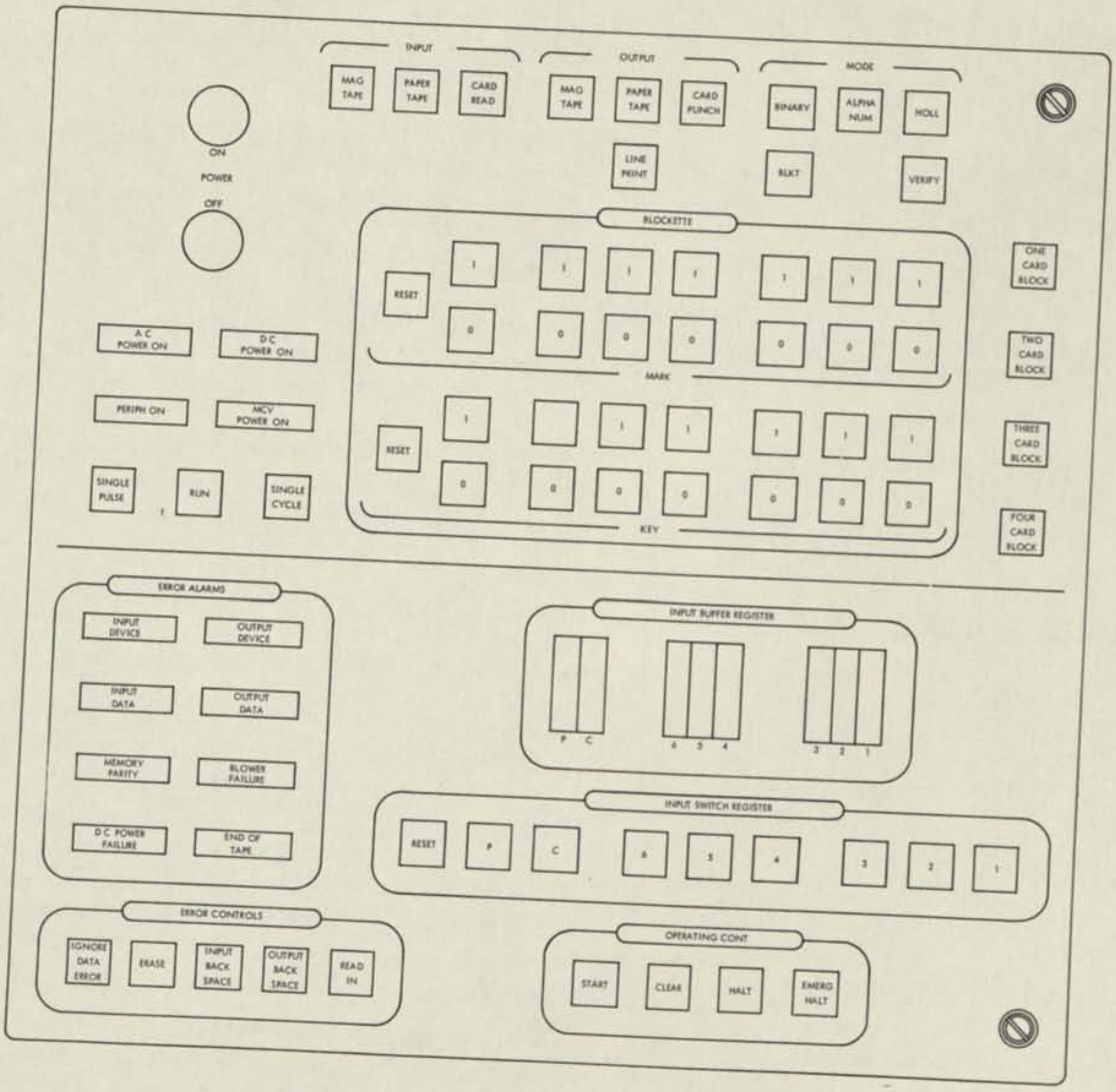


Figure 2-4. OLCU Operator's Panel

TABLE 2-4. DEVICE AND MODE SELECTION CONTROLS AND INDICATORS

Item	Function
Power Controls and Indicators	
Power ON Switch	Used to energize the DC power supplies and apply AC power to the peripheral equipment.
Power OFF Switch	Used to shut off the DC power supplies.
AC POWER ON Indicator	Lights to indicate that AC power is on.
DC POWER ON Indicator	Lights to indicate that DC power is on.
PERIPH ON Indicator	The PERIPH ON (peripheral on) indicator lights to indicate that power has been applied to the peripheral equipment.
MCV POWER ON Indicator	Lights to indicate that MCV power is being applied.
Operating Mode Controls	
SINGLE PULSE Switch	Used to provide the gating levels necessary to gate off the normal pulse distribution system and to provide a single pulse each time it is operated.
RUN Switch	Used to provide the gating levels necessary for normal operation of the OLCU.
SINGLE CYCLE Switch	Used to provide the gating levels necessary for completion of a single input or output cycle.
Input Media Controls	
MAG TAPE Switch	The MAG (magnetic) TAPE switch is used when magnetic tape is to be used for input.
PAPER TAPE Switch	The PAPER TAPE switch is used when paper tape is to be used for input.
CARD READ Switch	The CARD READ (card reader-punch) switch is used when punched cards are to be used for input.

TABLE 2-4. DEVICE AND MODE SELECTION CONTROLS AND INDICATORS (Cont.)

Item	Function
Output Media Controls	
MAG TAPE Switch	The MAG (magnetic) TAPE switch is used when magnetic tape is to be used for output.
PAPER TAPE Switch	The PAPER TAPE switch is used when paper tape is to be used for output.
CARD PUNCH Switch	The CARD PUNCH (card reader - punch) switch is used when punched cards are to be used for output.
LINE PRINT Switch	The LINE PRINT (line printer) switch is used when the line printer is to be used for output.
Mode Selection Controls	
BINARY Switch	Usually used when input data has been prepared in the interpret sign mode.
ALPHANUM Switch	The ALPHANUM (alphanumeric) switch is usually used when input data has been prepared in the non-interpret sign mode.
HOLL Switch	The HOLL (Hollerith) switch is used with magnetic or paper tape input when card output in Hollerith code is desired. It is also used when cards punched in Hollerith code are to be used as input. In this case, however, the output will be in Fieldata code regardless of whether line printer, magnetic tape, or paper tape output is selected.
BLKT Switch	The BLKT (blockette) switch is used to select the blockette mode of operation when using input from either magnetic tape or paper tape.
VERIFY Switch	Used when it is desired to operate the OLCU in the VERIFY mode. Use of this mode enables checking of information on the input media. No output device is selected when operating in the VERIFY mode.

TABLE 2-4. DEVICE AND MODE SELECTION CONTROLS AND INDICATORS (Cont.)

Item	Function
Blockette Mode Controls	
MARK Switches (7)	Used to set up the MARK character when operating in the blockette mode. Only the MARK character is used with paper tape input. With magnetic tape input both MARK and KEY characters are used.
MARK RESET Switch	Used to reset the MARK switches to the OFF state.
KEY Switches (7)	Used to set up the KEY character when operating in the blockette mode with magnetic tape input.
KEY RESET Switch	Used to reset the KEY switches to the OFF state.
Card Input Controls	
ONE CARD BLOCK Switch	Used with punched card input when one card is to be written per block of output.
TWO CARD BLOCK Switch	Used with punched card input when two cards are to be written per block of output.
THREE CARD BLOCK Switch	Used with punched card input when three cards are to be written per block of output.
FOUR CARD BLOCK Switch	Used with punched card input when four cards are to be written per block of output. Four-card output can be employed only in off-line control systems which use a card reader-punch which does not have capability for performing a checking operation.

2.6.2.3 Operating and Error Checking Section

The lower section of the operator's panel is the operating and error checking section of that panel. The arrangement of the controls and indicators is shown on Figure 2-4. The controls and indicators on the operating and error checking section are listed on Table 2-5. This table also describes the principal function of each item.

2.6.3 Memory

The OLCU uses a coincident-current, high-speed, magnetic core memory. The memory has a maximum storage capacity of 336 sixteen-bit words. The memory consists of 8 memory planes, each approximately 2 inches by 3 inches by 1/2 inches in volume. Each of the memory planes is of the double-matted, magnetic-core type. The magnetic cores in each plane are arranged in 2 mats, each having 336 magnetic cores arranged in a 14 by 24 matrix.

The memory capacity is such that it allows storage of up to either 336 Hollerith-coded characters (16 bits each) or 672 Fieldata-coded characters (8 bits each). This capacity is sufficient for storing the information contained on four punched cards. To simplify memory operation, particularly when operating with card equipment, the memory is considered as being divided into four quadrants, each having a matrix dimension of 7 by 12 by 16.

The OLCU memory is essentially similar in operation to that of MOBIDIC B. The principal difference is in the drive line selection method. The drive line selection method used for the OLCU memory is described in paragraph 2.7.1.

2.6.4 OLCU Registers

The OLCU contains three closely integrated registers. These are the input buffer, memory buffer, and output buffer registers (IBR, MBR, and OBR, respectively). Figure 2-5 is a block diagram indiating their interconnections.

2.6.4.1 Input Buffer Register

The input buffer register (IBR) is an eight-bit register. This register receives data from the selected input device and, either directly or indirectly,

TABLE 2-5. OPERATING AND ERROR CHECKING CONTROLS AND INDICATORS

Item	Function
Error Alarms	
INPUT DEVICE Indicator	Lights to indicate that the selected input device is inoperable. For example, this indicator lights if a fuse is blown, or the tape broken.
OUTPUT DEVICE Indicator	Lights to indicate that the selected output device is inoperable.
INPUT DATA Indicator	Lights if an error has occurred in transferring information from the selected input device to the OLCU.
OUTPUT DATA Indicator	Lights if an error has occurred in transferring information from the OLCU to the selected output device.
MEMORY PARITY Indicator	Lights if a memory parity error occurs during OLCU operation.
BLOWER FAILURE Indicator	Lights if a blower failure occurs.
DC POWER FAILURE Indicator	Lights if there is a failure in the DC power supply.
END OF TAPE Indicator	Lights if the end of tape area on magnetic tape has been reached.
Error Controls	
IGNORE DATA ERROR Switch	When this switch is operated the OLCU will continue to operate despite existence of data error conditions.
ERASE Switch	Used in correcting errors when using magnetic tape.
INPUT BACK SPACE Switch	Used to backspace the input tape during magnetic tape correction procedures.
OUTPUT BACK SPACE Switch	Used to backspace the output tape during magnetic tape correction procedures.

TABLE 2-5. OPERATING AND ERROR CHECKING CONTROLS AND INDICATORS (Cont.)

Item	Function
READ IN Switch	Used to transfer information from the input switch register (ISR) to the input buffer register (IBR).
INPUT BUFFER REGISTER Indicator	These indicators display the contents of the input buffer register.
INPUT SWITCH REGISTER Switches	Used to insert information in the input buffer register when correcting errors on paper tape or when manually entering information in the OLCU.
Operating Controls	
START Switch	Used to start OLCU operation.
CLEAR Switch	Used to clear all OLCU registers and flip-flops. Operation of this switch does not, however, clear the OLCU memory.
HALT Switch	Used to stop OLCU operation at the end of the next output cycle following its use.
EMERG HALT Switch	The EMERG HALT (emergency halt) switch is used to stop the OLCU operation at the end of the next timing pulse following its use.

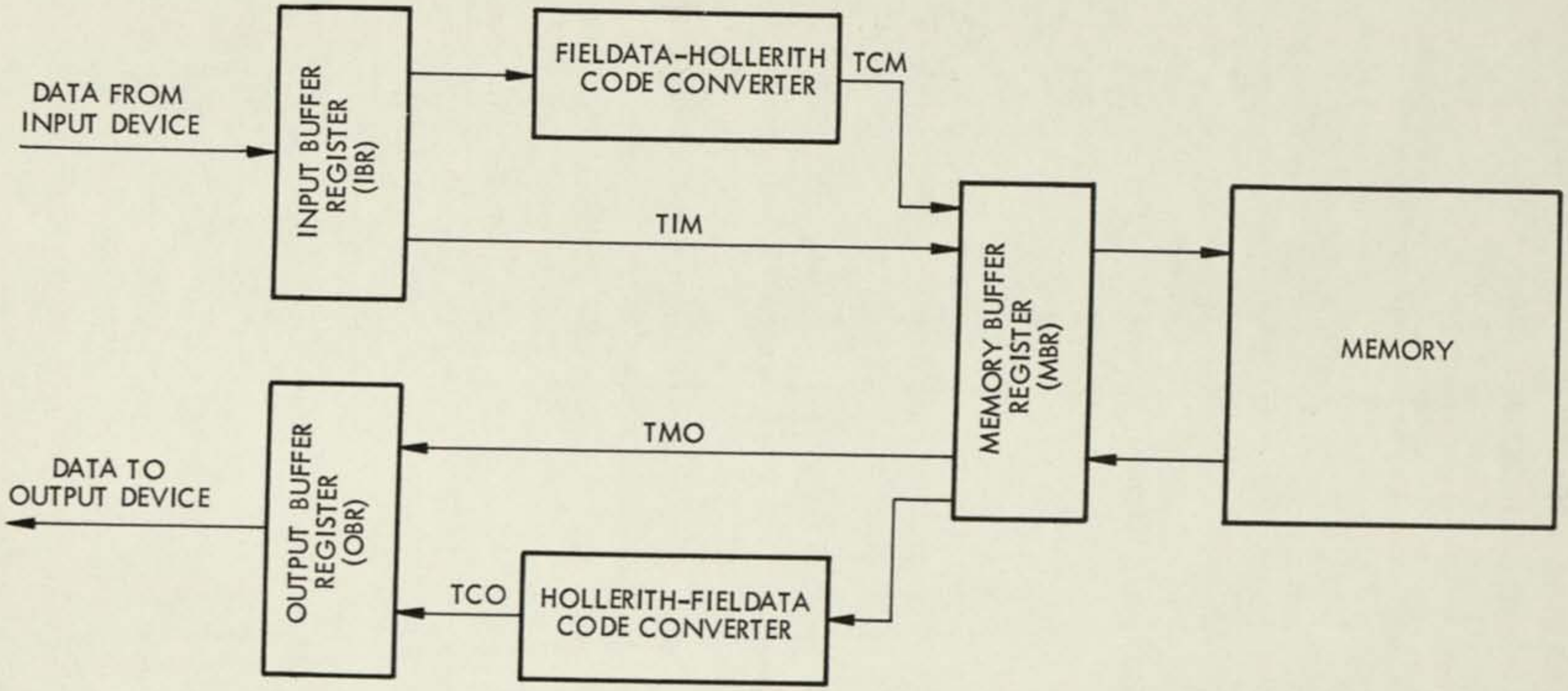

DATA FROM INPUT DEVICE
INPUT BUFFER REGISTER (IBR)
FIELDATA-HOLLERITH CODE CONVERTER
TCM
TIM
MEMORY BUFFER REGISTER (MBR)
MEMORY
TMO
DATA TO OUTPUT DEVICE
OUTPUT BUFFER REGISTER (OBR)
TCO
HOLLERITH-FIELDATA CODE CONVERTER

Figure 2-5. OLCU Buffer Register Interconnections, Block Diagram

transfers that data to the memory buffer register. Data in the IBR is checked for presence of any of the following characters or indications:

LDEL	Delete code
LTPF	Paper tape blank
LSTP	Stop code
LEOF	End-of-file mark
SLCB	Start-of-control-block mark
LBLS	Start-of-block mark
IPAR	IBR even parity
BKT	Blockette mark
KEY	Blockette key

Data received correctly in IBR is transferred directly to the memory buffer register unless a Fieldata to Hollerith code conversion is required. In such cases, the transfer takes place through the Fieldata-Hollerith code converter. This converter is described in 2.6.5.

Even parity is employed for paper tape input, odd parity is used for all other input media. Input data from paper tape is checked for even parity in the IBR. The parity is changed, however, to odd when the transfer of data from the IBR to the MBR takes place. The states of bits 6 and 7 of the IBR are used to distinguish between control and data characters during paper tape input. If both bits 6 and 7 are in the same state, a control character is indicated; if bits 6 and 7 are not in the same state, a data character is indicated. The state of bit 7 only is used in the IBR to distinguish between control and data characters during input from any media other than paper tape. Decoding of control or data characters is always carried out in the IBR. The method of decoding depends upon the input media used. Characters are always stored in memory in either magnetic tape or Hollerith format. Any required format changes are made during the transfer from the IBR to the MBR.

The design of the IBR is such that its contents can be shifted one bit at a time. This shift affects only the data bits. This capability permits acceptance of alphanumeric input from cards which are to be interpreted in either column format (column cards) or row format (row cards). Information from cards to be interpreted in row format is always stored in memory in the form in which it is received from the card reader-punch buffer. Information from cards to be interpreted in column format (either Fieldata or Hollerith coded) is shifted to the MBR, one bit at a time, as explained in 2.6.6.

2.6.4.2 Memory Buffer Register

The memory buffer register (MBR) is a 16 bit register. This register is capable of two-way operation. During the input cycle it accepts data from the IBR, either directly or through the Fieldata to Hollerith code converter, and sends it to memory for storage. During the output cycle it receives data from the memory and sends it, either directly or through the Hollerith to Fieldata code converter, to the output buffer register (OBR) or for transmission to the selected output device.

During the input cycle data entering the MBR directly from the IBR is received 8 bits (one Fieldata character) in parallel. After the first character is received the contents of the MBR are shifted and the second character is accepted, thus filling the MBR. Data entering the MBR from the IBR through the code converter is received 16 bits in parallel. When the MBR is filled its contents are sent to the memory for storage in the memory location specified by the state of the memory address counter (MAC).

During the output cycle data entering the MBR is received 16 bits in parallel from the memory location specified by the state of the MAC. Parity is checked for each half of the contents of the MBR. If a parity error is detected the MEMORY PARITY indicator on the operator's panel will light. If paper tape output has been selected the conversion from magnetic tape format ("normal" format) to paper tape format is carried out when the data is transferred from the MBR to the OBR. If data is transferred directly from the MBR to the OBR, the transfer is carried out 8 bits in parallel. The first character is transmitted to OBR, the contents of the MBR are shifted, and the second character is transmitted to OBR. When data is transferred from the MBR to the OBR through the Hollerith to Fieldata code converter the input to the code converter is 16 parallel bits; output from the code converter to the OBR is 8 parallel bits.

End of block marks are decoded in the MBR.

2.6.4.3 Output Buffer Register

The output buffer register (OBR) is an 8-bit register. This register receives data, either directly or indirectly, from the MBR and sends that data to the selected output device. Data in the OBR is checked for presence of any of the following characters or indications:

EBM	End-of-file mark,end-of-control-block mark, or end-of-block mark
SBM	Start-of-block mark or start-of-control-block mark

Data from the MBR is transferred directly to the OBR unless a Hollerith to Fieldata code conversion is required. In such a case, the transfer takes place through the Hollerith to Fieldata code converter. This converter is described in 2.6.5.

The design of the OBR is such that itscontents can be shifted left one bit at a time. This shift affects only the six data bits. This capability permits cards to be punched in either column or row format. The capability of shifting one information bit at a time from the MBR to the OBR is required in carrying out column to row conversions. Such conversions are necessary when punching a card which is to be interpreted as having a column format. The conversion process is described in 2.6.6.

2.6.5 Code Converters

The OLCU is able to perform Hollerith-Fieldata code conversions. This capability has been included because, although MOBIDIC has been built to process data prepared in Fieldata code, information on cards is commonly in Hollerith code. Two code converters, one for Fieldata to Hollerith (F-H) conversions and the other for Hollerith to Fieldata (H-F) conversions, are provided. Their location is shown, in block diagram form, on Figure 2-5.

During input operation data enters the IBR first and is then transferred into the MBR. Although data enters the F-H code converter whether or not conversion is required, data is transferred from the converter into the MBR only if conversion is required. During a conversion operation the converted data is transferred from the F-H code converter into the MBR in response to a transfer from converter to memory (TCM) signal. There is no transfer of data directly from the IBR to the MBR. If conversion is not required, a transfer into memory (TIM) signal transfers data from the IBR into the MBR. No transfer of data from the F-H code converter into the MBR takes place.

During output operation data enters the MBR from the memory and is then transferred into the OBR. Although data enters the H-F code converter whether or not conversion is required, data is transferred from the converter into the OBR only if conversion is required. During a conversion operation the converted data is transferred from the H-F code converter into the OBR in response to a transfer from converter to OBR (TCO signal). There is no transfer of data directly from the MBR to the OBR. If conversion is not required, a transfer from memory to OBR (TMO) signal transfers data from the MBR into the OBR. No transfer of data from the H-F code converter into the OBR takes place.

2.6.5.1 Fieldata to Hollerith Code Converter

The Fieldata to Hollerith (F-H) code converter is used when Hollerith-coded punched card output is desired and the input media is either magnetic or paper tape. The media and mode selection controls on the operator's panel are set as follows for such a conversion: input media controls - MAG TAPE or PAPER TAPE switch ON, output media controls - CARD PUNCH switch ON, mode selection controls - HOLL switch ON.

During a conversion operation each Fieldata character received in the IBR during the input cycle is transferred into the F-H code converter. There a decoded and an appropriate signal is generated. This signal is then used to place binary ONES in appropriate bit positions as necessary to produce a Hollerith-coded character equivalent to the Fieldata-coded character received in the IBR from the selected input device. The character, in Hollerith-code, is then transferred from the F-H code converter into the MBR in response to a transfer from converter to memory (TCM) signal.

The assigned bit positions in the MBR for Hollerith-coded characters are as follows:

	Stage															
MBR	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Hollerith bit	9	8	7	6	5	4	*	L	3	2	1	10	11	12	*	H

Note: L = low parity, H = high parity, * = a ONE is always stored.

Two parity bits are associated with each Hollerith-coded character, one for the high parity check, the other for the low parity check. The high parity bit is associated with Hollerith bits 12, 11, 10, 1, 2, 3; the low parity bit is associated with Hollerith bits 4, 5, 6, 7, 8, 9. The parity bits are generated as necessary by the F-H code converter when the Fieldata-coded character is decoded. The parity bits are generated on the basis of the input to the code converter, rather than on the basis of the code converter output. This provides a check on code converter operation. The actual parity check is made when reading information from the MBR.

2.6.5.2 Hollerith to Fieldata Code Converter

The Hollerith to Fieldata (H-F) code converter is used when Fieldata-coded output is desired and the input media is Hollerith-coded punched cards. The media and mode selection controls on the operator's panel are set as follows for such a conversion: input media controls - CARD READ switch ON, output media controls - MAG TAPE or PAPER TAPE or LINE PRINT switch ON, mode selection controls - HOLL switch ON

During a conversion operation each Hollerith character received in the MBR during the output cycle is transferred into the H-F code converter. There the Hollerith character bit configuration is decoded to generate ONES in appropriate bit positions as necessary to produce a Fieldata data character equivalent to the Hollerith character sent from the MBR to the H-F code converter. Since the seventh bit of a Fieldata data character is always a ONE and since the input to the code converter can only be data characters, the seventh bit of the Fieldata character formed in the H-F code converter is always a ONE. The eighth (parity) bit of the Fieldata character formed in the H-F code converter is a ONE or a ZERO as necessary to produce correct parity. The parity bit is set, when required, before the Fieldata character is sent to the OBR. As in the case of the F-H code converter, the proper parity bit is generated by the code converter to provide a check on the conversion. The Fieldata character is transferred from the H-F code converter in response to a transfer out of converter (TOC) signal.

2.6.5.3 Special Characters Code

Although there is only one Hollerith code, there are several possible sets of Hollerith special character (characters neither alphabetic nor numerical) representations available. One particular set of representations has been selected for use in the OLCU. For descriptive purposes this set is identified as the M (for MOBIDIC) set. Correspondence in the representation of special characters in the M set and in Fieldata code is shown on Table 2-6. The representations in the M set have been selected to obtain the special characters available on the Flexowriter and line printer used in MOBIDIC 7A. Characters identified on Table 2-6 as either fixed conversions or as variable wired conversions are wired into the code converters. The logical and packaging design is such that any character identified as variable unwired can be substituted for any character identified as variable wired. The substitution, which is made by changing jumper wires on the appropriate logic-package assemblies, permits the representation of certain of the special characters to be varied to suit special requirements. The characters identified as fixed conversions cannot be changed. The code conversions have been designed to be both unique and bilateral. They are unique in that there is one representation for each character; they are bilateral in that a conversion from Hollerith to Fieldata and then from Fieldata to Hollerith will always result in production of the same code combination as was used as input in the Hollerith to Fieldata conversion.

An error-detection network is provided in each converter. Detection of any Fieldata code for which there is not an equivalent Hollerith code causes the INPUT DATA error indicator on the operator's panel to light.

TABLE 2-6. SPECIAL CHARACTERS, HOLLERITH CODE - FIELDATA CODE

Fieldata Representation	Code			Conversion	
	Hollerith	Fieldata Standard Code	Fixed	Variable Wired	Variable Unwired
+	100 000 000 000	01 100 010		*	
>	101 000 000 000	11 100 101		*	
.	100 001 000 010	11 111 101	*		
)	100 000 100 010	11 100 000		*	
-	010 000 000 000	01 100 001	*		
<	011 000 000 000	11 100 011		*	
$	010 001 000 010	01 100 111	*		
*	010 000 100 010	01 101 000	*		
,	001 001 000 010	01 101 110	*		
(	001 000 100 010	11 101 001		*	
/	001 100 000 000	01 111 100	*		
=	000 001 000 010	01 100 100		*	
;	000 000 100 010	11 111 011		*	
Master Space	000 000 000 000	01 000 000	*		
Back Space	111 111 111 111	01 111 111	*		
"	-	11 101 010			*
:	-	01 101 011			*
?	-	11 101 100			*
!	-	01 101 101			*

2.6.6 Row-Column Conversions

Provision for automatic row to column and for column to row conversions has been included in the design of the OLCU. This has been done since, although the card reader-punch used with the off-line control system reads or punches cards one row at a time, information is often stored or punched on cards in column format, one or two characters to a column. Any required conversions are performed automatically by coordinated combinations of register shifts and memory operation. The shifting and memory operations are detailed in paragraph 2.6.6.1. The sequence of operations that take place during a row to column or column to row conversion is indicated in paragraph 2.6.6.2. Paragraph 2.6.6.3 describes the memory address counter, used in column-row conversions and in memory operations.

2.6.6.1 Shifting Operations

Shifting operations are required during row to column or column to row conversions. The shifting operations involve the input, output, and memory buffer registers (IBR, OBR, and MBR). The shifts occur in response to the following-listed signals.

SHMBR	-	Shift memory buffer register
SLIBR	-	Shift left input buffer register
SLOBR	-	Shift left output buffer register

The design is such that the MBR is capable of both one-character and one-bit shifts. During one-character shifts the contents of the lower half (low-order bit positions 1 through 8) of the MBR are shifted into the upper half (high-order bit positions 9 through 16) of the MBR and, simultaneously, the contents of the upper half of the MBR are shifted into the lower half of the MBR. The one-bit shift capability is essential to row-column conversions. During such conversions the six data bits are shifted; the proper control bit is inserted; and the proper parity bit is generated and inserted.

Mechanization is such that the same signal, the SHMBR signal, is used to perform either the one-bit or the one-character shift. A single SHMBR signal causes a one-character shift during which the contents of the upper and lower halves of the MBR are exchanged. Two successive SHMBR signals cause first a one-character shift and then a one-bit shift. The original contents of each MBR bit position and the contents of that position after a one-character and after a one-bit shift are indicated on Figure 2-6.

The SHMBR signal always causes the proper parity bit to be placed in MBR8. The parity bit depends upon the bit being transferred into MBR1 and upon the bits being transferred from MBR14 and MBR16. It should be noted that if MBR6 is placed in MBR1, as shown by the first shift, the second shift results in a one-bit shift of the twelve data bits originally (i.e., before the first and second shifts) in the MBR. Shifting of MBR14 into OBR1, which occurs during an output operation, must be carried out during the first (one-character) shift. Shifting of IBR6 into MBR1, which occurs during an input operation, must be carried out during the second (one-bit) shift. The bit stored in MBR1 (Y on Figure 2-6) is contingent upon the results of other shifts occurring at the time the MBR is shifted as follows:

SHMBR.SLOBR.SLIBR	:	IBR6 =) MBR1
SHMBR.SLOBR.SLIBR-	:	MBR6 =) MBR1
SHMBR.SLOBR-.SLIBR	:	OBR1 =) MBR1
SHMBR.SLOBR-.SLIBR-	:	∅ =) MBR1

It should be noted that the logical conditions noted above are given in symbolic "machine-language" form. The various conventions used in "machine-language" coding used here and in other paragraphs in this section are listed on Table 2-7.

2.6.6.2 Operating Sequence

Information is read or punched on cards one row at a time by the card reader-punch used in MOBIDIC. Provision has been made, however, for automatic row to column or column to row conversion by the OLCU. This capability permits automatic conversion between Fieldata and Hollerith codes and enables

MBR BIT POSITION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CONTENTS BEFORE SHIFT	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
CONTENTS AFTER FIRST SHIFT	X	C9	C10	C11	C12	C13	C15	P1	C1	C2	C3	C4	C5	C6	C7	C8
CONTENTS AFTER SECOND SHIFT	Y	C1	C2	C3	C4	C5	C7	P2	X	C9	C10	C11	C12	C13	C15	P1
P1, P2 = PARITY BITS. FORMED BY SHIFT MBR SIGNAL.																

Figure 2-6. MEMORY BUFFER BIT POSITION CONTENTS, BEFORE AND AFTER SHIFTING.

Figure 2-6. Memory Buffer Bit Position Contents, Before and After Shifting

TABLE 2-7. MOBIDIC SYMBOLIC "MACHINE-LANGUAGE" CONVENTIONS

Symbol or Abbreviation	Definition
Letter abbreviations	Letter abbreviations are used to designate registers, counter, control pulses, control flip-flops, etc. For example, SQ designates the sequence counter.
Subscripts	Subscripts added to register or bus abbreviations are used to specify a particular register stage or bus line. For example, IBR_3 represents stage 3 of the input buffer register.
Superscripts	Alphanumerical superscripts are used to represent one particular unit out of several which have the same abbreviations. For example, CNV^2 is the abbreviation for converter number two.
*	An asterisk is used to designate pulsed control lines as opposed to level control lines. For example, A1ST* is the abbreviation for a control line which, when pulsed, adds ONE to the contents of the ST counter.
-	In machine language coding, a minus sign added to an abbreviation indicates logical negation (prime). For example, INPUT- means "not input".
.	In machine language coding, the period placed between terms of a logical statement is equivalent to the logical AND. For example, IHSQ-. t* represents the statement "Not IHSQ and a t-pulse".

TABLE 2-7. MOBIDIC SYMBOLIC "MACHINE-LANGUAGE" CONVENTIONS (Cont.)

Symbol or Abbreviation	Definition
,	In machine language coding, the comma placed within an abbreviation indicates that the letters or numbers following it represent a superscript. For example, CNV, N represents CNV^N.
$	In machine language coding, the dollar sign represents summation, and is the equivalent of the Greek "sigma". ($\sum$). For example, $SQ represents the summation of the states of the SQ counter.
/	In machine language coding, the slash sign placed within an abbreviation indicates that the letters or numbers following it represent a subscript. For example, BUS/N represents BUS_N.
=	The equal sign is used for several purposes. It can denote equality. When with the abbreviation for a register or a counter, the term to the right of the equal sign specifies the internal state of the register or counter, for example, SQ = N means that the SQ carries an internal count of N. The equal sign is also used in specifying the state of a control line. In such a case, the equal sign is used in conjunction with a ONE or a ZERO, indicating whether the specified control level is high or low, respectively. For example, the expression 1 = CLRWAR states that the CLRWAR control line is at the high level.

TABLE 2-7. MOBIDIC SYMBOLIC "MACHINE-LANGUAGE" CONVENTIONS (Cont.)

Symbol or Abbreviation	Definition
⊕	This symbol is used to represent the EXCLUSIVE OR logical function. For example, (A/SN⊕B/SN) represents the logical conditions (A_{SN} $B\text{-}_{SN}$ + $A\text{-}_{SN}$ B_{SN}) where the plus sign is used in the Boolean sense of OR.
=) =>	The =) symbol is used in machine language coding to designate transfer operations. It is equivalent to the symbol =>. In addition to denoting simple information transfers, the transfer symbol is also used to represent counting operations, setting, clearing or complementing of Flip-flops or registers, and for pulse gating operations. The various uses are illustrated in the following examples. 1. Information Transfers - the operation BUS/N =) IBR/N indicates that the information on BUS N is transferred into stage N of the IBR. 2. Counting Operations - the operation MAC + 1 =) MAC indicates that the quantity ONE is added to the contents of the memory address counter. 3. Setting of Flip-flops or Registers - the operation 1 =) GO indicates that the GO Flip-flop is set to the ONE state. 4. Clearing of Flip-flops or Registers - the operation ∅ =) GO indicates that the GO Flip-flop is reset, i. e., cleared, to the ZERO state.

TABLE 2-7. MOBIDIC SYMBOLIC "MACHINE-LANGUAGE" CONVENTIONS (Cont.)

Symbol or Abbreviation	Definition
	5. Complementing of Flip-flops or Registers- the operation SEC- =) SEC indicates that the SEC flip-flop is complemented. (ONEs complement). 6. Pulse Gating - the operation 1 =) A1SQ* indicates that a pulse is applied to the A1SQ* control line.

handling of cards to be interpreted in column format without requiring an additional programmed conversion for use during on-line card reader-punch operation. When conversion is required, row to column conversion is always done during the input portion of the OLCU cycle while column to row conversion is always done during the output portion of the OLCU cycle.

The card is always treated as if it has 84 columns for storage of information. This has been done to achieve compatibility with standard MOBIDIC words which have either six or seven characters each, depending upon whether the word is written in the interpret sign or in the non-interpret sign mode. The first four characters sent or received by the card reader punch, however, are padding characters and as such they never appear on the card itself.

Simplified flow charts, one for a row to column conversion (input operation) and the other for a column to row conversion (output operation), are shown on Figures 2-7 and 2-8, respectively. To simplify understanding of OLCU conversions, only the general sequence of operations performed during a conversion are shown. For the mechanization of row to column or column to row conversion, both with and without card checking, see 2.6.7.

When a card is to be read, the bits from a row must be placed in successive memory locations. Accordingly, each bit, by a suitable combination of register shifts and memory operations, is placed in a memory location which corresponds to the card column from which it came. The first 14 characters to enter memory contain the 84 bits (including the 4 padding characters) from the first row on the card. These bits are placed in the first bit positions of the first 84 successive memory locations. The second 14 characters to enter memory contain the 84 bits from the second row. As these bits enter memory they are placed in the first bit positions of the first 84 successive memory locations while simultaneously the 84 bits from the first row are shifted into the second bit positions in these same 84 memory locations. Operation continues in this manner until all rows on the card have been read. After the last row has been read into memory, each of the first 84 memory locations will contain the bits which were on a corresponding column on the card. A separate set of 84 successive memory locations is used for storage

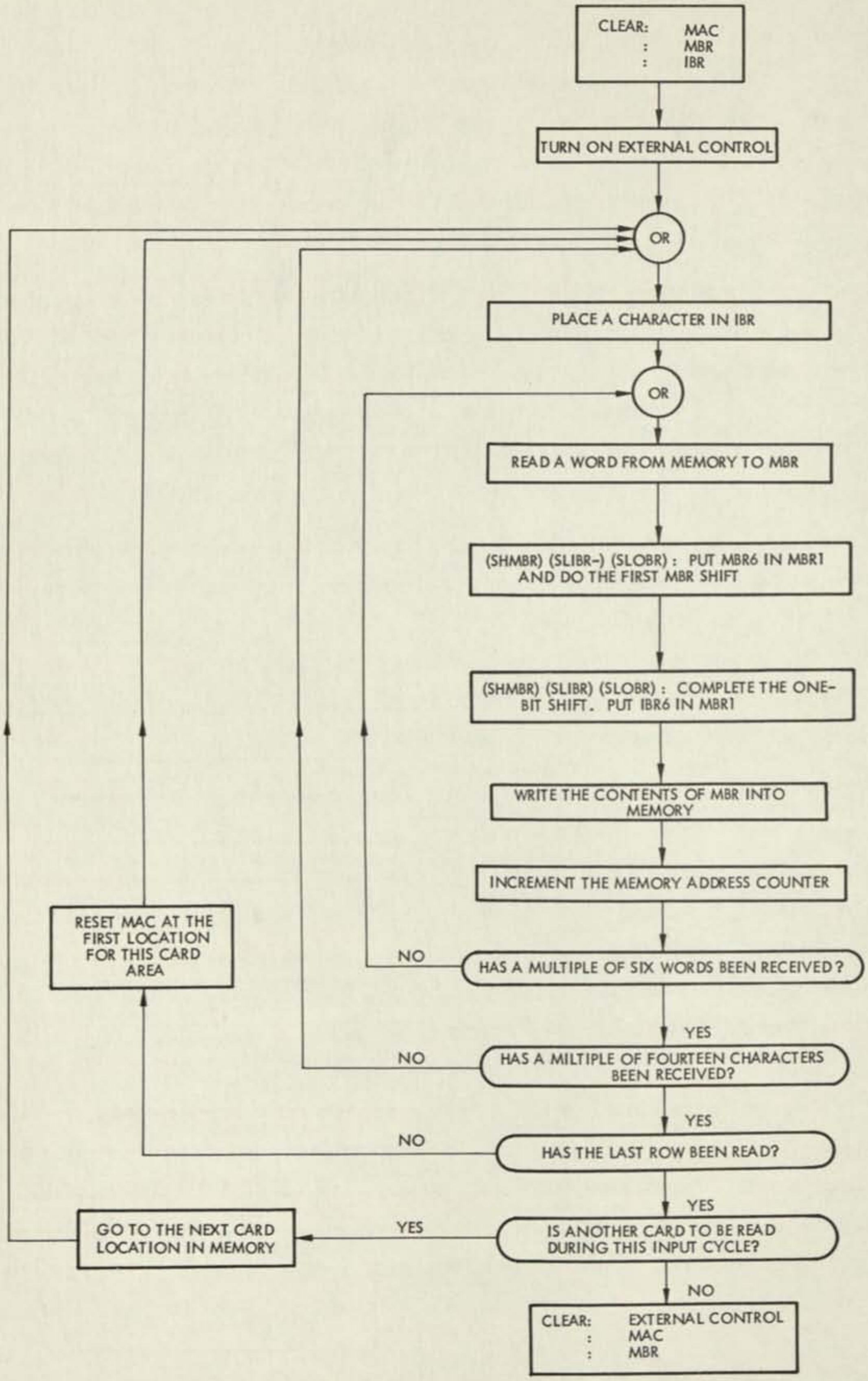

CLEAR: MAC
: MBR
: IBR
TURN ON EXTERNAL CONTROL
OR
PLACE A CHARACTER IN IBR
OR
READ A WORD FROM MEMORY TO MBR
(SHMBR) (SLIBR–) (SLOBR) : PUT MBR6 IN MBR1 AND DO THE FIRST MBR SHIFT
(SHMBR) (SLIBR) (SLOBR) : COMPLETE THE ONE-BIT SHIFT. PUT IBR6 IN MBR1
WRITE THE CONTENTS OF MBR INTO MEMORY
INCREMENT THE MEMORY ADDRESS COUNTER
NO
HAS A MULTIPLE OF SIX WORDS BEEN RECEIVED?
YES
NO
HAS A MILTIPLE OF FOURTEEN CHARACTERS BEEN RECEIVED?
YES
RESET MAC AT THE FIRST LOCATION FOR THIS CARD AREA
NO
HAS THE LAST ROW BEEN READ?
YES
GO TO THE NEXT CARD LOCATION IN MEMORY
YES
IS ANOTHER CARD TO BE READ DURING THIS INPUT CYCLE?
NO
CLEAR: EXTERNAL CONTROL
: MAC
: MBR

Figure 2-7. Flow Chart, Row to Column Conversion

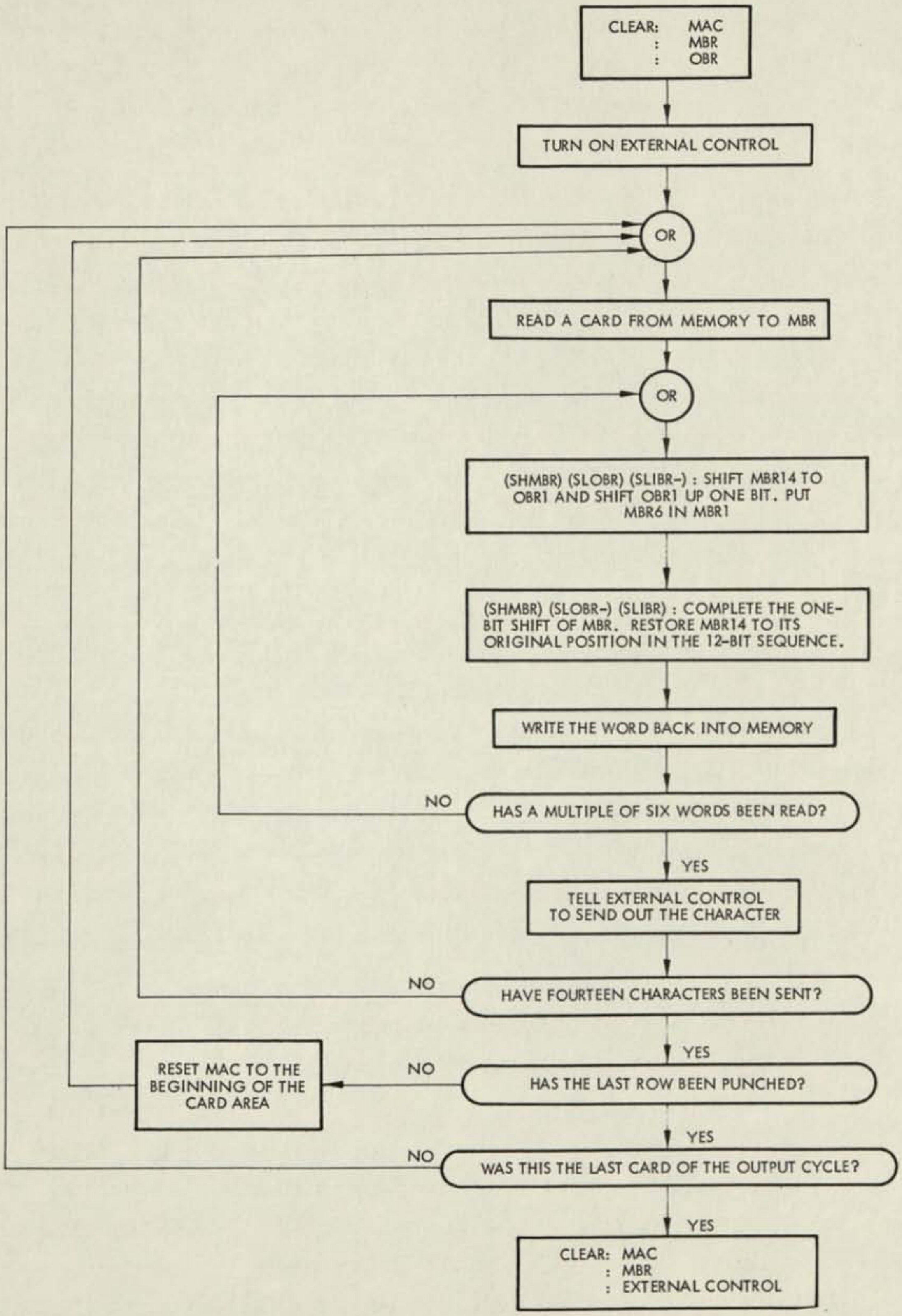

CLEAR: MAC
: MBR
: OBR
TURN ON EXTERNAL CONTROL
OR
READ A CARD FROM MEMORY TO MBR
OR
(SHMBR) (SLOBR) (SLIBR–) : SHIFT MBR14 TO OBR1 AND SHIFT OBR1 UP ONE BIT. PUT MBR6 IN MBR1
(SHMBR) (SLOBR–) (SLIBR) : COMPLETE THE ONE-BIT SHIFT OF MBR. RESTORE MBR14 TO ITS ORIGINAL POSITION IN THE 12-BIT SEQUENCE.
WRITE THE WORD BACK INTO MEMORY
NO
HAS A MULTIPLE OF SIX WORDS BEEN READ?
YES
TELL EXTERNAL CONTROL TO SEND OUT THE CHARACTER
NO
HAVE FOURTEEN CHARACTERS BEEN SENT?
YES
RESET MAC TO THE BEGINNING OF THE CARD AREA
NO
HAS THE LAST ROW BEEN PUNCHED?
YES
NO
WAS THIS THE LAST CARD OF THE OUTPUT CYCLE?
YES
CLEAR: MAC
: MBR
: EXTERNAL CONTROL

Figure 2-8. Flow Chart, Column to Row Conversion

of information from each card read during an input operation. A maximum of four cards (three, with checking,) can be read during a single input operation. The order of reading and storing bits in memory is such that Hollerith to Fieldata conversion during output is facilitated.

The column to row conversion during output operation is essentially similar to the row to column conversion. One bit at a time is taken from each of the first 84 successive memory locations and shifted from the MBR into the OBR. When six bits have been shifted into OBR they are sent to the card reader-punch buffer. After the first bit in each of the 84 successive memory locations has been read out, the second bit in each of the same memory locations is read out. This process continues until all bits have been read out, thus completing output for one card. If more than one card is to be punched, memory readout from successive sets of memory locations continues until the specified number of cards has been punched. A maximum of four cards (three, with checking) can be punched during a single output cycle.

2.6.6.3 Memory Address Counter

Consideration of the requirements of row to column and column to row conversions indicated that the OLCU must take appropriate action when a sixth shift has been made (by which one character is transferrred or formed) and when all 84 bits representing one card row of information have been processed. It was decided that the memory address counter (MAC) would be used for counting shifts as well as for normal memory addressing.

The MAC is made up of four smaller counters, these are the XA, YA, YB, and XB counters. The smaller counters count modulo six, modulo seven, modulo two, and modulo four, respectively. The XA counter is used to count the six shifts required for each character transfer. The YA and YB counters are used in conjunction to count the 14 characters in each row. The XB counter is used to indicate which of the four groups of 84 memory locations each is being used during a particular row to column or column to row conversion. The X (XA and XB) and Y (YA and YB) counters are used in X- and Y-drive line selection during memory operation. The number of X and Y switches required for selecting a location in memory has been made equal to the number of states each part of the MAC can count. Thus, each of the 24 X-drive lines is selected by a 4 (XB) by 6 (XA) switch matrix and each of the 14 Y-drive lines is selected by a 7 (YA) by 2 (YB) switch matrix. The decoding networks on the MAC are, therefore, minimized.

The XA and YA counters have each been designed as shift counters, rather than as binary counters. This has minimized the number of transistors used. The XA counter counts in the following sequence:

State	XA3	XA2	XA1
Ø	Ø	Ø	Ø
1	Ø	Ø	1
2	Ø	1	1
3	1	1	1
4	1	1	Ø
5	1	Ø	Ø
Ø	Ø	Ø	Ø

The YA counter counts as follows:

State	YA3	YA2	YA1
Ø	Ø	Ø	Ø
1	Ø	Ø	1
2	Ø	1	Ø
3	1	Ø	1
4	Ø	1	1
5	1	1	Ø
6	1	Ø	Ø
Ø	Ø	Ø	Ø

These counters shift left one bit on every pulse to be counted. The rule for setting the lowest order bit is as follows.

For XA counter:

XA3- =) XA1

For the YA counter:

(YA3)(YA1) + (YA3-)(YA1-) : 1 =) YA1

((YA3)(YA1) + (YA3-)(YA1-))- : ∅ =) YA1

2.6.7 Control

It was evident early in the OLCU design phase that the control problem would be a complex one since the OLCU was to be a fixed-program device capable of operating with a wide variety of in-out devices and formats. Analysis indicated that the various control functions could be separated into three major groups: external control, internal control, and program control. This approach offers the advantages of optimum control simplicity and flexibility, reduction in equipment requirements, ease of comprehension and, thus, ease of trouble-shooting and maintenance. The three control areas are described briefly in this paragraph and more extensively in paragraphs 2.6.7.1 through 2.6.7.3. External control regulates transfers between the input and output buffer registers (IBR and OBR) on one hand and the in-out devices on the other hand. It provides all synchronizing and control signals required to operate the in-out devices in the off-line control system. Internal control regulates memory, register, and memory address counter operation. It contains all of the logic required for such data handling functions as row-to-column and column-to-row conversions, code conversions, and card checking. Program control both initiates and regulates the proper sequence of internal control operations. Program control accepts inputs from the operator's panel as well as alarm signal inputs.

2.6.7.1 External Control

External Control, which may be defined as device sequencing and timing control, provides all synchronizing and control signals necessary to operate the in-out devices and, when necessary, changes the media format into a standard format which can be utilized by internal control. The external control boundaries are well defined: devices on one side, internal control on the other side. External control is concerned with only two registers in the OLCU. These are the input and output buffer registers (IBR and OBR). External control performs all of the transformations necessary, including all required timing, for buffering between internal control and the in-out devices. Communication between external and internal control takes place over two sets of lines. The communication is carried out in a manner similar to the ready-strobe system used with in-out devices. During input, characters are transferred from the selected in-out device to the IBR and a signal is sent to internal control to indicate the fact that data is present in the IBR ready for processing in the OLCU. After this data has been processed, a suitable signal from internal control is sent to the external control to indicate that the data has been processed. Operation during an output operation is essentially similar, with data passing from the OLCU to the selected output device.

Analysis of the external control functions indicated that there were approximately 50 distinct control states. Accordingly, a six-bit, 64-state binary counter has been provided to define the necessary time sequences. This is the external control sequence counter (SQ). Operation of magnetic tape units required that a serial timing operation be carried out before data could be written or read from magnetic tape. The memory address counter (MAC) is used in conjunction with the SQ for counting timing intervals during magnetic tape operation. This arrangement avoids the necessity of providing a separate counter for use in magnetic tape timing. The MAC can be used in this application since it is not being used for memory operation during the time that magnetic tape timing intervals must be measured. The MAC, however, can only count to a maximum of 672 microseconds before it resets automatically. The SQ is required to provide the additional timing necessary for the relatively long intervals required for magnetic tape unit operation. These intervals are from two to four milliseconds in duration in definite increments. The SQ count is automatically incremented by one every two microseconds unless incrementing is prevented by an inhibit signal. Such a signal can come from several sources. External control cycling is achieved by first inhibiting the SQ count and then forcing the appropriate count into SQ. A separate, three-bit, eight-state, binary counter, the read control counter (RC), is employed for synchronization during read operations. This has been done because the read operation must be performed independently of the write operation when using a magnetic tape unit as the output device. The RC counter is described in paragraph 2.6.7.1.7.

External control mechanization has been divided into several relatively independent portions. These are described in paragraphs 2.6.7.1.1 through 2.6.7.1.8. External control mechanization lists are given in paragraph 2.6.7.1.9.

2.6.7.1.1 Standby

The SQ counter remains in the ZERO state (ØØ) until program control indicates that an operation is to start. When either an INPUT, OUTPUT, or ERASE signal is received, the input and output buffer registers and a number of control flip-flops are cleared and SQ steps to state ONE (Ø1).

2.6.7.1.2 Device Selection

When SQ, is in state ONE (Ø1), the go flip-flop (GO) is set, enabling the address of the selected in-out device to be gated onto the appropriate device switching unit (DSU) address lines to start the selected device. The device selected depends on the type of operation (input, output, etc.) and upon the setting of the device selection switches on the operator's panel. Simultaneously with gating the address onto the DSU lines, a decision is made as to the direction of data transfer to take place and appropriate control signals are sent to the selected device. To provide adequate turn-on time for the DSUs, states TWO (Ø2) through FIVE (Ø5) of SQ are not used.

2.6.7.1.3 Device Start

States SIX (Ø6) through SIXTEEN (16) of the SQ are used only when magnetic tape unit operation is required. Magnetic tape unit operation requires that a definite start-time interval be counted off prior to reading or writing information. These states of the SQ, in conjunction with cycling of the memory address counter (MAC), are used to count off that interval. The MAC has 336 states and a MAC cycle is completed in 672 microseconds. MAC cycling is initiated by a COUNT signal. The TL levels used in magnetic tape timing (start) mechanization are produced by decoders used to identify those intervals in the MAC cycle which are not modulo 672 microseconds in duration.

2.6.7.1.4 Read Operation

The read synchronizer controls the read operation and communicates with the selected in-out device and with internal control. The read synchronizer is described separately in paragraph 2.6.7.1.7. States SEVENTEEN (17) through

FORTY-TWO (42) of the SQ are used to monitor incoming information from the selected device. As a result of the monitoring, decisions are made as to data format (e.g., block marks, stop codes) and any special timing requirements (e.g., that two start-of-block marks must be received within 34 microseconds of each other when starting to read from magnetic tape). Read synchronizer operation starts at state SEVENTEEN of the SQ, as indicated by a start read (STRE) signal. The first start-of-block mark and the second end-of-block mark (or the end-of-file mark) are discarded when the CANCEL level is made high (raised to a logical ONE). This arrangement prevents the read synchronizer from sending a signal, the character in the input buffer register (CINI) signal, to internal control when such marks are received.

2.6.7.1.5 Write Operation

The set of SQ states (7 through 42) used for the read operation are also used to control timing of the write operation. This arrangement does not, however, prevent doing simultaneous read and write operations since the read synchronizer may still be used. Write operation control provide all synchronization necessary for the available output devices. Characters are transferred from the memory buffer register (MBR) to the output buffer register (OBR) under internal control. A signal, the character in the output buffer register (CINO) signal, indicates that such a transfer has occurred. After the data has been written by the selected output device, external control raises the finished data with output buffer register (FINO) level (to a ONE) to signal internal control to send another character to the OBR. Cycling of the sequence counter for proper internal timing during magnetic tape operation is carried out by looping the SQ at time X, as indicated in the Write Operation Mechanization listed in paragraph 2.6.7.1.9. The particular state of the SQ determines the cycle time, as shown in that mechanization.

2.6.7.1.6 Device Stop

No halt procedure is required for in-out devices other than magnetic tape units. External control continues to step the SQ until a clear external control signal (CLRSYN) is received from internal control. When this is received the SQ returns to the standby state (ØØ) and the control flip-flops are cleared. When magnetic tape units are used, however, states FIFTY (5∅) through SIXTY-SEVEN (67) of the SQ are used in counting off the required magnetic tape unit stop time. The time interval is counted off in the same manner as the magnetic tape unit start time, as described in 2. 6. 7. 1. 3.

2. 6. 7. 1. 7 Read Synchronizer

The read synchronizer employs a binary counter to control read operations. This counter is the read control (RC) counter, also known as the ST counter. The read synchronizer communicates with the selected input device, regulates acceptance of characters in the input buffer register (IBR), checks incoming characters for proper parity, and, by providing a character in the input buffer register (CINI) signal, informs internal control of arrival of information in the IBR. If external control decides that a particular character in the IBR is to be disregarded, the read synchronizer sends a CANCEL signal to internal control. Receipt of information from the IBR by internal control is acknowledged by raising of the finished data with input buffer register (FINI) level. The FINI level is raised by internal control at the same time as the contents of the IBR are transferred into the memory buffer register (MBR).

The read control (RC) counter is used to generate the timing states used in the read synchronizer. The RC counter is, essentially, a three-bit, eight-state, binary counter whose contents are automatically incremented by ONE every two microseconds except when it is in the standby state. The states of the RC flip-flops establish the ST states as shown on Table 2-7. Table 2-8 lists the logical equations for setting and clearing the RC flip-flops.

TABLE 2-8. READ COUNTER FLIP-FLOPS AND ST STATES

Read Counter Flip-Flop States			ST States
RC3	RC2	RC1	
Ø	Ø	Ø	STØ
Ø	Ø	1	ST7
Ø	1	Ø	ST1
Ø	1	1	ST2
1	Ø	Ø	ST3
1	Ø	1	ST4
1	1	Ø	ST5
1	1	1	ST6

TABLE 2-9. EQUATIONS FOR SETTING AND CLEARING THE READ COUNTER FLIP-FLOPS

IHST-. T*	1 =) A1ST*
ST1. STBD. Q*	1 =) RC1
ST2. SCL. STBD-. T*	Ø =) RC1
	1 =) RC3
ST=N. (IMT + OMT).SCL-. SKW. T*	Ø =) RC1
	1 =) RC2
	1 =) RC3
ST6. (IMT + OMT + FINI + CANCEL). T*	Ø =) RC2
	Ø =) RC3
STØ. STRE-	1 =) IHST
(FIN + CLRDC). T*	Ø =) RC1
	Ø =) RC2
	Ø =) RC3

2.6.7.1.8 Erase Operation

Erasure of information on a section of magnetic tape used during input can be carried out by using the ERASE switch on the operator's panel. States TWO (Ø2) through SEVENTY (7Ø) of the SQ are then used to count cycles of the memory address counter (MAC). A total erase time of 36.288 microseconds can be obtained.

Use of the erase capability of the OLCU should be confined only to cases in which an echo parity check error is found and attempts to rewrite the block of information continue to give an error indication.

2.6.7.1.9 External Control Mechanization Lists

The external control mechanization lists follow. These lists have been written in "machine language". The conventions used were explained on Table 2-7.

Standby

```
ØØ.INPUT-.OUTPUT-.BACK-.ERASE-                    1 =  IHSQ
ØØ.IHSQ-.t*                                       Ø =) GO
                                                  Ø =) RVF
                                                  Ø =) FRC
                                                  Ø =) RCF
                                                  Ø =) WCF
                                                  Ø =) ST
                                                  Ø =) RDYF
                                                  Ø =) STBF
                                                  Ø =) IBR
                                                  Ø =) OBR
```

Device Selection

Ø1.t*		1 =) GO
Ø1.(OLPD + BACK).t*		1 =) RVF
Ø1.BACK-.t*		1 =) FRC
Ø1.OMT-.OCP-.t*		1 =) RCF
Ø1.(OMT + ERASE).t*		1 =) WCF
	GO.RVF	1 = BUS9
	GO.FRC	1 = BUS1Ø
	GO.RCF	1 = BUS11
	GO.WCF	1 = BUS12

Erase

$SQ = N.ERASE N = Ø2 to 7Ø	1 = COUNT
$SQ = N.ERASE.TLM N = Ø2 to 7Ø	1 = IHSQ
Ø2.ERASE.TL5	1 = CLRWAR
71.ERASE.p*	1 =) H Ø =) ERASE
71.ERASE	1 = CLRSYN

Magnetic Tape Timing (Start)

$SQ = N.(IMT + OMT + BACK) N = Ø6 to 15	1 = COUNT
$SQ = N.(IMT + OMT + BACK).TLM- N = Ø6 to 15	1 = IHSQ
Ø6.IMT.TLM.t*	17 =) SQ
Ø6.BACK.TLM.t*	17 =) SQ
Ø6.OMT.TL2	1 = CLRWAR
13.OMT.TL4.t*	17 =) SQ

Magnetic Tape Timing (BLS to Data Gap, and Stop Time)

```
$SQ = N. (IMT + OMT + BACK)                              1 =  COUNT
      N = 5Ø to 67
$SQ = N. (IMT + OMT + BACK). TLM-                        1 =  IHSQ
      N = 5Ø to 67
5Ø. IMT. TL2. t*                                         Ø =) RCF
5Ø. IMT. TL5. t*                                         Ø =) FRC
54. IMT. TL6.. t*                                       7Ø =) SQ
5Ø. OMT. TL2. SBM. t*                                    1 =) RCF
5Ø. OMT. TL5. SBM-.                                      1 =  CLRWAR
5Ø. OMT. TLM. SBM.                                       1 =  STRE
53. OMT. TLM                                             1 =  FIN
54. OMT. TL7. SBM. t*                                   2Ø =) SQ
55. OMT. TL8. t*                                         Ø =) FRC
61. OMT. TL3. t*                                        7Ø =) SQ
5Ø. BACK. TL2. t*                                        Ø =) RCF
57. BACK. TL2. t*                                        Ø =) RVF
63. BACK. TL9. t*                                       7Ø =) SQ
```

Read Operation

```
17. (INPUT + BACK)                                       1 =  STRE
2Ø. (IMT + ICD + BACK). ST = 5-                          1 =  IHSQ
21. (IMT + ICD + BACK). LBS-. t*                        2Ø =) SQ
22. (IMT + ICD + BACK)                                   1 =  CANCEL
SQ = ICC. (IMT + ICD + BACK). ST = Ø-. ST = 1-. t*      36 =) SQ
35. (IMT + ICD + BACK). t*                              2Ø =) SQ
36. (IMT + ICD + BACK). st = 6-                          1 =  IHSQ
```

36. (IMT + ICD + BACK). LBS-. ST = 6. t*	2Ø =) SQ
36. BACK. ST = 6. LBS	1 = FIN
37. INPUT. ST = 5-	1 = IHSQ
4Ø. (IMT + ICD). LEBM-. t*	37 =) SQ
4Ø. IPT. LSTP-. t*	37 =) SQ
41. (IMT + ICD). ST = 5-	1 = IHSQ
42. (IMT + ICD). LEBM-. t*	1 =) IDTF
42. (IMT + ICD). LEBM	1 = CANCEL 1 = FIN

Write Operation

17. OCP	1 = STRE
17. (OUTPUT. SCL + OPT. SCL-). R*	1 =) OGT
2Ø. OUTPUT. OMT-. Q*	Ø =) OBR
2Ø. OMT. (EBM + SBM. RCF-)-. Q*	Ø =) OBR
2Ø. SCL. OMT-. OUTPUT. RDYD-	1 = IHSQ
21. OUTPUT	1 = FINO
21. (SCL-. OPT + SCL. OMT-. OUTPUT). CINO-	1 = IHSQ
22. SCL. OUTPUT. OMT-. RDYD-	1 = IHSQ
22. SCL-. OPT. (SYNCH1. SYNCH2-)-	1 = IHSQ
22. OUTPUT. R*	1 =) OGT 1 =) STBF
23. OUTPUT. R*	Ø =) STBF
23. OMT. SCL-. R*	Ø =) OGT
23. OPT. SCL-. R*	1 =) RVF
24. OPT. SCL-. (SYNCH1-. SYNCH2)-	1 = IHSQ
24. SCL. OUTPUT. OMT-. R*	2Ø =) SQ
25. OPT. SCL-. R*	2Ø =) SQ Ø =) RVF

to count odd microseconds for MT timing

SQ = X.OMT.A-.Q*   1 =) B

SQ = X.OMT.(RCF-.RDYF- + SBM-.RDYF).R*   2Ø =) SQ

SQ = X.OMT.SBM.R*   1 =) RDYF

SQ = X.OMT.EBM.R*   Ø =) RDYF

to count even microseconds for MT timing

SQ = X.OMT.(RCF-.RDYF- + SBM-.RDYF).R*   2Ø =) SQ

SQ = X.OMT.SBM.R*   1 =) RDYF

SQ = X.OMT.EBM.R*   Ø =) RDYF

The state of the sequence counter should be chosen according to the following table:

state (x)		24	25	26	27	3Ø	31	32	33	34	35	36
cycle time	even	1Ø	12	14	16	18	2Ø	22	24	26	28	3Ø
	odd	11	13	15	17	19	21	23	25	27	29	31

The R* is given by

A.Q*   1 =) R*

The sequence counter is initially synchronized with a t*

SQ = ØØ.(INPUT + OUTPUT + BACK + ERASE).t*   Ø =) A

SQ = ØØ-.A.-Q*   1 =) A

The sequence counter is incremented by

$SQ = N-.t*   1 =) A1SQ*
N = 17 to 41

$SQ = N.A.B.-.Q*   1 =) A1SQ*
N = 17 to 41

An odd count is obtained by using the B flip-flop

A.B.-.Q*   Ø =) A

A.B.Q*   Ø =) B

Read Control Synchronizer

BUS24	1 = STBD
IGT.(SCL- + STBD.SCL)	BUS/N =) IBR/N
SCL-.(IMT + OMT).(IBR6 + IBR7)	1 = SYNCH
SCL-.IPT.BUS13	1 = SYNCH
BUS15	1 = IHP
STØ.STRE.t*	7 =) ST
STØ-.IHST-.t*	1 =) A1ST*
FIN	1 = IHST

FIN.t*	Ø =) ST
	Ø =) IGT

ST7.SCL-.$BUS/N.(IMT + OMT + IPT + OCP)	1 = IHST
ST7.t*	0 =) IBR
ST1.IGT-.t*	1 =) IGT
ST1.SCL	1 = IHST
ST1.SYNCH-	1 = IHST
ST1.STBD.Q*	2 =) ST
ST2.STBD	1 = IHST
ST2.SCL.STBD-.t*	5 =) ST
ST2.(IMT + OMT + OCP).SCL-.t*	5 =) ST
IHP.IPT	1 = INBUS7
ST4.IHP.t*	1 =) IBR7
ST4.IHP.IPAR-.t*	1 =) CIBR8
ST5.t*	Ø =) IGT
ST5.IPT.IPAR-.IHP-	1 = IHST

ST5.IPT.IPAR-.IHP.p*	14 =) PC
	1 =) H
	∅ =) GO
	1 =) IDTF
ST6.CANCEL-	1 = CINI
ST6.(IMT + OMT + FINI + CANCEL).t*	7 =) ST
ST6	1 = IHST
ST5.(IMT + ICD).IPAR-.p*	1 =) IDTF
ST5.(OMT + OCP).IPAR-.p*	1 =) ODTF
IPT.(LDEL + LPTF)	1 = CANCEL

2.6.7.2 Internal Control

Internal control, which may also be described as data-handling control, is primarily concerned with performing all necessary conversions, card checking, and data storage and retrieval. Internal control regulates manipulation of data within the OLCU. Internal control generates and sequences the commands necessary to operate the input, output, and memory buffer registers (IBR, OBR, and MBR), the memory, and the memory address counter (MAC). It also monitors the states of the various OLCU registers. All decisions as to data transfer, storage, conversion, and retrieval are made by internal control. Internal control monitors the memory parity error circuits and also the various error checking circuits to insure that error indications are provided. Communication between internal and external control is carried on over two sets of lines, two lines to each set. Information sent over these lines indicates the status of data in the input and output buffer registers.

2.6.7.2.1 Internal Control Equipment

Internal control equipment includes a basic cycle counter (CY), a set of state flip-flops (SC1, SC2, and SC3), and several control flip-flops (SEC, END, etc.). Basic to all of the internal control mechanization is the memory basic cycle. The cycle counter generates the basic cycle of the OLCU memory. A ten-microsecond cycle was chosen to meet both memory and control requirements. A ten-microsecond cycle permits conservative memory design while, at the same time, allowing operation of the OLCU with magnetic tape equipment handling data at maximum average per character rates of up to 1ØØ,ØØØ characters per second if conversion is required and up to 2ØØ,ØØØ characters per second if conversion is not required.

The cycle counter is used to generate the basic memory cycle. It is designed to generate the read and write levels (RLV and WLV, respectively) with conservative timing and spacing. Unless specifically inhibited by the inhibit cycle counter sequencing (IHCY) signal, the cycle counter cycles automatically. The basic timing signals generated by the cycle counter are shown on Figure 2-9. The read level is generated during CYØ, CY1, and CY2. The write level is generated during CY5 and CY6.

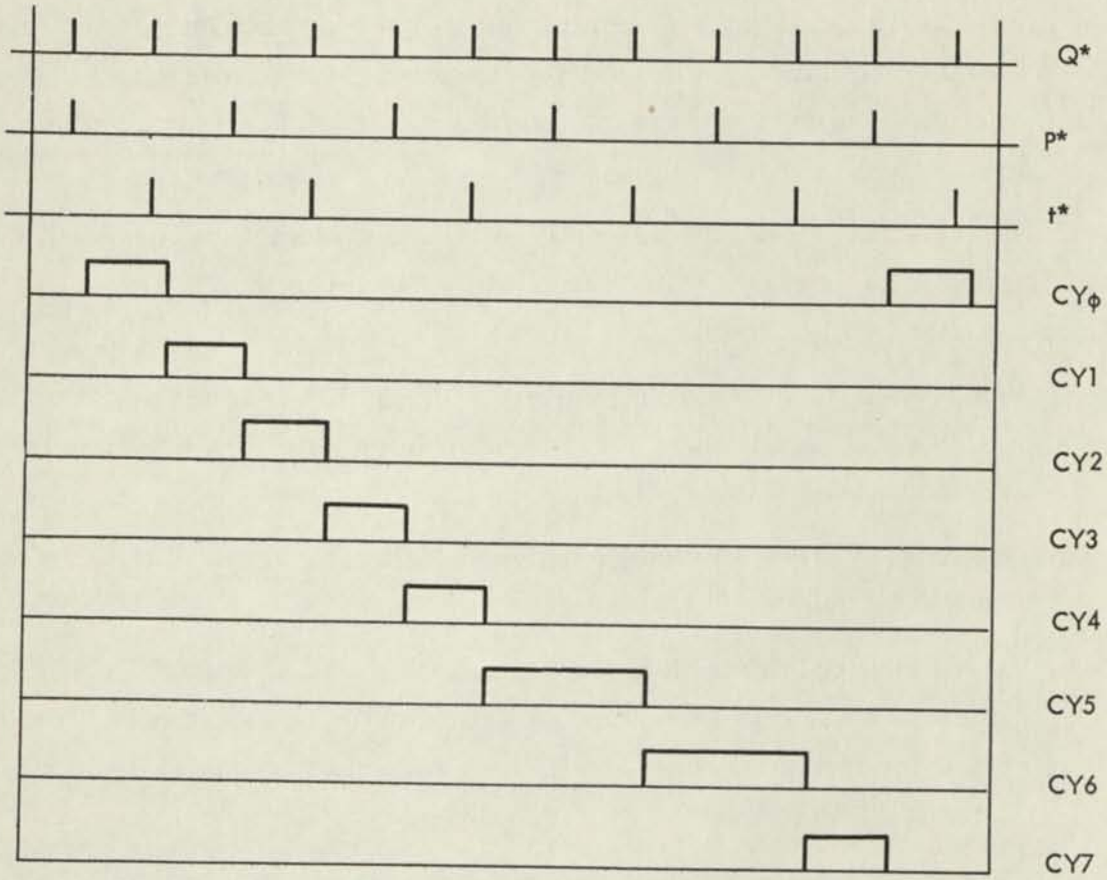


Figure 2-9. Timing Signals Generated by Cycle Counter (CY)

2.6.7.2.2 Internal Control Mechanization

Internal control is described in the form of mechanization lists, one for each internal control sequence. The mechanization lists describe the sequence of operations necessary to read in or read out data and to perform required format control or conversion. Wherever feasible, internal control has been so mechanized that the same internal control equipment is used in each mechanization. Care has also been taken to have similar actions take place at similar times even though such actions occur in the mechanization of different internal control sequences.

The mechanization of the internal control sequences is shown, in chart form, on Figure 2-1Ø through 2-23. The following notes will assist understanding of these mechanization charts.

1. The mechanizations are written in "machine language". For an explanation of the various "machine language" symbols and conventions, see Table 2-1Ø.
2. Rows indicate the events taking place during a specific state of the cycle counter (CY).
3. The inner columns (those columns between the vertical double lines) are read in sequence, starting with the left-hand column. Each inner column describes the action during one state of the internal control.
4. The initial state for a particular mechanization is always indicated in the first inner columns.
5. The time sequence of events within a state can be followed by reading down each column.
6. The first and last columns on each chart include the following-listed information:
   a. Above the horizontal line - the flip-flops or inputs whose states are used as conditions of the mechanization.
   b. Below the horizontal double line - the CY states and the pulse (p or t) that ends each CY state.

	CLE			
SC1 SC2 SC3	SC1- SC3-	SC1 SC2- SC3-	SC1 SC2- SC3	

Ø t		RLV = 1
		ISA = 1
		(XB = 0-): 3 =) XM
			RLV = 1
			INCR-: SXBM = 1
			INCR : SXBM + 1 = 1	

1 p		RLV = 1
		ISA = 1
			RLV = 1	

2 t		RLV = 1
		ISA = 1
			RLV = 1	

3 p		1 =) MBR7 1 =) MBR15		
4 t				
5 t		WLV = 1	WLV = 1	
6 t		WLV = 1	WLV = 1	

7 p	1 =) SC1
	Ø =) MBR
	Ø =) MAC
		Ø =) MBR
		(WCHK)(XB = Ø): 1 =) SC3
		(MAX- + DXBØ).(WCHK· DXBØ)-.
		MAC + 1 —>MAC
		(MAX)·(DXBØ) -: Ø =) SC1
		Ø =) MAC
			MAC + 1 =) MAC
			Ø =) MBR
			Ø =) SC3	

Figure 2-10. Clear Mechanization

		BACK
∅	t	
1	p	(COUNT-): IHCY = 1
2	t	
3	p	(COUNT): IHCY = 1 (COUNT-): ∅ = MAC
4	t	(CINI-) : IHCY = 1 (CINI)(LSCB): 1 =) CON
5	t	(COUNT-) : IHCY = 1 (COUNT)(p*): ∅ =) MAC
6	t	
7	p	(COUNT) : IHCY = 1 (COUNT-): ∅ =) MAC

Figure 2-11. Backspace Mechanization

	IMT																					
SC1 SC2 SC3	SC1- SC2- SC3-	SC1 SC2- SC3-						SC1 SC2 SC3-						SC1 SC2- SC3				SC1- SC2 SC3	SC1- SC2 SC3	SC1 SC2 SC3	SC1- SC2- SC3	SC1- SC2- SC3
CINI LEOF LEB BKT SEC		CINI- SEC-	CINI- SEC	CINI LEB- BKT-	CINI LEB- BKT	CINI LEOF- LEB	CINI LEOF LEB	CINI-	CINI LEB- BKT- KEY-	CINI LEB- KEY	CINI LEOF- LEB- BKT	CINI LEOF- LEB	CINI LEOF LEB	CINI-	CINI LEOF- LEB-	CINI LEOF- LEB	CINI LEOF LEB					CINI LEOF LEB BKT SEC
ø t	3 =) XBM ø =) END ø =) CON	RLV = 1 ISA = 1		▨	▨	▨	▨	▨	▨						▨	▨	▨				XB - 1 =) XBM	ø t
1 p	COUNT-: IHCY	RLV = 1 ISA = 1		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨		(COUNT-): IHCY = 1		(COUNT-): IHCY = 1 ø =) MAC	1 p
2 t		RLV = 1 ISA = 1		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨					2 t
3 p	(COUNT): IHCY = 1	IHCY = 1	IHCY = 1	IBR =) MBR (OUTHOL): SEC- =) SEC	1 =) SC2	IHCY = 1 1 =) SC2 1 =) SC3 (EMPTY): ø =) SC1	IHCY = 1 1 =) SC3 1 =) SC3 1 =) END	IHCY = 1	IHCY = 1	(KEY): IHCY = 1		IHCY = 1 1 =) SC3 (EMPTY): ø =) SC1	IHCY = 1 1 =) SC3 1 =) END	IHCY = 1	IHCY = 1	(EMPTY): ø =) SC1 1 =) SC2	1 =) END (OUTLP)(CON): EOB =) MBR		(COUNT): IHCY = 1	(OUTCP-):EOB =)MBR (OUTCP) :1 =) MBR7 1 =) MBR15	(COUNT): IHCY = 1	3 p
4 t	(CINI-): IHCY = 1 (LSCB): 1 =) CON	▨	▨	(SEC-): SLMBR = 1				▨						▨	▨		(OUTLP)(CON): SLMBR = 1			(OUTCP-)(SEC-): SLMBR = 1		4 t
5 t	p*: ø =) MAC		WLV = 1	▨	▨	▨	▨		▨	▨	▨	▨		(OUTLP)(CON): WLV = 1	▨	▨	▨	p*: ø =) MAC	p*: ø =) MAC	WLV = 1	(p*): ø =) MAC	5 t
6 t			WLV=1	▨	▨	▨	▨		▨	▨	▨	▨		(OUTLP)(CON): WLV = 1	▨	▨	▨			WLV = 1		6
7 p	ø =) MBR 1 =) SC1 (OUTCP)(CON): 1 =) SC3 (OUTLP)(CON): 1 =) SC3 (OUTCP)(WCHK): XB + 1 =) XB	SEC- =) SEC	ø =) MBR SEC- =) SEC (MAX-): MAC + 1 =) MAC (MAX): 1 =) SC3	▨	▨	▨	▨	IHCY = 1		(KEY-): IHCY = 1 (KEY): ø =) SC2		1 =) SC3 (EMPTY): ø =) SC1	1 =) SC3 1 =) END	ø =) SC1 (OUTCP)(CON)(WCHK): 1 =) LAST (OUTCP-)(OUTLP-): XB + 1 =) XB (CON-): XB + 1 =) XB	▨	▨	▨	ø =) SC3	ø =) SC2 ø =) IDTF	ø =) SC1 ø =) SC2 (SEC): XB + 1 =) XB 0 =) SEC (Møøø-): XB + 1 =) XB	CLRSYN = 1 ø =) SC3	7 p

Figure 2-12. Magnetic Tape Input Mechanization

	IPT										
SC1 SC2 SC3	SC1- SC2- SC3-			SC1 SC2- SC3-		SC1 SC2- SC3				SC1- SC2- SC3	SC1 SC2 SC3
LSTP + BKT MIS SEC END	SEC- END-	SEC- END	SEC END-	SEC-	SEC	(LSTP-)(BKT-) MIS SEC-	(LSTP-)(BKT-) MIS SEC	LSTP + BKT SEC-	LSTP + BKT SEC		LSTP + BKT MIS SEC END
Ø t	3 =) XBM	(OUTMT-)(OUTCP-): RLV=1, ISA=1 3=) XBM	3 =) XBM RLV = 1 ISA = 1	RLV = 1 ISA = 1		RLV = 1 ISA = 1		RLV = 1 ISA = 1		XB - 1 =) XBM	Ø t
1 p	Ø =) CON	(OUTMT-)(OUTCP-): RLV=1, ISA=1 3=) XBM	RLV = 1 ISA = 1	RLV = 1 ISA = 1		RLV = 1 ISA = 1		RLV = 1 ISA = 1			1 p
2 t		(OUTMT-)(OUTCP-): RLV=1, ISA=1 3=) XBM	RLV = 1 ISA = 1	RLV = 1 ISA = 1		RLV = 1 ISA = 1		RLV = 1 ISA = 1			2 t
3 p		EOB =) MBR		IBR =) MBR (OUTHOL): SEC- =) SEC	IBR =) MBR	(OUTCP): 1 =) MBR7 1 =) MBR15 (OUTLP): EOB =) MBR (OUTCP-)(OUTLP-): IBR =) MBR	(OUTCP): 1 =) MBR7 1 =) MBR15 (OUTLP): EOB =) MBR (OUTCP-)(OUTLP-): IBR =) MBR	(OUTCP): 1 =) MBR7 1 =) MBR15 (OUTCP-):EOB=)MBR	(OUTCP): 1 =) MBR7 1 =) MBR15 (OUTCP-):EOB=)MBR		3 p
4 t	SLMBR = 1	SLMBR = 1	(MBR15-): 1 =) CON	SLMBR = 1		SLMBR = 1		SLMBR = 1			4 t
5 t		OUTCP- · OUTMT-: WLV = 1			WLV = 1	WLV = 1	WLV = 1	WLV = 1	WLV = 1		5 t
6 t	FINI = 1	FINI = 1 (OUTCP-)(OUTMT-): WLV = 1	FINI = 1	FINI = 1	WLV = 1 FINI = 1	WLV = 1 FINI = 1	WLV = 1 FINI = 1	WLV = 1 FINI = 1	WLV = 1 FINI = 1	FINI = 1	6 t
7 p	(CINI-): IHCY = 1 (CINI)(LSTP): 1 =) END (OUTCP)(WCHK)(DXBØ): XB + 1 =) XB (CINI)(LSTP-)(BKT-)(OUTCP-)(OUTLP-): 1 =) SC1 (CINI)(LSTP-)(BKT-)(I7E6-): 1 =) SC1 (CINI)(LSTP-)(BKT-)(OUTCP-)(OUTLP-)(I7E6): 1 =) CON	CLRSYN = 1 Ø =) MAC Ø =) MBR (OUTCP)(WCHK): 1 =) LAST	(CINI-): IHCY = 1 (CINI) : 1 =) SC1 (CINI)(LSTP): 1 =) SC3 (CINI)(BKT): 1 =) SC3 (CINI)(MIS): 1 =) SC3	(CINI-): IHCY = 1 (CINI)SEC- =) SEC CINI· LSTP: 1 =) END : 1 =) SC3 CINI· BKT: 1 =) SC3 CINI· MIS: 1 =) SC3	(MAX): CLRSYN = 1 (CINI-): IHCY = 1 (CINI): Ø =) MBR : SEC- =) SEC : MAC + 1 =) MAC (CINI)(LSTP): 1 =) END 1 =) SC3 (MAX): Ø =) SC1 Ø =) MAC (CINI)(BKT + MIS): 1 =) SC3	CLRSYN = 1 Ø =) MBR Ø =) SC1 Ø =) SEC (MØØØ-): XB + 1 =) XB (OUTCP-): XB + 1 =) XB	CLRSYN = 1 XB + 1 =) XB Ø =) MBR Ø =) SC1 Ø =) SEC	(MØØØ-): XB + 1 =) XB (OUTCP-): XB + 1 =) XB Ø =) MBR Ø =) SC1 Ø =) SEC CLRSYN = 1	XB + 1 =) XB Ø =) MBR Ø =) SC1 Ø =) SEC CLRSYN = 1	CLRSYN = 1 Ø =) MAC Ø =) MBR Ø =) SEC Ø =) SC3	7 p

Figure 2-13. Paper Tape Input Mechanization

	ICC · WCHK													
SC1	SC1		SC1-		SC1									SC1
ACB LEBM ALARM NCB SEC	LEBM-	ACB- LEBM-	ACB- ALARM- SEC-	ACB LEBM- ALARM- SEC-	ACB LEBM- SEC-	ACB LEBM- SEC	ACB- LEBM- SEC	ACB- LEBM- SEC-	ACB- LEBM ALARM- NCB- SEC	ACB- LEBM ALARM- NCB SEC	ACB- LEBM ALARM- NCB SEC	ACB- LEBM ALARM SEC	ACB- LEBM ALARM SEC-	ACB LEBM ALARM LCARD + XB = NCB SEC
ϕ t	ICXB = 1	SXBL = 1	∅ =) END	▨	RLV = 1	▨	▨	(CHECK): RLV = 1	▨	▨	XB - 1 =) XBM	▨	XB -1 → XBM	∅ t
1 p				▨	RLV = 1	▨	▨	(CHECK): RLV = 1	▨	▨		▨		1 p
2 t				▨	RLV = 1	▨	▨	(CHECK): RLV = 1	▨	▨		▨		2 t
3 p				▨	SEC- =) SEC SLMBR = 1 SLOBR = 1	▨	▨	SEC- =) SEC SLMBR = 1 SLOBR = 1 (CHECK)(IBR6 = MBR14)-: 1 =) IDTF	▨	▨		▨	XB + 1 =, XB	3 p
4 t				▨	▨	SHMBR = 1 SLIBR = 1 SLOBR = 1	SHMBR = 1 SLIBR = 1	▨	▨	▨		▨		4 t
5 t				▨	▨	WLV = 1	(CHECK): WLV = 1	▨	▨	▨		▨		5 t
6 t			FINI = 1	▨	▨	WLV = 1 (MXA5): FINI = 1	(CHECK): WLV = 1 (MXA5): FINI = 1	▨	▨	▨	(MXA5):FINI=1	▨	(MXA5): FINI	6 t
7 p			IHCY = 1	(CINI-): IHCY = 1 (CINI): 1 =) SC1 ∅ =) MBR (CINI)(CHECK)(DXBϕ):XB + 1 =) XB	▨	(CINI-)(MXA5): IHCY=1 [CINI + (MXA5)-]: MAC + 1 =) MAC ∅ =) MBR SEC- =) SEC	CINI + MXA5-: MAC + 1 =) MAC ∅ =) MBR SEC- =) SEC (CINI-) (MXA5): IHCY = 1	▨	(CINI-): IHCY = 1 (CINI): SEC- =) SEC ∅ =) MBR MAC + 1 =) MAC (CHECK-): 1 =) CHECK (CINI): ∅ =) SC1	(CINI-): IHCY = 1 (CINI): SEC- =) SEC ∅ =) MBR (CINI)(LCARD): 1 =) END (CHECK-): 1 =) CHECK	CLRSYN = 1 ∅ =) MAC ∅ =) SC1	(CINI-): IHCY = 1 (CINI): ∅ =) CHECK SEC- =) SEC ∅ =) MBR MAC + 1 =) MAC	CLRSYN = 1 XB + 1 =) XB ∅ =) SC1	7 p

Figure 2-14. Column Card Input (with Checking) Mechanization

	ICC · WCHK-									
SC1	SC1	SC1-	SC1							SC1
LEBM ALARM NCB SEC	LEBM-	ALARM- SEC-	LEBM- SEC-	LEBM- SEC	LEBM ALARM- NCB- SEC	LEBM ALARM- NCB SEC	LEBM ALARM- NCB SEC-	LEBM ALARM SEC	LEBM ALARM	LEBM ALARM NCB SEC
Ø t		Ø =) END	RLV = 1	▨	▨	▨	XB - 1 =) XBM	▨	XB - 1 =) XBM	Ø t
1 p			RLV = 1	▨	▨	▨		▨	XB + 1 =) XB	1 p
2 t			RLV = 1	▨	▨	▨		▨		2 t
3 p			SEC- =) SEC SLMBR = 1 SLOBR = 1	▨	▨	▨		▨	XB + 1 =) XB	3 p
4 t			▨	SHMBR = 1 SLIBR = 1 SLOBR = 1	▨	▨		▨		4 t
5 t			▨	WLV = 1	▨	▨		▨		5 t
6 t		FINI = 1	▨	WLV = 1 (MXA5): FINI = 1	▨	▨	(MXA5): FINI = 1	▨	(MXA5): FINI = 1	6 t
7 p	ICXB = 1	(CINI-): IHCY = 1 (CINI): 1 =) SC1 Ø =) MBR	▨	CINI-.(XA=5) IHCY = 1 (CINI + XA = 5-): MAC + 1 =) MAC Ø =) MBR SEC- =) SEC	(CINI-): IHCY = 1 (CINI): SEC- =) SEC Ø =) MBR MAC + 1 =) MAC Ø =) SC1	(CINI-): IHCY = 1 (CINI): SEC- =) SEC Ø =) MBR XB + 1 =) XB (CINI)(LCARD): 1 =) END	CLRSYN = 1 Ø =) MAC Ø =) SC1	(CINI-): IHCY (CHECK): Ø =) CHECK (CINI): SEC- =) SEC Ø =) MBR MAC + 1 =) MAC	CLRSYN = 1 XB + 1 =) XB Ø =) SC1	7 p

Figure 2-15. Column Card Input (without Checking) Mechanization

	ICR · WCHK													
			SC1-	SC1-	SC1									SC1
ACB LEBM ALARM NCB SEC	ACB	LEBM-	ACB- ALARM- SEC-	ACB LEBM- ALARM- SEC-	ACB LEBM- SEC-	ACB LEBM- SEC	ACB- LEBM- SEC-	ACB- LEBM- SEC	ACB- LEBM ALARM- NCB- SEC	ACB- LEBM ALARM- NCB SEC-	ACB- LEBM ALARM- NCB SEC-	ACB- LEBM ALARM SEC	ACB- LEBM ALARM SEC-	ACB LEBM ALARM NCB SEC
ø t	ISA = 1	ICXB = 1	ø =) END	▨	RLV = 1		SXBL = 1 (CHECK): RLV = 1		▨	▨	XB - 1 =) XBM	▨	XB - 1 =) XBM	ø t
1 p				▨	RLV = 1		(CHECK): RLV= 1		▨	▨		▨		1 p
2 t				▨	RLV = 1		(CHECK): RLV = 1		▨	▨		▨		2 t
3 p				▨	IBR =) MBR	IBR =) MBR		SLMBR = 1	▨	▨		▨	XB + 1 =) XB	3 p
4 t				▨	SLMBR = 1		(CHECK)(IBR = MBR): 1 =) IDTF	(CHECK)(IBR = MBR): 1 =) IDTF	▨	▨		▨		4 t
5 t				▨		WLV = 1	(CHECK): WLV = 1		▨	▨		▨		5 t
6 t			FINI = 1	▨	FINI = 1	WLV = 1 FINI = 1	FINI = 1 (CHECK): WLV = 1	FINI = 1	▨	▨	FINI = 1	▨	FINI = 1	6 t
7 p			IHCY = 1	(CINI-): IHCY = 1 (CINI): 1 =) SC1 ø =) MBR (CINI)(CHECK)(DXBø): XB + 1 =) XB		(CINI): MAC + 1 =) MAC ø =) MBR		(CINI): ø =) MBR (CINI)(YA = XA + 1): YA - 1 =) YA (CINI)(YA = XA + 1)- : MAC + 1 = MAC	(CINI): MAC + 1 =) MAC ø =) SC1	(CINI)(LCARD): 1 =) END	CLRSYN = 1 ø =) MAC ø =) SC1	(CINI-): IHCY = 1 (CINI): SEC- =) SEC ø =) MBR MAC + 1 =) MAC ø =) CHECK	CLRSYN = 1 ø =) SC1 XB + 1 =) XB	7 p
					(CINI-): IHCY = 1 (CINI): SEC- =) SEC		(CINI-): IHCY = 1 (CINI): SEC- =) SEC		(CINI-): IHCY = 1 (CINI): SEC- =) SEC ø =) MBR (CHECK-): 1 =) CHECK					

Figure 2-16. Row Card Input (with Checking) Mechanization

	ICR · WCHK-									
SC1		SC1-	SC1							SC1
LEBM ALARM NCB SEC		ALARM- SEC-	LEBM- SEC-	LEBM- SEC	LEBM ALARM- NCB- SEC	LEBM ALARM- NCB SEC	LEBM ALARM- NCB SEC-	LEBM ALARM SEC	LEBM ALARM SEC-	LEBM ALARM NCB SEC
∅ t	ISA	∅ =) END	RLV = 1		▨	▨	XB-1 =) XBM	▨	XB - 1 =) XBM	∅ t
1 p			RLV = 1		▨	▨		▨	XB + 1 =) XB	1 p
2 t			RLV = 1		▨	▨		▨		2 t
3 p			IBR =) MBR	IBR =) MBR	▨	▨		▨	XB + 1 =) XB	3 p
4 t			SLMBR = 1		▨	▨		▨		4 t
5 t				WLV = 1	▨	▨		▨		5 t
6 t		FINI	FINI = 1	WLV = 1 FINI = 1	▨	▨	FINI =1	▨	FINI = 1	6 t
7 p		(CINI-): IHCY = 1 (CINI): 1 =) SC1 ∅ =) MBR	(CINI-): IHCY = 1 (CINI): SEC- =) SEC	(CINI-): IHCY = 1 (CINI): SEC- =) SEC ∅ =) MBR MAC + 1 =) MAC	(CINI-): IHCY = 1 (CINI): SEC- =) SEC ∅ =) MBR MAC + 1 =) MAC ∅ =) SC1	(CINI-): IHCY = 1 (CINI): SEC- =) SEC ∅ =) MBR XB + 1 =) XB (CINI)(LCARD): 1 =) END	CLRSYN = 1 ∅ =) MAC ∅ =) SC1	CIHI -: IHCY CINI: SEC-=) SEC ∅ =)MBR MAC + 1 =)MAC ϕ =) CHECK	CLRSYN = 1 XB + 1 =) XB ∅ =) SC1	7 p

Figure 2-17. Row Card Input (without Checking) Mechanization

	OMT											
SC1 SC2 SC3	SC1- SC2- SC3-	SC1 SC2- SC3-						SC1- SC2 SC2-	SC1 SC2			SC1 SC2 SC3
BIY' MEB + MISM SEC MXAØ + MXA3		MEB- · MISM- SEC-	MEB- · MISM- SEC	BIY- MEB + MISM SEC	BIY- MEB + MISM SEC- (MXAϕ-)(MXA3)- (MXAϕ)-·(MXAB)-	BIY- MEB + MISM SEC- MXAØ + MXA3	BIY MEB + MISM		SEC- (XA=ϕ)-·(XA=3)-	SEC	SEC- (XA=ϕ) + (XA=3)	BIY MEB + MISM SEC (MXAØ)MXA3
Ø t		RLV = 1		▨	▨	▨	▨					Ø t
1 p	(COUNT-): IHCY	RLV = 1		▨	▨	▨	▨	(COUNT-): IHCY = 1				1 p
2 t		RLV=1		▨	▨	▨	▨					2 t
3 p	(COUNT): IHCY	(INHOL): SEC- =) SEC	SLMBR = 1	▨	▨	▨	▨	(COUNT): IHCY = 1				3 p
4 t	(FINO-) IHCY = 1 (FINO)(CON-): SOB =) OBR (FINO)(CON): SCB =) OBR	(FINO-): IHCY = 1 (FINO): MBR =) OBR	(FINO-): IHCY = 1 (FINO): MBR =) OBR	(FINO-): IHCY = 1 (FINO): 1 =) OBR7	(FINO-): IHCY = 1 (FINO): 1 =) OBR7	(FINO-): IHCY = 1 (FINO)(END-)(CON-): EOB =) OBR (FINO)(END-)(CON): ECB =) OBR (FINO)(END) : EOF =) OBR			(FINO-): IHCY = 1 (FINO): 1 =) OBR7	(FINO-): IHCY = 1 (FINO): 1 =) OBR7	(FINO-): IHCY = 1 (FINO)(END-)(CON-): EOB =) OBR (FINO)(END-)(CON): ECB =) OBR (FINO)(END) : EOF =) OBR	4 t
5 t	(p*): Ø =) MAC (COUNT-): IHCY = 1	WLV = 1	(INHOL): WLV = 1	WLV = 1								5 t
6 t		WLV = 1	(INHOL): WLV = 1	WLV = 1								6 t
7 p	(COUNT): IHCY = 1 (COUNT-): Ø =) MAC Ø =) MBR 1 =) SC1	SEC- =) SEC	SEC- =) SEC Ø =) MBR (MAX-): MAC + 1 =) MAC (MAX): Ø =) MAC 1 =) SC2	MAC + 1 =) MAC SEC- =) SEC 1 =) SC2	SEC- =) SEC 1 =) SC2	Ø =) SC1 Ø =) MAC Ø =) SEC 1 =) SC2		Ø =) MAC Ø =) SC2 Ø =) CON CLRSYN = 1 (MISM)(MEB-)(INPT): SEC- =) SEC (MISM-) + (MEB) + (INPT-): Ø =) MBR	SEC- =) SEC	SEC- =) SEC MAC + 1 =) MAC	Ø =) SC1 Ø =) MAC	7 p

Figure 2-18. Magnetic Tape Output Mechanization

	OLPT					
SC1 SC2 SC3	SC2-				SC2	SC1 SC2 SC3
MISM + MEB SEC	(MISM + MEB)- SEC-	(MISM + MEB)- SEC	MISM + MEB SEC-	MISM + MEB SEC		MISM + MEB SEC
∅ t	RLV = 1		RLV = 1			∅ t
1 p	RLV = 1		RLV = 1			1 p
2 t	RLV = 1		RLV = 1			2 t
3 p	FINO-: IHCY FINO. (INHOL):SEC-=) SEC	SLMBR = 1				3 p
4 t	MBR =) OBR	MBR =) OBR	(MEB)(END): STOP =) OBR (MEB-)(END-): MBR =) OBR		STOP =) OBR	4 t
5 t	WLV = 1	(INHOL): WLV = 1	WLV = 1			5 t
6 t	WLV = 1 CINO = 1	(INHOL): WLV = 1 CINO = 1	WLV = 1 (MEB)(END): CINO = 1 (MEB-)(END-): CINO = 1		CINO = 1	6 t
7 p	(FINQ-): IHCY = 1 (FINO): SEC- =) SEC	(FINO-): IHCY = 1 (FINO): SEC- =) SEC ∅ =) MBR (FINO)(MAX-): MAC + 1 =) MAC (FINO)(MAX): ∅ =) MAC (FINO)(MAX)(END-): CLRSYN = 1 ∅ =) CON (FINO)(MAX)(END): 1 =) SC2	(FINO-): IHCY = 1 (FINO): CLRSYN = 1 ∅ =) SEC ∅ =) MBR ∅ =) MAC ∅ =) CON		FINO: CLRSYN = 1 ∅ =) CON ∅ =) SC2 ∅ =) SEC (FINO-): IHCY = 1	7 p

Figure 2-19. Line Printer, Paper Tape Output Mechanization

	OCC · WCHK											
SC1 SC2		SC1- SC2-	SC1 SC2-		SC1 SC2	SC1 SC2			SC1- SC2			SC1 SC2
ALARM XBXM ACB EBM	EBM-	ALARM-	ACB- EBM-	ACB EBM-		ACB EBM-	ACB EBM	ACB- EBM-	ALARM- XBXM- ACB EBM	ALARM- XBXM ACB EBM	ALARM ACB EBM	ALARM XBXM ACB EBM
ϕ t	ICXB = 1	(LAST): 1 ⇒XBM	(LAST-): RLV = 1	▨	SXBL	(CHECK): RLV = 1	▨	▨				ϕ t
1 p			(LAST-): RLV = 1	▨		(CHECK): RLV = 1	▨	▨				1 p
2 t			(LAST-): RLV = 1	▨		(CHECK): RLV = 1	▨	▨				2 t
3 p			SLMBR = 1 SLOBR = 1	▨		SLMBR = 1 SLOBR = 1 (CHECK)(MBR14 = IBR6)-: 1 ⇒ODTF	▨	▨				3 p
4 t			SLMBR = 1 SLIBR = 1 1 ⇒OBR7	▨		SLMBR = 1 SLIBR = 1	▨	▨				4 t
5 t			(LAST-): WLV = 1	▨		(CHECK): WLV = 1	▨	▨				5 t
6 t			(LAST-): WLV = 1 (MXA5-): CINO = 1	▨		(CHECK): WLV = 1 (MXA5): FINI = 1	▨	▨	FINI = 1	FINI = 1	FINI = 1	6 t
7 p		(FINO-): IHCY = 1 (FINO): ϕ ⇒MBR 1 ⇒SC1 (FINO)(DXBϕ): XB + 1 ⇒XB	(FINO-)(MXA5): IHCY = 1 (FINO + MXA5-): ϕ ⇒MBR MAC + 1 ⇒MAC	IHCY = 1 ϕ ⇒MBR 1 ⇒SC2		(CINI-)(MXA5): IHCY (CINI + MXA5-): ϕ ⇒MBR MAC + 1 ⇒MAC	(CINI-): IHCY = 1 (CINI): ϕ ⇒MBR ϕ ⇒SC1 (CHECK-): 1 ⇒CHECK	IHCY = 1 ϕ ⇒SC2 ϕ ⇒MBR	ϕ ⇒MBR ϕ ⇒SC2 MAC + 1 ⇒MAC	ϕ ⇒MBR ϕ ⇒SC2 ϕ ⇒MAC (END): 1 ⇒LAST CLRSYN = 1	CLRSYN = 1 ϕ ⇒SC2 ϕ ⇒MBR ϕ ⇒MAC ϕ ⇒CHECK	7 p

Figure 2-20. Column Card Output (with Checking) Mechanization

	OCC · WCHK-							
SC1	SC1-	SC1						SC1
LROW ALARM XBXM	ALARM-	LROW-	LROW-	LROW ALARM- XBXM-	LROW ALARM- XBXM	LROW ALARM		LROW ALARM XBXM
Ø t	(LAST): 1 =) XBM	ICXB = 1	(LAST-): RLV = 1	▨	▨	▨		Ø t
1 p			(LAST-): RLV = 1	▨	▨	▨		1 p
2 t			(LAST-): RLV = 1	▨	▨	▨		2 t
3 p			SLMBR = 1 SLOBR = 1	▨	▨	▨		3 p
4 t			SLMBR = 1 SLIBR = 1 1 =) OBR7	▨	▨	▨		4 t
5 t			(LAST-): WLV = 1	▨	▨	▨		5 t
6 t			(LAST-): WLV = 1 (MXA5): CINO = 1	▨	▨	▨		6 t
7 p	(FINO-): IHCY = 1 (FINO): Ø =) MBR 1 =) SC1		(FINO-)(MXA5): IHCY = 1 (FINO + MXA5-): Ø =) MBR MAC + 1 =) MAC	Ø =) SC1 Ø =) MBR MAC + 1 =) MAC	CLRSYN = 1 Ø =) SC1 Ø =) MBR Ø =) MAC	CLRSYN = 1 Ø =) SC1 Ø =) MBR Ø =) MAC (CHECK): Ø =) CHECK		7 p

Figure 2-21. Column Card Output (without Checking) Mechanization

	OCR · WCHK															
SC1 SC2	SC1- SC2-		SC1 SC2-			SC2	SC1 SC2				SC1- SC2					SC1 SC2
EBM ACB ALARM XBXM SEC	ALARM- SEC-	EBM- ACB- ALARM- SEC-	EBM- ACB- SEC-	EBM- ACB- SEC	EBM- ACB SEC		EBM- ACB SEC-	EBM- ACB SEC	EBM- ACB- SEC	EBM ACB SEC	EBM ACB ALARM- XBXM- SEC	EBM - ACB ALARM- XBXM- SEC	EBM- ACB- ALARM- XBXM- SEC	EBM ACB ALARM- XBXM SEC	EBM ACB ALARM SEC	EBM ACB ALARM XBXM SEC
Ø t	LAST: 1 =) XBM	LAST: 1 =) XBM	(LAST-): RLV = 1		▨	SXBL = 1	(CHECK): RLV = 1		▨	▨		▨	▨			Ø t
1 p			(LAST-): RLV = 1		▨		(CHECK): RLV = 1		▨	▨		▨	▨			1 p
2 t			(LAST-): RLV = 1		▨		(CHECK): RLV = 1		▨	▨		▨	▨			2 t
3 p				SLMBR = 1	▨			SLMBR = 1	▨	▨		▨	▨			3 p
4 t			(LAST-): MBR =) OBR (LAST) : 1 =) OBR7	(LAST-): MBR =) OBR (LAST) : 1 =) OBR7	▨		(CHECK)(IBR = MBR-): 1 =) ODTF	(CHECK)(IBR = MBR)-: 1 =) ODTF	▨	▨		▨	▨			4 t
5 t			(LAST-): WLV = 1		▨		(CHECK): WLV = 1		▨	▨		▨	▨			5 t
6 t			CINO = 1 (LAST-): WLV = 1	CINO = 1	▨		FINI = 1 (CHECK): WLV = 1	FINI = 1	▨	▨	FINI = 1	▨	▨	FINI = 1	FINI	6 t
7 p	IHCY = 1	(FINO-): IHCY = 1 FINO: Ø =) MBR 1 =) SC1 (FINO)(WCHK): XB + 1 =) XB	(FINO-): IHCY = 1 (FINO) : SEC- =) SEC	(FINO-): IHCY = 1 (FINO) : SEC- =) SEC Ø =) MBR MAC + 1 =) MAC	(CINI-): IHCY = 1 CINI: SEC- =) SEC Ø =) MBR YA - 1 =) YA 1 =) SC2		(CINI-): IHCY = 1 (CINI) : SEC- =) SEC	(CINI-): IHCY = 1 (CINI) : SEC- =) SEC Ø =) MBR MAC + 1 =) MAC	Ø =) SC2 IHCY = 1	(CINI-): IHCY = 1 (CINI) : Ø =) MBR Ø =) SC1 (CHECK-): 1 =) CHECK	IHCY = 1	IHCY = 1	1 =) SC1 Ø =) SC2 IHCY = 1	CLRSYN = 1 SEC- =) SEC Ø =) SC2 Ø =) MBR Ø =) MAC (END): 1 =) LAST	CLRSYN = 1 SEC- =) SEC Ø =) SC2 Ø =) MBR Ø =) MAC Ø =) CHECK	7 p

Figure 2-22. Row Card Output (with Checking) Mechanization

	OCR · WCHK-						
SC1	SC1-	SC1					SC1
LROW ALARM XBXM SEC	LROW- ALARM- SEC-	LROW- SEC-	LROW- SEC	LROW ALARM- XBXM- SEC	LROW ALARM- XBXM SEC	LROW ALARM SEC	LROW ALARM XBXM SEC
∅ t	(LAST): 1 =) XBM	(LAST-): RLV = 1		▨	▨	▨	∅ t
1 p		(LAST-): RLV = 1		▨	▨	▨	1 p
2 t		(LAST-): RLV = 1		▨	▨	▨	2 t
3 p			SLMBR = 1	▨	▨	▨	3 p
4 t		(LAST-): MBR =) OBR (LAST): 1 =) OBR7	(LAST-): MBR =) OBR (LAST): 1 =) OBR7	▨	▨	▨	4 t
5 t		(LAST-): WLV = 1		▨	▨	▨	5 t
6 t		CINO = 1 (LAST-): WLV = 1	CINO = 1	▨	▨	▨	6 t
7 p	(FINO-): IHCY = 1 (FINO): 1 =) SC1 ∅ =) MBR	(FINO-): IHCY = 1 (FINO): SEC- =) SEC	(FINO): MAC + 1 =) MAC ∅ =) MBR : SEC- =) SEC	∅ =) SC1 SEC- =) SEC ∅ =) MBR MAC + 1 =) MAC	CLRSYN = 1 ∅ =) MBR SEC- =) SEC ∅ =) MAC ∅ =) SC1	CLRSYN = 1 ∅ =) MBR SEC- =) SEC ∅ =) MAC ∅ =) SC1	7 p

Figure 2-23. Row Card Output (without Checking) Mechanization

7. Each of the inner columns includes the following-listed information:

   a. Above the horizontal double line - the state of the flip-flops or inputs which are necessary for the indicated action to take place. These states indicate the state of the internal control. If a flip-flop or input noted in the first or last column is not listed on an inner column, the state of that flip-flop or input does not affect the indicated action.

   b. Below the horizontal double line - the actions and any special conditions required for some actions which occur during the CY state indicated in the first or last column.

8. Conditions are listed as follows:

   a. Above the horizontal double line - as actual signal names or abbreviations of actual signal names.

   b. Below the horizontal double line - as actual signal names or abbreviations of actual signal names, enclosed in parentheses.

9. Shaded areas on the charts represent internal control states which are "impossible" during normal operation.

2.6.7.3 Program Control

Program control, also identified as master control, was designed to integrate the external and internal control functions, the registers, and memory into the overall OLCU design. Separation of the program control functions from internal and external control functions was in keeping with the aim of simplicity and directness of design. The necessity of employing a highly-complex sequential circuit design was avoided. The logical design task was simplified since the internal states of external and internal control did not need to be considered in determining the proper OLCU operating sequences. Program control functions in a manner similar to use of an executive program in modern programming practice. Program control regulates the status and instruction signals required for OLCU operation. These signals are obtained either from a program counter (PC) or from controls on the operator's panel.

Each discrete state of the PC specifies a unique OLCU operation. The control logic associated with the PC determines the next operation to be performed in sequence. No decisions of this nature are made by external or internal control logic. External and internal control are, therefore, subsidiary, or slave, areas in relation to program control.

Two general conditions are usually considered by program control during input, output, or clear operations. The first condition indicates that the specified operation is actually taking place. The second condition indicates that the specified operation has been completed. After completion of a specified operation, program control logic selects the next state of the program counter. This state is selected after program control has checked the various alarm condition circuits and the device and mode selection circuits. During normal OLCU operation, all operator's panel controls except those used for halting the OLCU are inoperative until the specified data transfer or conversion has been completed. Transients on the lines between the operator's panel and the program control will not cause OLCU failures, assuming, of course, that no attempt is made to change from one device to another during the normal data processing sequence.

Program control operation is started by operation of the START switch on the operator's panel. The PC then automatically assumes an initial start state (state 11). In that state a check is made to determine whether or not the data transfer or conversion determined by the setting of the controls on the operator's panel is consistent with the media, formats, and OLCU capabilities. If there is any inconsistency, the PC assumes a standby state (state ØØ) and no data is processed. If there are no inconsistencies, the PC immediately and automatically assumes the state appropriate to the action specified by the setting of the controls on the operator's panel. Except when a card reader-punch is to be used the OLCU starts in an input operation state. When a card reader-punch is to be operated, the OLCU starts in a clear operation state.

It is not necessary to clear the OLCU when starting a data transfer operation. Clearing is accomplished automatically during one of the PC states. When the program counter is in that state, all registers and several control flip-flops are cleared. Normally, the operator does not need to operate the CLEAR switch.

2.6.7.3.1 Program Counter

The program counter (PC) is a 4-stage, 16-state binary counter. It is used to indicate the 16 possible program control states. Figure 2-24 is a chart

ØØ STANDBY	Ø1 DO BACKSPACE OPERATION AND RETURN TO STANDBY	Ø3 DO BACKSPACE OPERATION	Ø2 INPUT OPERATION COMPLETE
Ø4 DO OUTPUT OPERATION	Ø5 BACKSPACE OPERATION COMPLETE	Ø7 ALARM DURING OUTPUT OPERATION	Ø6 DO ERASE OPERATION AND RETURN TO STANDBY
14 SINGLE CYCLE MODE, OUTPUT OPERATION COMPLETE	15 SPECIAL	17 CLEAR OPERATION COMPLETE	16 SINGLE CYCLE MODE, INPUT OPERATION COMPLETE
1Ø OUTPUT OPERATION COMPLETE	11 INITIAL START	13 DO CLEAR OPERATION	12 DO INPUT OPERATION

Figure 2-24. Program Control State Diagram

indicating the meaning of each state and the actions taken in each state. Each "do" action represents an internal control sequence. Simplification of PC logic was a prime factor in determining the PC states which correspond to the various program control states.

2.6.7.3.2 Program Control Mechanization

The set of equations which corresponds to each of the program counter states describes exactly how program control operates in that state. The program control mechanization lists follow.

```
PC = ØØ          STANDBY

(PC = ØØ)(STR).($W).H.CLOCK*                        : Ø =) H
(PC = ØØ)($W-).BIM-.BOM-.H-.ERASE-.p*               : 11 =) PC
(PC = ØØ)(BSI + BSO).($W).H.CLOCK*                  : Ø =) H
(PC = ØØ).BSI.($W).CLOCK*                           : 1 =) BIM
(PC = ØØ).BSO.($W).CLOCK*                           : 1 =) BOM
(PC = ØØ).(BIM + BOM)($W-).H-.p*                    : Ø1 =) PC
(PC = ØØ).ERS.($W).CLOCK*                           : 1 =) ERASE
                                                      Ø =) H
(PC = ØØ).ERASE.($W-).H-.p*                         : Ø6 =) PC


PC = Ø1      DO BACKSPACE OPERATION AND RETURN TO STANDBY

(PC = Ø1)                                           : BACK = 1
(PC = Ø1)(CY = 7).COUNT-.p*                         : ØØ =) PC
                                                      1 =) H
                                                      Ø =) BIM
                                                      Ø =) BOM
```

```
PC = Ø2          INPUT OPERATION COMPLETE

(PC = Ø2). ALARM-. VY-. RUN. p*                            : Ø4 =) PC
(PC = Ø2). ALARM-. VY. INCR-. RUN. p*                      : 12 =) PC
(PC = Ø2). ALARM-. VY. INCR. RUN. p*                       : 13 =) PC
(PC = Ø2). ALARM-. VY-. SCY. p*                            : 16 =) PC
(PC = Ø2). ALARM-. VY. SCY. INCR. p*                       : 13 =) PC
(PC = Ø2). MPE. NHE-. INCR-. p*                            : Ø =) MPE
(PC = Ø2). MPE. NHE-. INCR. p*                             : ØØ =) PC
                                                             1 =) H
(PC = Ø2). IDTF. NHE-. p*                                  : ØØ =) PC
                                                             1 =) H
(PC = Ø2). ALARM-. p*                                      : Ø =) MPE
                                                             Ø =) IDTF

PC = Ø3          DO BACKSPACE OPERATION

(PC = Ø3)                                                  : BACK = 1
(PC = Ø3). (CY = 7). (COUNT)-. p*                          : Ø5 =) PC

PC = Ø4          DO OUTPUT OPERATION

(PC = Ø4)                                                  : OUTPUT = 1
(PC = Ø4)(OUTPUT + OUTLP)(CY = 7). SC2-. SEC. MAX.
        END-. FINO. p*                                     : 1Ø =) PC
(PC = Ø4)(OUTPTP + OUTLP)(CY = 7). SC2-. MEM1.FINO. p*     : 1Ø =) PC
(PC = Ø4)(OUTPTP + OUTLP)(CY = 7). SC2. FINO. p*           : 1Ø =) PC
```

```
(PC = Ø4).OUTMT.(CY = 7).SC1-.SC2.SC3-.p*                 : 1Ø =) PC

(PC = Ø4).OUTCP.WCHK.(CY = 7).SC1-.SC2.
        (XBXBM + ALARM).p*                                 : 1Ø =) PC

(PC = Ø4).OUTCP.WCHK-.(CY = 7).SC1.LROW.
        (XBXBM + ALARM).p*                                 : 1Ø =) PC


PC = Ø5          BACKSPACE OPERATION COMPLETE

(PC = Ø5).BOM-.p*                                          : Ø4 =) PC

(PC = Ø5).BOM.p*                                           : Ø7 =) PC
                                                             1 =) H
                                                             Ø =) BOM


PC = Ø6          ERASE AND RETURN TO STANDBY

(PC = Ø6).ERASE-.p*                                        : ØØ =) PC
                                                             1 =) H


PC = Ø7          ALARM DURING OUTPUT OPERATION

(PC = Ø7)(STR + ERS + BSO)($W).H.CLOCK*                    : Ø =) H

(PC = Ø7)($W-).BOM-.ERASE-.H-.p*                           : Ø4 =) PC
                                                             Ø =) MPE
                                                             Ø =) ODTF

(PC = Ø7).ERS.($W).CLOCK*                                  : 1 =) ERASE

(PC = Ø7).BSO.($W).CLOCK*                                  : 1 =) BOM

(PC = Ø7).BOM.($W-).p*                                     : Ø3 =) PC
```

```
PC = 1Ø          OUTPUT OPERATION COMPLETE

(PC = 1Ø).ALARM-.OUTCP-.END-.(INCR- + WCHK-).
        RUN.p*                                        : 12 =) PC

(PC = 1Ø).ALARM-.OUTCP-.END.p*                        : ØØ =) PC
                                                        1 =) H

(PC = 1Ø).ALARM-.INCR.WCHK.END-.p*                    : 13 =) PC

(PC = 1Ø).ALARM-.OUTCP.LAST-.WCHK.p*                  : 13 =) PC

(PC = 1Ø).ALARM-.OUTCP.(LAST + END.WCHK-).p*          : ØØ =) PC
                                                        1 =) H

(PC = 1Ø).ALARM-.OUTCP.WCHK-.END-.p*                  : 12 =) PC

(PC = 1Ø).ALARM-.OUTCP-.(INCR- + WCHK-).END-.
        SCY.p*                                        : 14 =) PC
                                                        1 =) H

(PC = 1Ø).NHE.p*                                      : Ø =) MPE
                                                        Ø =) ODTF

(PC = 1Ø).ALARM.p*                                    : Ø7 =) PC
                                                        1 =) H


PC = 11          INITIAL START

(PC = 11).OPERROR.p*                                  : ØØ =) PC
                                                        1 =) H

(PC = 11).OPERROR-.p*                                 : Ø =) OLCU
                                                        Ø =) MPE
                                                        Ø =) IDTF
                                                        Ø =) ODTF

(PC = 11).OPERROR-.OUTCP-.p*                          : 12 =) PC

(PC = 11).OPERROR-.OUTCP.p*                           : 13 =) PC
```

```
INMT-. INPT-. INCR-                                              : OPERROR = 1
OUTMT-. OUTPT-. OUTCP-. OUTLP-. VY-                              : OPERROR = 1
INCR. INCROW-. INCL-. INHOL-                                     : OPERROR = 1
OUTCP. OUTCROW-. OUTCL-. OUTHOL-                                 : OPERROR = 1
BIY-. ALF-. OUTMT. INCR-                                         : OPERROR = 1

PC = 12              DO INPUT OPERATION

PC = 12                                                          : INPUT = 1
(PC =12). INPT. (CY = 7). SC1-. SC3-. END. OUTMT. p*             : Ø3 =) PC
(PC = 12). INPT. (CY = 7). SC1-. SC3-. END. OUTCP.
        WCHK-. p*                                                : ØØ =) PC
                                                                   1 =) H
(PC = 12). INPT. (CY = 7). SC1-. SC3-. END. (OUTLP + OUTPTP).
        p*                                                       : Ø2 =) PC
(PC = 12). INPT. (CY = 7). SC1-. SC3-. END. OUTCP. WCHK. p*      : Ø2 =) PC
(PC = 12). INPT. (CY = 7). SC1. SC3-. MAX. SEC. p*               : Ø2 =) PC
(PC = 12). INPT. (CY = 7). SC1-. SC3. p*                         : Ø2 =) PC
(PC = 12). INMT. (CY = 7). SC1-. SC2-. SC3.
        (OUTCP- + CON- + WCHK). p*                               : Ø2 =) PC
(PC = 12). INMT. (CY = 7). SC1-. SC2-. SC3. OUTCP. CON.
        WCHK-. p*                                                : ØØ =) PC
                                                                   1 =) H
(PC = 12). INCR. (CY = 7). SC1. LEBM. SEC-. p*                   : Ø2 =) PC
```

```
PC = 13              DO CLEAR OPERATION

PC = 13                                                   : CLE = 1

(PC = 13)(CY = 7). SC1. MAX. DXBØ-. p*                    : 17 =) PC

PC = 14        SINGLE CYCLE MODE, OUTPUT COMPLETE

(PC = 14). STR. ($W). H. CLOCK*                           : Ø =) H

(PC = 14). ($W-)(OUTCP- + LAST-). H-. p*                  : 12 =) PC

(PC = 14). ($W-). OUTCP. LAST. H-. p*                     : Ø4 =) PC

PC = 15                  SPECIAL

(PC = 15). RI. $W                                         : ISR =) IBR

(PC = 15). STR. ($W). H. CLOCK*                           : Ø =) H

(PC = 15). ($W-). H-. p*                                  : 12 =) PC
                                                            1 =) GO
                                                            Ø =) IDTF

PC = 16        SINGLE CYCLE MODE, INPUT COMPLETE

(PC = 16). STR. ($W). H. CLOCK*                           : Ø =) H

(PC = 16). ($W-). H-. VY-. p*                             : Ø4 =) PC

(PC = 16). ($W-). H-. VY. p*                              : 12 =) PC

PC = 17          CLEAR OPERATION COMPLETE

(PC = 17). LAST-. RUN. p*                                 : 12 =) PC

(PC = 17). LAST. RUN. p*                                  : Ø4 =) PC
```

```
(PC = 17). VY-. SCY. p*                                    : 14 =) PC
                                                             1 =) H

(PC = 17). VY. SCY. p*                                     : 16 =) PC
                                                             1 =) H
```

2.6.7.4 Miscellaneous Mechanizations

Three sets of mechanizations, not described elsewhere in the OLCU mechanization lists, are given in 2.6.7.4.1 through 2.6.7.4.3.

2.6.7.4.1 Run and Single Cycle

The OLCU ordinarily operates in the RUN mode. At times, however, it is desirable to perform a single input or output operation and then halt the OLCU. Provision has been made for such operation. This operating mode is used primarily for maintenance and error correction.

Manual selection of the SINGLE CYCLE or RUN modes is done by operation of the appropriate switch on the operator's panel. In general, the RUN level is high (logical ONE) except when operating in the SINGLE CYCLE mode (i.e., except when the SINGLE CYCLE switch has been operated). If the HALT switch is operated, the single cycle control flip-flop (SCYF) is set and the OLCU is forced to operate in the SINGLE CYCLE mode. When the end-of-tape area is detected during magnetic tape unit operation, the end-of-tape flip-flop (EOTF) is set. The state of the EOTF is sensed periodically during each OLCU operating cycle. When the fact that EOTF is set is sensed, the OLCU is forced into the single cycle mode. This action takes place automatically and is not under operator control. If operation of the OLCU was halted by setting of EOTF, clearing of EOTF permits operation to resume in the RUN mode.

When the OLCU is re-started after having been halted either by operation of the HALT switch or by detection of the end-of-tape area, the SCYF and EOTF are reset automatically during re-starting and operation resumes in the RUN mode.

The mechanization is as follows:

```
SCYF-. SY-. (EOTF- + PC=16-. PC=17-)        : 1 = RUN
RUN-                                        : 1 = SCY
```

2.6.7.4.2 End-of-Tape Signal

The OLCU receives an end-of-tape signal from the magnetic tape unit via bus 28 when the end-of-tape area is detected. If this area is detected during magnetic tape input to the OLCU, the OLCU completes the succeeding output operation and then halts. The mechanization is as follows:

```
        BUS28.(IMT + OMT)                          : 1 = EOTW
        EOTW.p*                                    : 1 =) EOTF
```

EOTF is cleared as follows:

```
        PC=11.operror-.p*                          : Ø =) EOTF
        CSY.t*                                     : Ø =) EOTF
        CLR.$W.H.CLOCK*                            : Ø =) EOTF
```

When the OLCU halts as a result of detection of the end-of-tape area, the operator should remove the completed tape and install a new tape. Ordinarily, no attempt should be made to read or write another block of information on a magnetic tape whose end-of-tape area has been detected. It is possible, however, to read or write one more block of information even after detection of the end-of-tape area. In such a case, the START switch is operated and the OLCU will complete one input operation, one output operation, and then halt.

2.6.7.4.3 Device Alarms

A device alarm signal may be generated by such conditions as a power failure, broken tape, insufficient tape loop, out of paper. The device alarm signal is sent to the OLCU via bus 14. When the device alarm signal is received, appropriate indicators on the operator's panel are lighted and the OLCU halts. Device alarm signals are associated with equipment malfunctions or conditions; they are not associated with data errors. Device alarm mechanization is as follows:

```
        BUS14                                      : 1 = DVA
        DVA.(INPUT + OUTPUT + BIM + BOM +          : Ø =) GO
        ERASE).p*
                                                   : 1 =) H
```

DVA.(INPUT + BIM).p* : 1 =) IDVA
DVA.(OUTPUT + ERASE + BOM).p* : 1 =) ODVA

The input device alarm flip-flop (IDVA) and the output device alarm flip-flop (ODVA) drive appropriate indicators on the operator's panel. By noting which indicator is lighted the operator can determine whether an input or an output device alarm condition halted the OLCU. The IDVA and ODVA flip-flops are cleared automatically when the OLCU is re-started after appropriate corrective action has been taken. These flip-flops are cleared as follows:

PC=11. operror-.p* : ∅ =) IDVA
: ∅ =) ODVA

2.6.8 Timing Pulse Generator

The timing pulse generator is used to generate the p and t pulses required for OLCU operation. The timing pulse generator uses a crystal-controlled oscillator in conjunction with a commutating flip-flop to generate two series of pulses. The pulses in each series occur at 2-microsecond intervals. Those occurring in one series are known as the p pulses (p*); those in the other series as the t pulses (t*). The p and t pulses are shifted in phase in such a way that there is a 1-microsecond interval between a p pulse and the next succeeding t pulse. In addition to the p and t pulses, the OLCU also used q pulses (q*). Q pulses, which are generated at 1-microsecond intervals, are in phase with the p and t pulses. Relative timing of the p, t, and q pulses is shown on Figure 2-9. Timing pulse generator operation is regulated by the state of the halt flip-flop (H) as follows. When H is set, the timing pulse generator is turned off. Timingpulse generator operation is also regulated by the single pulse generator. The single pulse generator provides a 1-microsecond gate to the timing pulse generator. This mode of operation is for use during maintenance operation only.

2.6.9 Single Cycle Control Flip-Flop

The single cycle control flip-flop (SYCF) is set automatically whenever the HALT switch is operated. When the HALT switch is operated the SYCF is set and the OLCU is forced to halt at the end of the in-out cycle then being performed. When the OLCU is re-started, the SYCF is cleared automatically.

2.6.10 Alarm Indications

The OLCU design includes comprehensive alarm checking and error correction provisions. Error and alarm conditions are recorded by appropriate flip-flops whose states indicate the nature of the error or alarm condition. Suitable indication of such conditions is displayed on the operator's panel. Whenever feasible within the over-all OLCU design, a memory operation parity check is made during input and output. Indication that the end-of-tape area on magnetic tape has been detected is provided. Device and data errors are indicated as well as whether such errors occurred during an input or an output operation. The error correction procedure depends upon the nature and circumstance of the error, for example: a device error during an output operation. If a device error occurs during any operation the OLCU is immediately and automatically halted and the appropriate indicator on the operator's panel is lighted. The operator then takes suitable corrective action. No attempt should be made to re-start the OLCU until the cause of the device error has been corrected. A blower failure indicator is also provided. This light will indicate failure of any blower in van 2. In addition, failure of the blower in the S602A unit will cause the OLCU to stop after the present in-out cycle. This is done to prevent damage to information in the memory due to overheating. Memory parity errors are discussed in 2.6.10.1. The data and device alarm conditions are described in further detail in 2.6.10.2 and 2.6.10.3, respectively.

2.6.10.1 Memory Parity Errors

Memory parity is checked for errors only when information is read out from memory for use. It is not checked, for example, during the normal 'read before write' sequence that takes place in the basic OLCU cycle. Detected memory parity errors are indicated on the operator's panel.

2.6.10.2 Data Errors

The OLCU can operate in either of two modes in respect to data errors. Normally, it operates in the halt on error mode. When operating in this mode detection of a data error usually causes the OLCU to halt after completion of the current operation. With paper tape the OLCU stops on the character on which the error was detected. If, however, the IGNORE DATA ERROR switch is operated, the OLCU will continue to operate despite any parity or data errors. In that mode characters with wrong parity have the parity bits changed. If a data error is made when reading cards, due to card checking, the doubtful bit is assumed to be a ONE. This assumption is made because during card reading a ONE is more likely to be read as a ZERO than is a ZERO to be read as a ONE. The non-halt on error mode is intended primarily for use during OLCU maintenance and trouble-shooting.

2.6.10.2.1 Input Data Errors

Three types of input data errors cause error indication on the operator's panel. These are:

1. The parity of the incoming character is not correct. Data entering the OLCU is checked for parity as follows:
    a. Magnetic tape unit or magnetic tape-type device - checked for odd parity.
    b. Eight-channel paper tape unit - checked for even parity.
    c. Five-channel paper tape unit - not checked for parity and, therefore no input data error indication is provided.
2. Failure to receive two end-of-block marks at the end of each block or card read during magnetic tape or card reader-punch input. It should be noted that receipt of only a single start-of-block mark is not considered as an error. Even if only one such mark is received, external control continues searching for another. The OLCU does not accept data until the second has been received.
3. Occurrence of a check error when using card reader-punch input with checking.

If an input data error occurs, the program counter (PC) returns to the standby state. The operator can then either re-start the machine after first clearing both the input device and output device, and then starting the OLCU again, or he may re-start the OLCU without first correcting the error. If the latter course of action is followed, the error flip-flop will be cleared automatically during re-starting.

2.6.10.2.2 Output Data Errors

Two types of output data errors cause error indication on the operator's panel. These are:

1. Occurrence of a check error when using card reader-punch output with checking.
2. Receipt of a parity error signal when using an output device which operates according to SCL 1986. Certain of these devices, such as the militarized magnetic tape transports, may return a signal to the transmitting unit when a character having incorrect parity is received.

When a data error occurs during an output operation the program counter automatically cycles to state ∅7 and the OLCU halts. The operator then has his choice of alternative actions. One, he may re-start the OLCU by pressing the START switch and continue the operation without having corrected the error. Two, if a magnetic tape unit is being used as either an input or output device, he may backspace the tape and then re-start the OLCU. Three, if a magnetic tape unit is being used as an output device, he may first backspace the tape and then, after erasing the tape, re-start the OLCU. Operation of the CLEAR switch clears the OLCU and returns the program counter to the standby state. This switch must not, therefore, be used as an alternative to backspacing the tape or backspacing and then erasing the tape.

2.6.10.3 Device Errors

Device errors may occur at any time during off-line data processing and always require halting of the OLCU. When a device error is detected, a device error signal is produced which sets the halt flip-flop (H), thereby stopping the device and halting the OLCU at the end of the current operation. Regardless of the type of device, a power failure in the selected device produces a device error signal. Several specific device conditions also produce a device error signal, as follows:

1. Magnetic tape units - short tape loop, loss of vacuum, tape breakage, out of tape.

2. Paper tape punch - tape breakage, out of tape.
3. Card reader-punch - jammed cards.

When a device error halt occurs, the operator must then clear the OLCU and take appropriate corrective action to get the device back in operation before attempting to continue off-line data processing.

2.7 CIRCUIT DESIGN

This paragraph describes certain circuits which have been developed for use in MOBIDIC 7A. These are: the memory circuits, console switch interlocks, and the half-register driver. In addition, the power requirements for the off-line equipment are noted.

2.7.1 Memory System

The OLCU memory system requires four normally open switches to select any word in memory, two for the X and two for the Y direction. These switches are designated XA, XB, YA and YB. Reference should be made to Figure 2-25 to fully understand how a specific word is selected in memory. There are six XA switches, labeled XAØ to XA5, and four XB switches, labeled XBØ to XB3, making possible a selection from a total of twenty-four X co-ordinates (6 x 4 = 24). The fourteen Y co-ordinates (7 x 2 = 14) are selected by means of seven YA switches, YAØ through YA6, and two YB switches, YBØ and YB1.

The switch matrices chosen are economical and simplify understanding of switch matrix operation. Use of these matrices has reduced the cost of the decoders associated with the memory address counter. Simplicity of understanding eases trouble-shooting and maintenance. The switches are associated with the XA, XB, YA, and YB counters in the memory address counter (MAC). These counters are used not only to select a memory address but also to record pertinent information during read in or read out, as described in 2.6.6.3.

The operation of the OLCU memory is similar to all coincident-current memories, particularly the MOBIDIC B core memory system. The drivers are matrixed with diodes rather than transformers, mainly because of the lower power consumption. The main differences from the MOBIDIC B system is the method in which the drivers are used.

Since there are only two current sources used, one for the X-drive lines and one for the Y-drive lines, there are only two normally-closed switches employed, as shown on Figure 2-25. The function of these switches is to shunt the current sources to ground when neither a read nor write coincident current is required.

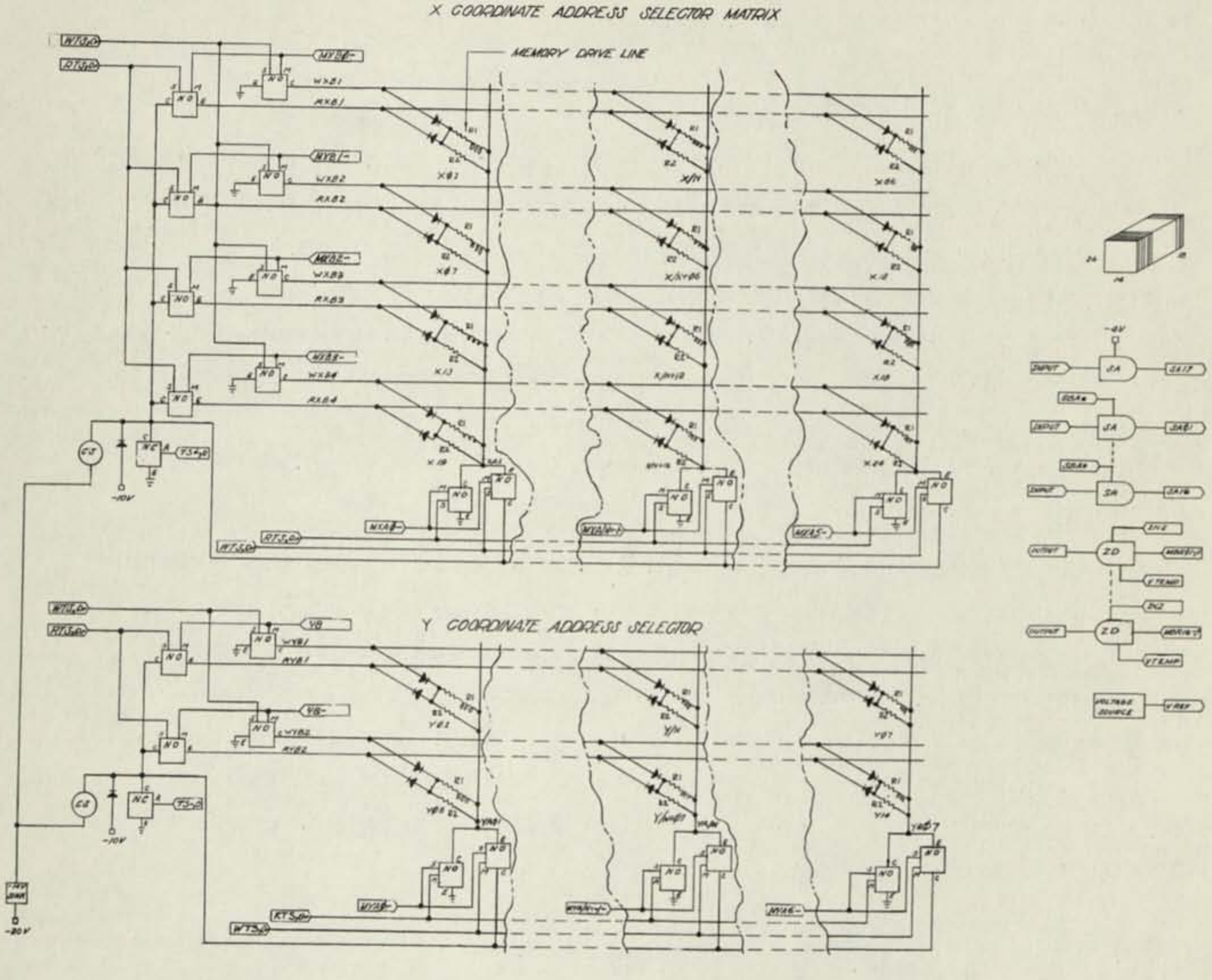


Figure 2-25. Memory Address Selection Matrix

Also unlike the MOBIDIC B system is the use of half of the normally-open transistor switches in series with the drive lines; the other half are used to shunt the drive lines to ground. A further explanation of the method of using the switches is given in detail in later descriptions.

2.7.1.1 Direct Drive System

A direct-drive, coincident-current system is employed by the OLCU for X-drive and Y-drive line selection in the core memory. A typical X-drive or Y-drive configuration is shown in Figure 2-26. A 2.7 ohm resistor is placed in series with each drive line for sampling while a 180 ohm resistor shunts each drive line for damping. The OLCU memory system employs smaller planes than does the MOBIDIC B memory system. The OLCU planes have less inductance and resistance per drive line. This permits use of the 180 ohm clamping resistor for damping purposes in place of the series RC shunt (a 330 ohm resistor and a 3300 mmf capacitor) used in MOBIDIC B.

Another major difference between the MOBIDIC B and OLCU memory systems is the use of but two diodes per drive line in the OLCU system. A saving of two diodes per drive line is accomplished in the OLCU system by employing one-half of the normally-open switch, output transistors in series with the read or write current paths. In this circuit arrangement, the emitter-collector diodes of the output transistors in the unselected normally-open switches serve the same isolation function as the extra diodes required in the MOBIDIC B system.

The memory system used in the OLCU permits use of an emitter follower type of direct drive current system. This more economical type of drive can be used since the smaller and slower OLCU memory has less inductance and tolerates slower rise and fall times. The difference in drive current rise times between the MOBIDIC B and OLCU system, for example, is 0.2 microseconds (0.4 microseconds and 0.6 microseconds, respectively).

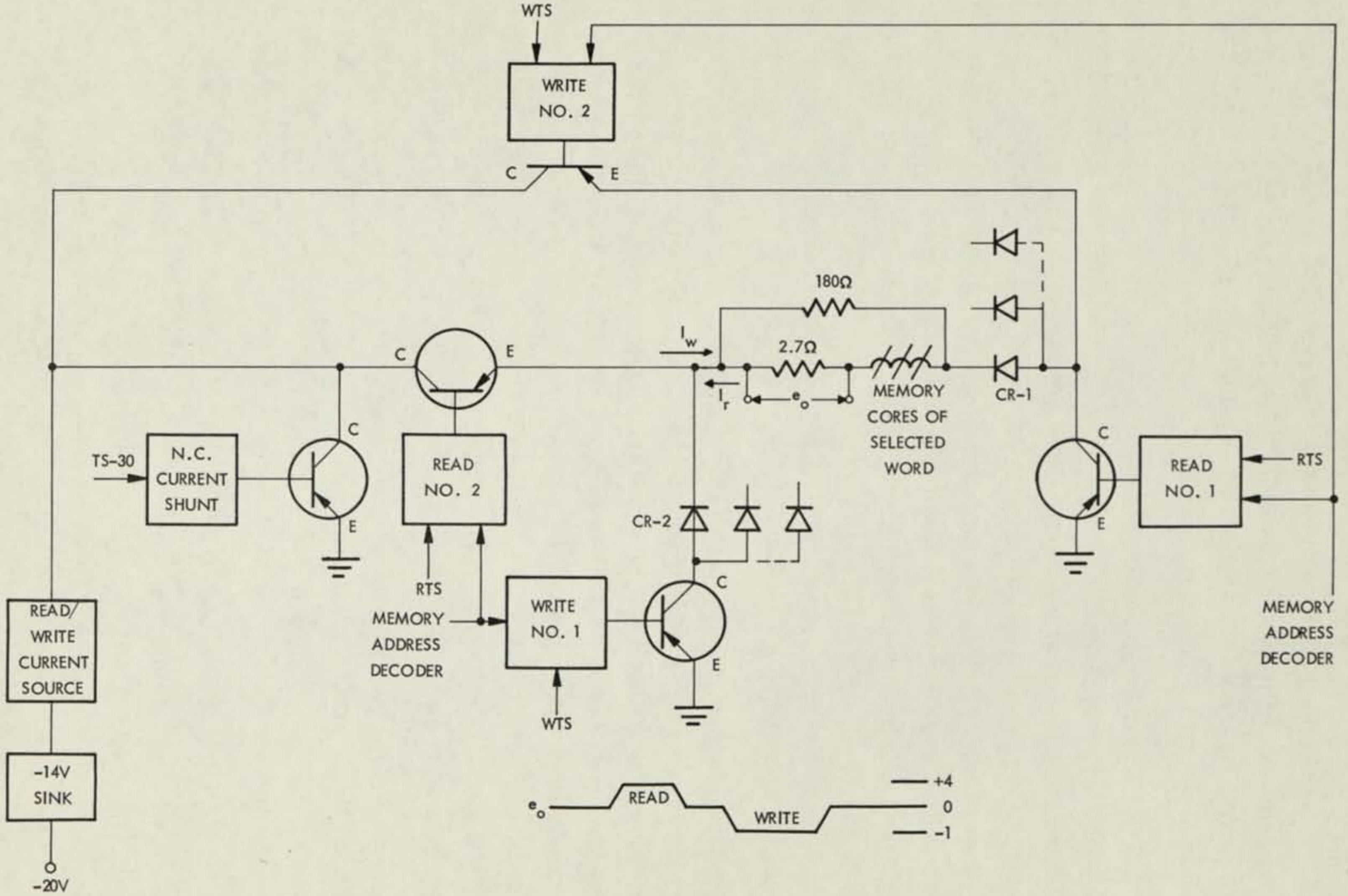


Figure 2-26. Typical X-Drive or Y-Drive Line Selection Configuration

2.7.1.2 Drive Line Selection

A total of 38 normally-open transistor switches are employed by the OLCU memory system for selecting word locations. In addition to these selection switches, two normally-closed switches are used to shunt the X-drive and Y-drive current sources to ground whenever a read or write drive current is not required.

Figure 2-25 shows the memory address selection matrices for both the X and Y co-ordinates. Generation of all the control signals (WTS, RTS and TS-), setting and clearing of the memory control and inhibit control flip-flops, and the strobe pulse (SBA*) which is used to provide time discrimination among the 16 read-out sense amplifiers are shown on Figure 2-27. The selection signals MXA, MXB, MYA and YB are decoded outputs of the memory address counters XA, XB, YA and YB. The decoders are shown on Figures 2-28 and 2-29, while the memory address counter is shown on Figure 2-30.

2.7.1.3 Memory Circuits

A MOBIDIC element, known as element number 92 and shown on Figure 2-31, was designed to be used as either a normally-closed or normally-open switch in the OLCU. This element, which is a double-sized element, contains essentially the same circuitry as was used in the MOBIDIC B core memory. The logic gate is used for the normally-open switch only. The purpose of this gate is to supply the current demands that are required by the emitter of the timing input transistor. This gate, in effect allows communication between the memory address decoder and the normally-open switch.

The input to element 92 when it is used as a normally-closed switch is three base loads. In the OLCU, emitter followers are used for the TS- logic, thereby satisfying the input requirements of three base loads each for the normally-closed switches.

In packaging these switches, 5 type-92 (double-sized) elements are included on a single logic-package assembly. No additional elements are placed on such an assembly since the five type 2N600 output transistors are mounted on brackets on the logic card itself. No decoupling elements are included in the package since the decoupling components used for the -4, -10, and +4 voltages are also mounted directly on the logic card.

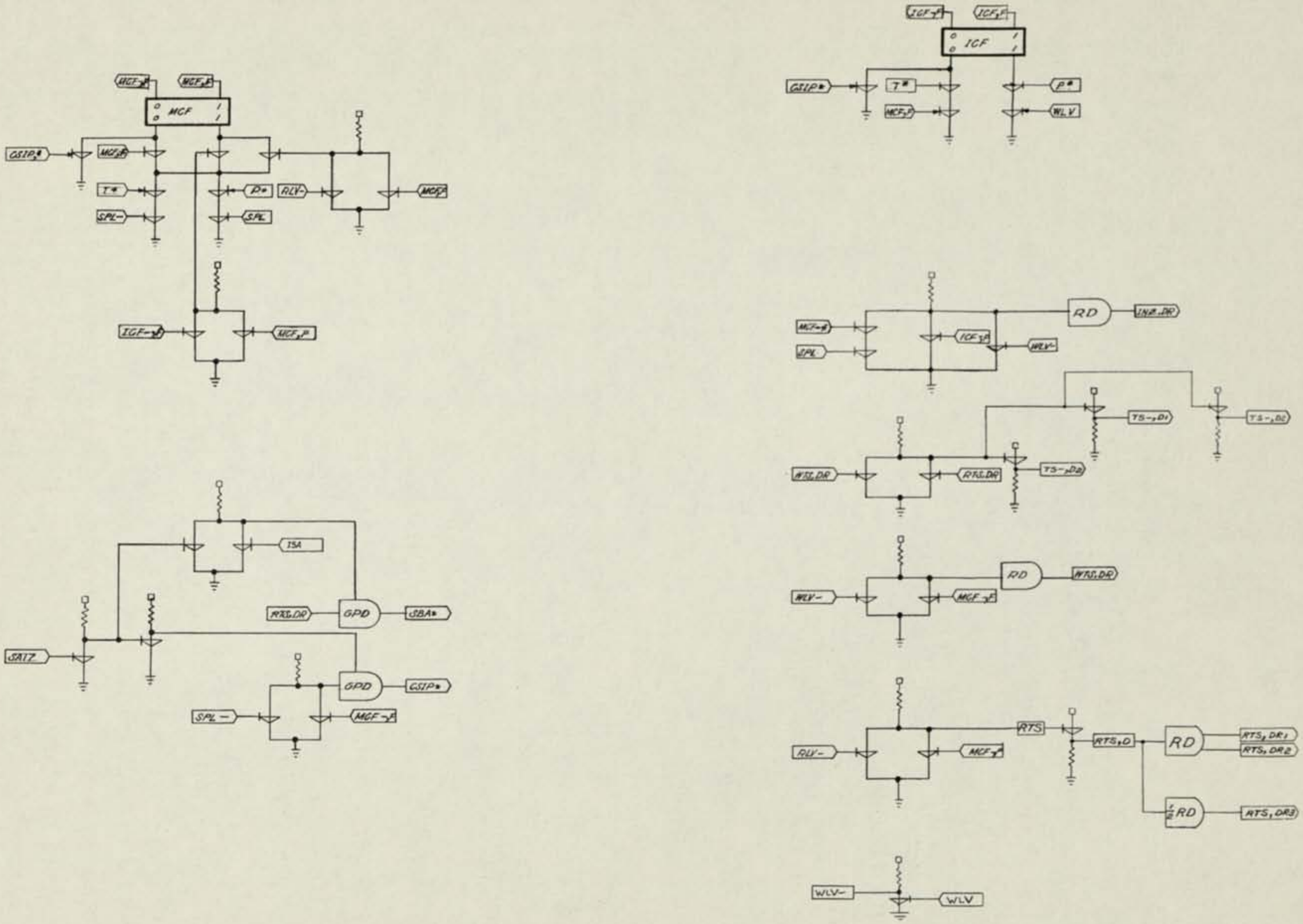


Figure 2-27. Memory Control

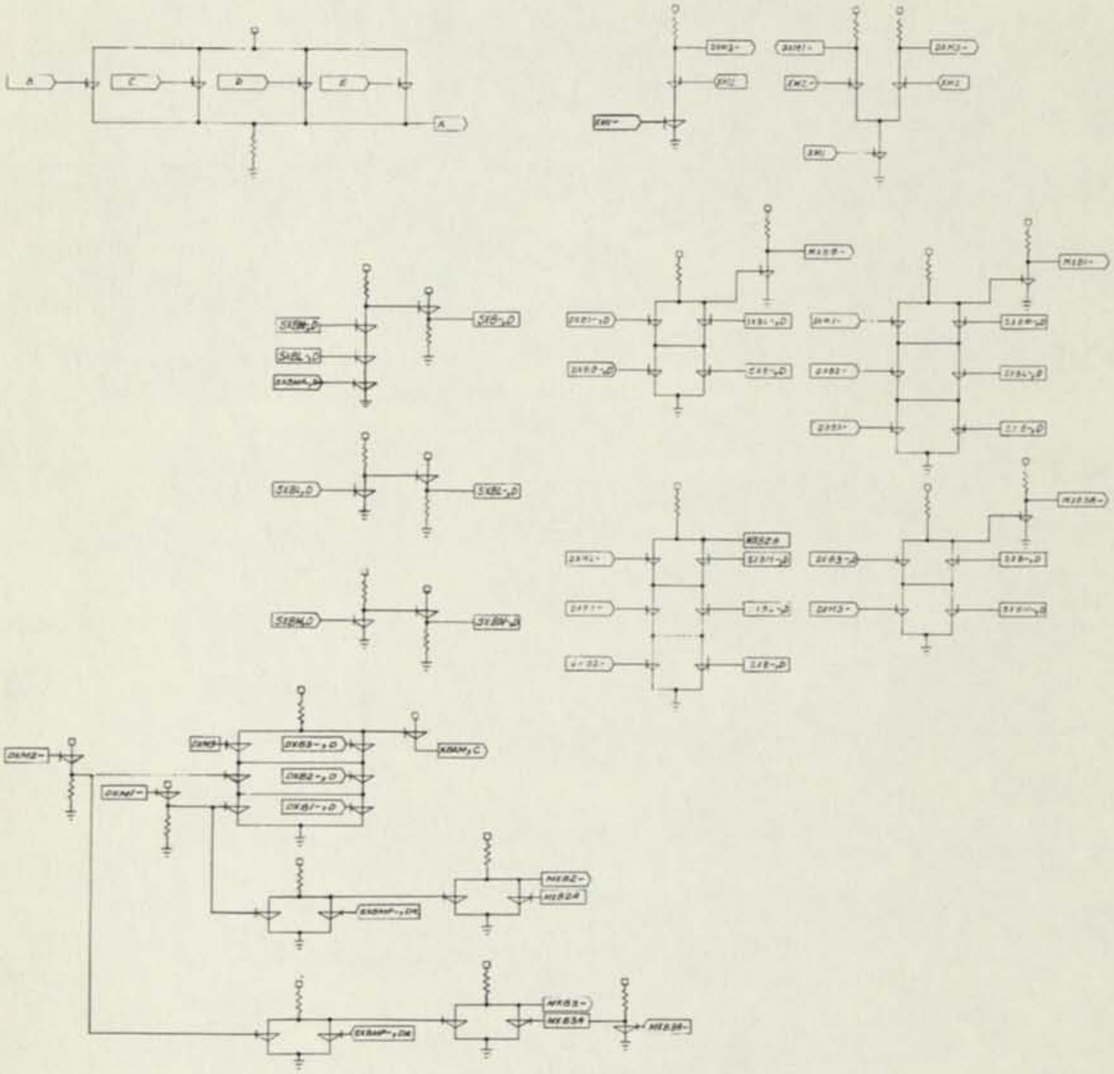
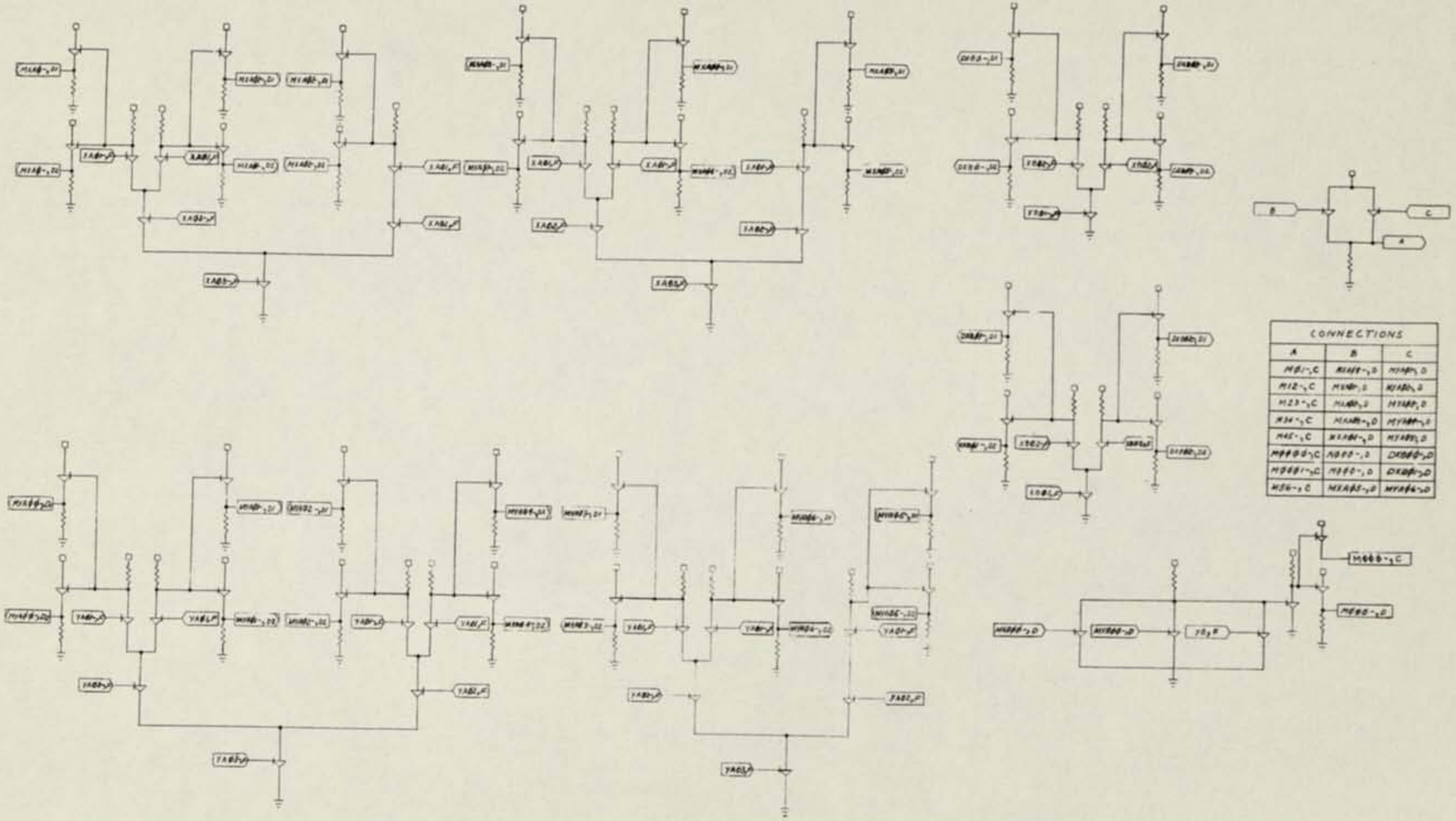


Figure 2-28. Memory Address Decoders (MXB, DXM, TL)



CONNECTIONS		
A	B	C
MØ1-,C	MXAØØ-,D	MYAØØ,D
M12-,C	MXAØ1,D	MYAØ1,D
M23-,C	MXAØ2,D	MYAØ2,D
M34-,C	MXAØ3-,D	MYAØ3-,D
M45-,C	MXAØ4-,D	MYAØ4,D
MØØØØ-,C	MØØØ-,D	DXBØØ-,D
MØØØ1-,C	MØØØ-,D	DXBØ1-,D
M56-,C	MXAØ5-,D	MYAØ6-,D

Figure 2-29. Memory Address Decoders (XA, YA, Special)

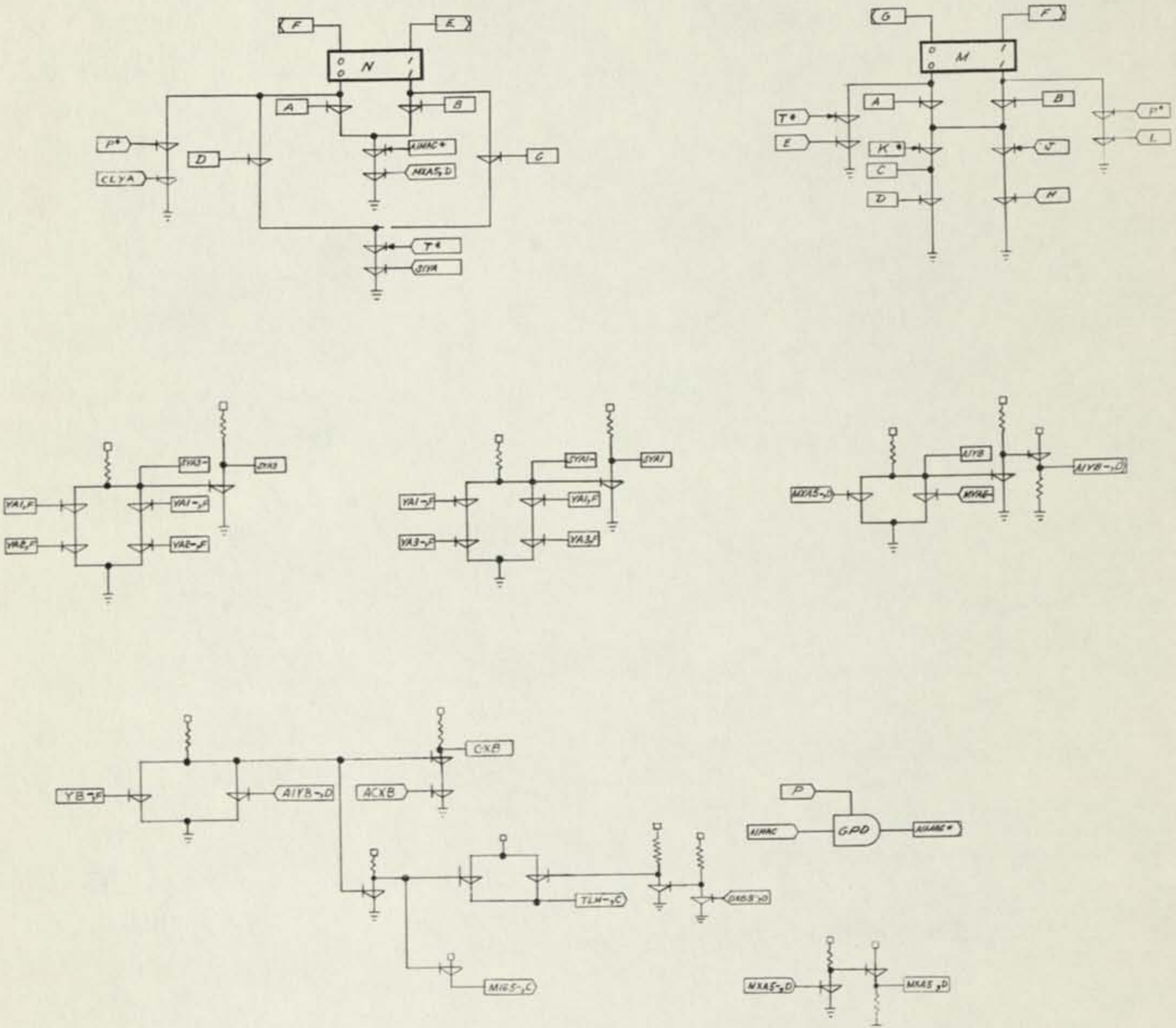
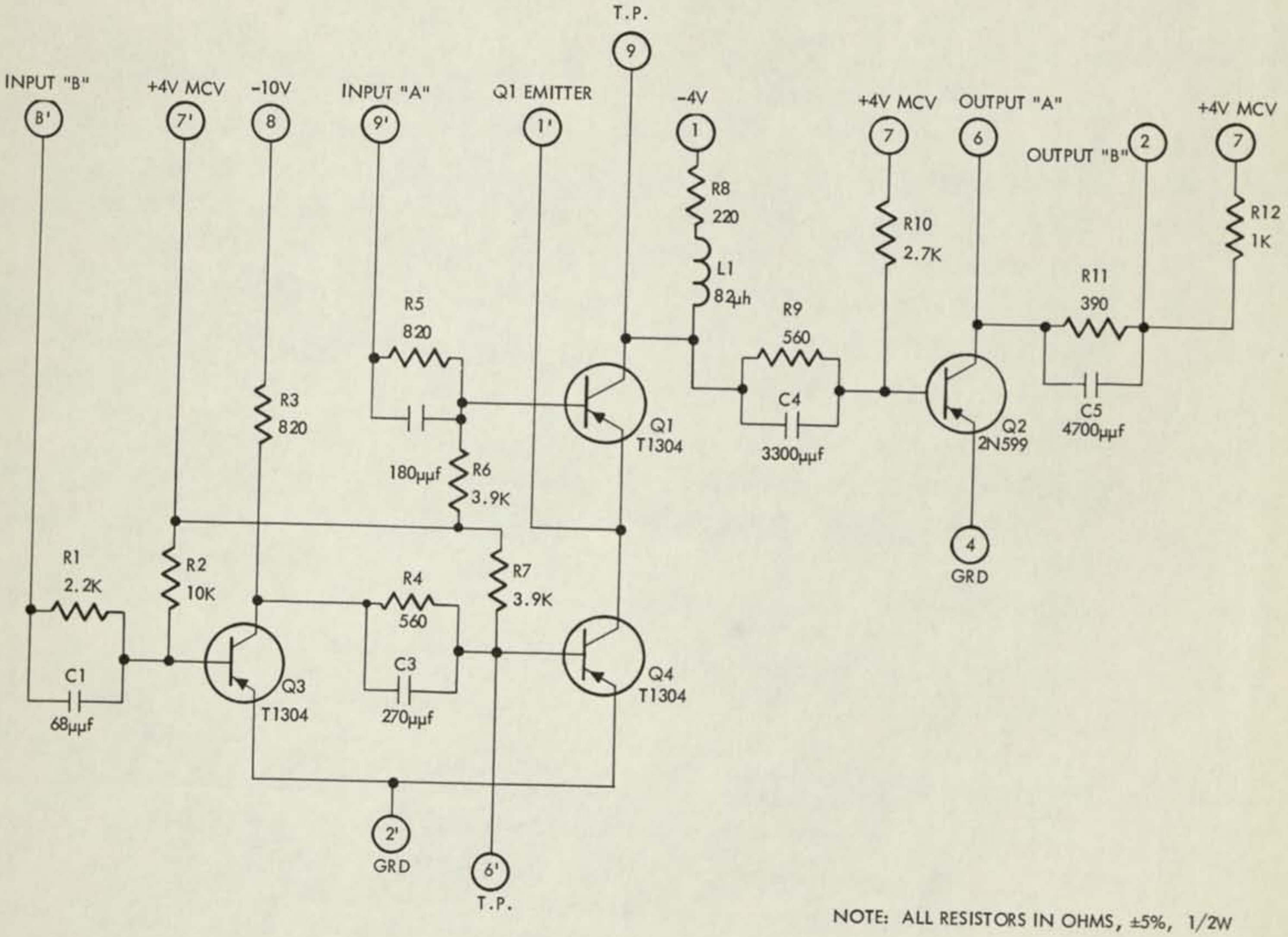


Figure 2-30. Memory Address Counter



NOTE: ALL RESISTORS IN OHMS, ±5%, 1/2W

Figure 2-31. Schematic Diagram, Element No. 92

2.7.1.4 Z-Drivers

The MOBIDIC B Z-driver package (type 609) had to be slightly modified for use in the OLCU memory system. This modification consists of a change in the inhibit winding circuits. Figure 2-32 shows the modification required in the MOBIDIC B Z-driver package for use in the OLCU core memory system.

The output current rise and fall time requirements for operation of the smaller and slower OLCU memory are not as stringent as those for the larger and faster MOBIDIC B memory. Therefore, the 4700 mmf capacitor used to reduce the output current rise and fall times in MOBIDIC B has been omitted in the OLCU. The drive current is of the half-read polarity and it is used to prevent the writing of a binary ONE during the write portion of the memory cycle.

The 78.7 ohm resistors were changed to 157 ohm resistors. These resistor values were changed because less current is required for the 336 cores per memory planes used in the OLCU than is required for the 4096 cores per matrix used in MOBIDIC B. A parallel resistor combination is used to increase the wattage dissipation of the network.

2.7.1.5 -14 Volt Sink

The same type of -14 volt sink package (type 611) used in MOBIDIC B is used in the OLCU. Two of the output transistors and their collector resistors have been removed, however. Elimination of these components is possible because the -14 volt sink is associated with only two current sources (2 base loads) in the OLCU rather than thirty-two current sources (32 base loads) in MOBIDIC B. Thus, the drive required from the OLCU -14V sink is very small compared with the requirement of MOBIDIC B.

2.7.1.6 Current Sources

The same type of current source used in MOBIDIC B is used in the OLCU. The type-600 package is used. It has been modified, however, by the removal of the components for two current sources. This modification was made since the package as used in the OLCU must provide only two current sources instead of the four required for use in MOBIDIC B.

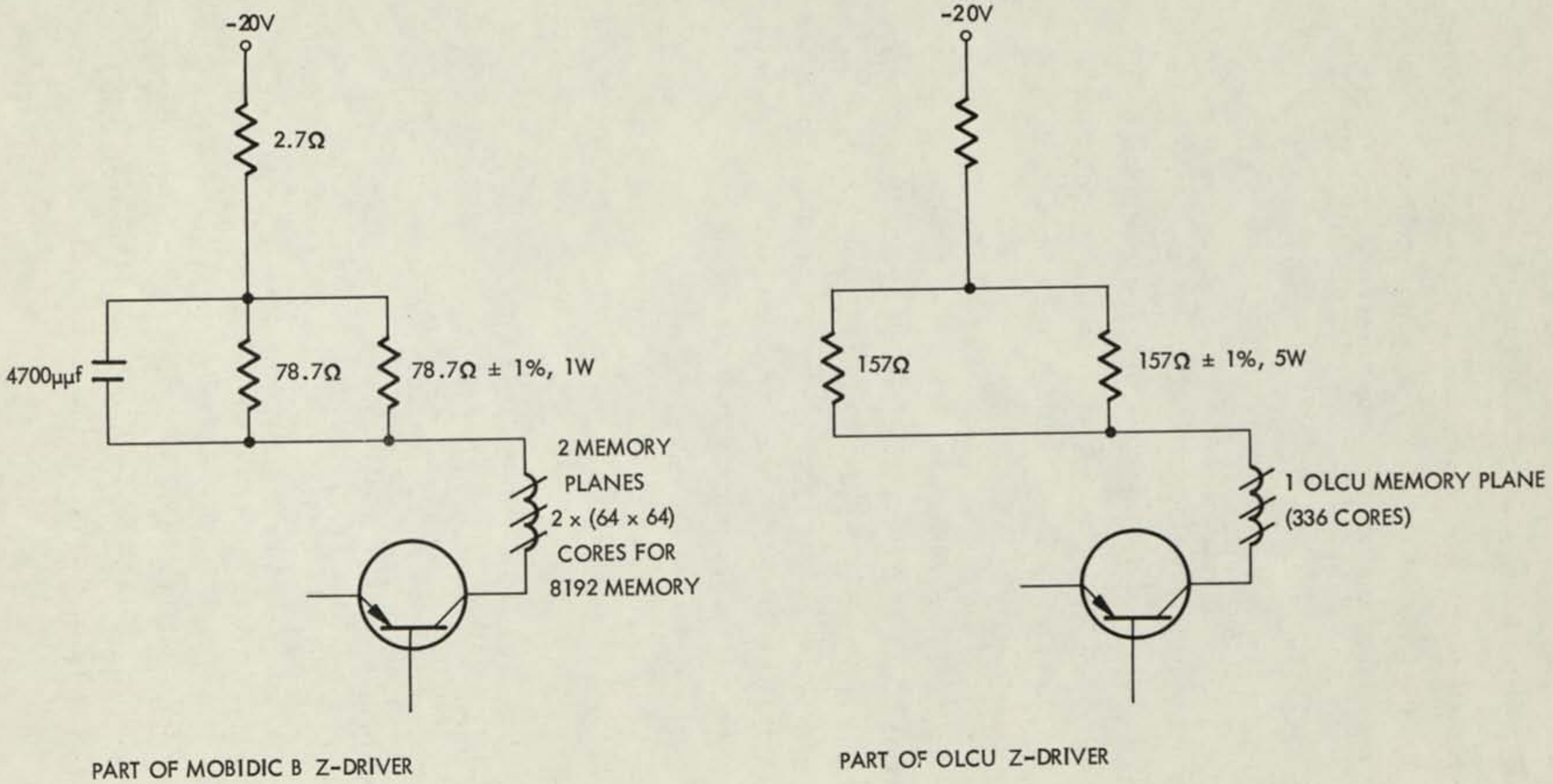


Figure 2-32. MOBIDIC B Z-Driver Package Modification

2.7.1.7 Sense Amplifiers and Voltage Reference

No circuit change or package modification is necessary to make the present MOBIDIC package adaptable to the OLCU memory system. Two sense amplifiers are on a type-600 package and one voltage reference circuit is on a type-607 package.

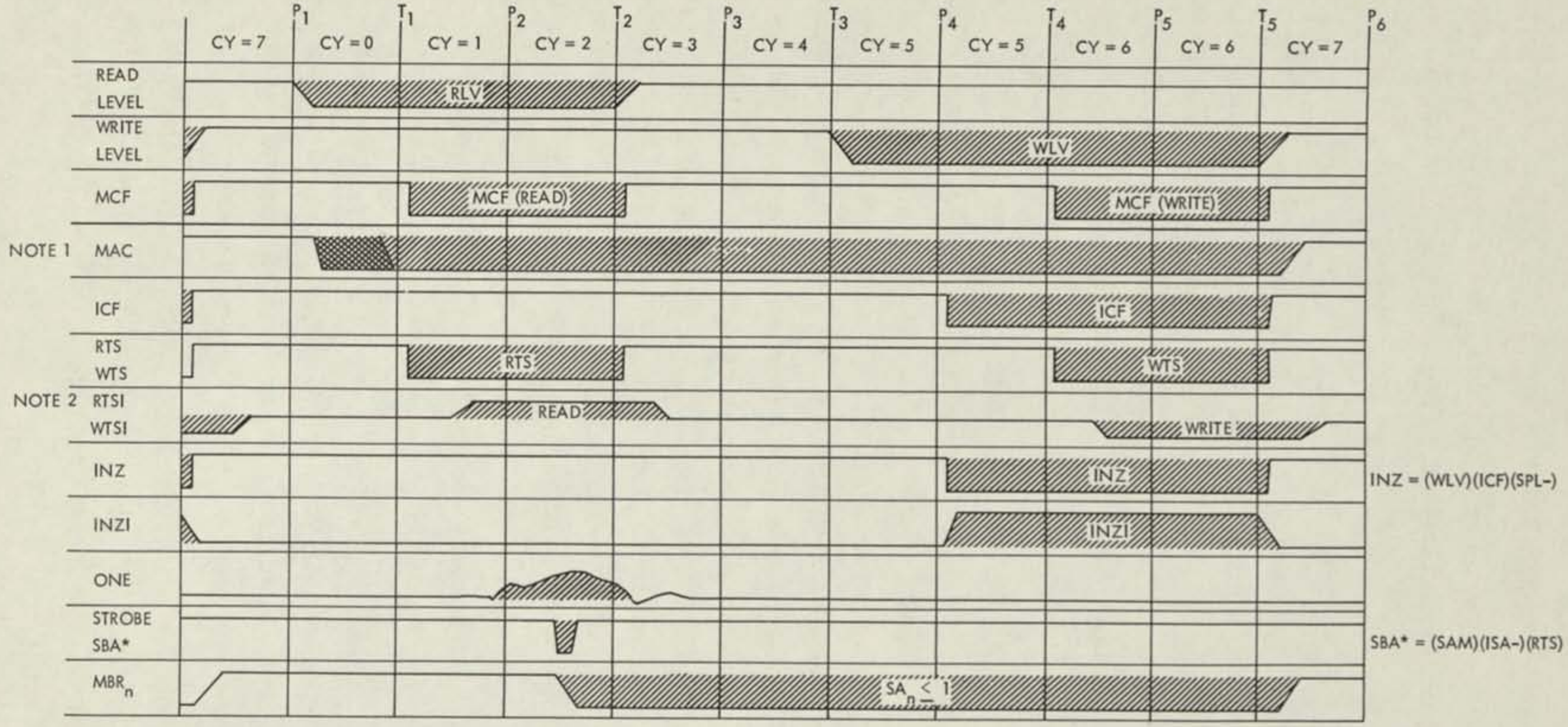
2.7.1.8 System Timing

The timing requirements of the OLCU are such that a controlled read-write cycle must be executed during a ten-microsecond interval. Level inputs are used along with t and p pulses to guarantee the setting of flip-flops which in turn generate timing levels. Thus a coincident current is diverted through the desired memory location at the desired time.

A ten-microsecond memory cycle was determined upon for the OLCU memory system. This cycle time is more than adequate for the purposes for which the OLCU was intended and it permits both conservative and economical design of the memory system. It has been found, in using the basic MOBIDIC B type memory circuits, that a minimum timing level of 1.25 microseconds duration is required to turn the core matrix switches on and allow enough current to switch the coincident cores completely. It is easiest, for OLCU purposes, to make this level of 2 microseconds duration. This permits raising of the level on one clock pulse and terminating it on another clock pulse. This technique has been extended so that all levels which control memory circuits are started and terminated by clock pulses. As a result, all signals which drive the memory during continuous operation remain high for lengths of time measured in multiples of one microsecond. Single pulse operation, of course, is an exception since clock pulses are not generated repetitively.

Figure 2-33 shows a 10-microsecond OLCU memory timing cycle. The following description is offered as an aid in understanding one cycle of this timing diagram.

The read (RLV) and write (WLV) timing levels are generated from the (Ø-7) states of the CY counter. States Ø, 1, and 2 constitute the three-microsecond read level while states 5 and 6 make up the four-microsecond write level. States



NOTE 1
DOUBLE CROSS HATCHED AREA
MAY BE DELAYED 0.75μsec

NOTE 2
READ AND WRITE CURRENTS
ARE DELAYED 300-400 mμsec
DUE TO TURN OFF OF N.C. AND
N.O. SWITCHES

Figure 2-33. OLCU Memory Timing Diagram

3 and 4 occur during the interval between the read and write currents, while state 7 occurs at the end of the read-write cycle. The read level starts on T1 and ends on T2; the write level starts on T3 and ends on T5 during the first ten microsecond read-write cycle.

The memory address counter (MAC) refers to the various flip-flops and decoders that select the word in memory to be read out or written into the core memory. This level comes up on the first pulse of the cycle, has a maximum delay of ∅.75 microseconds (shown in the double cross-hatched area in Figure 2-33), due to the transistor logic delays, and lasts for approximately ten microseconds before returning to zero on the last p-pulse of the ten microsecond read-write cycle.

When not in single pulse operation, the memory control flip-flop is set to 1 on the first t-pulse after the read level is generated and is high for two microseconds or until the next t-pulse, since MCF =) ∅ when RLV = ∅. The read timing level (RTS) is generated at the same time as the MCF and RLV levels and, therefore, lasts for two microseconds, RTS = (RLV)(MCF). As soon as the read timing level is present, the two normally-closed switches of the memory address system are turned off (opened) and four normally-open switches are closed, allowing an X- and Y-drive line to be pulsed simultaneously with half read currents (25∅ ma each) in such a direction that coincident cores receive aiding pulses. The other 37 cores (14 + 24 - 1) of each matrix are not switched since they receive only one half read current. The coincident current causes the selected cores, that are in the ONE state, to switch to the ZERO state and in so doing produce an output voltage. Cores in the ZERO state are not switched and produce no voltage output in the sense windings.

A binary one sense amplifier output is shown in the memory timing diagram, Figure 2-33. At the peak of this output voltage, a strobe pulse (SBA*) is generated. As a result of a ONE being read from the seventeenth or strobe matrix, time discriminating is produced among the other sixteen sense amplifiers since its output, SBA*, is used for the sense amplifier timing input. The output of each sense amplifier is directly connected to the "set to 1" input of its respective memory buffer register (MBR), thereby producing the required memory word in the form of binary ONEs and ZEROs.

At T2, the cycle counter changes from state 2 to state 3 while the read-level memory control flip-flop and read-timing signals return to the zero state. The read current is turned off approximately 3ØØ to 4ØØ milliseconds later. This delay is a result of turning-off of the normally-open and turning-on of the normally-closed switches. This is caused by the absence of the RTS (read timing level) signal.

A quiescent condition exists in the core memory during the cycle counter states three and four. During this period, operations such as shift left in IBR, MBR or OBR and transfers between IBR and MBR or between MBR and OBR are performed. Also, checks are made, such as comparing IBR with MBR.

After read-out of information is complete, a write-in cycle is performed starting at T3. A write level (WLV) is generated during states 5 and 6 of the cycle counter. Since each of these states is 2 microseconds long, the write level lasts for 4 microseconds, coming on at T3 and remaining at a ONE level until T5.

The inhibit control flip-flop (ICF) is set to ONE on the first p-pulse after the write level is at ONE. (ICF =) 1 when WLV = 1 and p-pulse.) When the ICF flip-flop is set to ONE and the system is not in single pulse operation, the INZ or inhibit level comes up. This INZ level is applied to one of the inputs of each Z-driver and is generated only during writing since this is the only time when it is desired to inhibit or prevent writing a ONE in the core memory word that has been selected. INZ = (WLV)(ICF)(MCF + SPL-) is the Boolean expression for INZ. A second input to the Z-drivers is the V/temperature input. This temperature-compensated voltage source employs a thermistor network which is located in the immediate vicinity of the core memory planes. The purpose of this network is to vary the inhibit drive current in accordance with the temperature surrounding the memory planes. The third input to each Z-driver is the memory buffer register (MBR) input. This input to each Z-driver is directly connected to the prime, or ZERO, side of its respective MBR flip-flop, thereby inhibiting or preventing the writing of a ONE back into the corresponding bit position of the core memory word. The inhibit level is of the same polarity as the read level, and, because of the slow rise time of the Z-driver, is turned on one microsecond before the write drive current.

On the following t-pulse (T4), the MCF flip-flop is set to ONE, and as a result, the write timing level (WTS) is turned on. This level turns off the two normally-closed switches and turns on four normally-open switches. Since the same memory address is present, the same X-drive and Y-drive lines are selected. However, due to the switching matrix, the write current pulses are diverted through the X-Y drive lines in the opposite direction to the read current. Therefore, all memory word cores are pulsed with two, half-write current pulses (25Ø ma each), both in an aiding direction, which automatically rewrites a binary ONE into every matrix. The matrices in which a ZERO is required also receives an inhibit or half-read current pulse, thereby receiving a net half-write current pulse which causes them to remain in the ZERO state. Figure 2-26 shows a typical X- or Y-drive line selection configuration.

At T5, state 6 of the cycle counter returns to zero. This immediately returns the write level, the memory control flip-flop, and the inhibit control flip-flop to the ZERO state. Therefore, since the writing and inhibiting conditions are no longer present, the writing and inhibiting currents are turned off. The normally-open switches are turned on, thereby shorting the current sources to ground. The memory address counter also returns to ZERO while the writing and inhibiting currents are delayed 3ØØ to 4ØØ milliseconds, due to turning off of the normally-open and turning of the normally-closed switches, before returning to zero. During cycle counter state a quiescent condition exists again in the core memory while the new address is selected and the memory address counter steps up one to count the number of read-write memory operations. At P6, as indicated on Figure 2-33, the next ten microsecond read-write cycle is commenced.

2.7.1.9 Single Pulse Operation

The OLCU uses a single pulse (SPL) mode of operation to aid in troubleshooting. Since the normal inhibit is started on a p-pulse and remains on until the second subsequent t-pulse, during single pulse operation the inhibit would be infinitely long as far as the capacitative coupling in the Z-drivers is concerned. It is poor practice to allow single pulse operation to cause the diodes in the X-Y coordinates to conduct at an effective 1ØØ% duty cycle. Therefore, in single pulse operation, precautions are taken to limit the diode duty cycle and to shut off the Z-drivers.

The output of strobe plane SA17 is applied to the timing input of the CSIP gated pulse drivers. This CSIP pulse is used to clear both the memory control and inhibit control flip-flops. When not in single pulse operation, this GPD is held off because of the presence of not-single-pulse (SPL-) level. During single pulse operation, the CSIP pulse is generated when the memory control flip-flop and SA17 outputs are present. The CSIP pulse is generated to guarantee that the drive current is turned off within a few milliseconds after a binary ONE is sensed from the strobe plane, during both the read and write memory cycle operations. This differs from MOBIDIC B in that the strobe is used to turn off both the read and write, rather than just the read drive currents during single pulse operations. The inhibit control flip-flop is also cleared by the CSIP pulse, thereby turning off the half-read inhibit current at the same time as the write current is turned off.

Figure 2-34 illustrates the single pulse timing cycle. While in single pulse operation, the memory control flip-flop is set during the read cycle on the first p-pulse, rather than the first t-pulse after the read level is present. This means that the read drive current is of shorter duration, approximately one microsecond, because it is cleared by the CSIP pulse. Although this drive current duration is shorter than normal, it is still long enough to switch the memory cores completely.

During single pulse operations for the write cycle, the memory control flip-flop is not set until the second p-pulse after the write level is generated. This again means that the write current is of shorter duration due to the late turn on. The inhibit control flip-flop is turned on at the same time for both single pulse and normal operations. However, during single pulse operations the inhibit level (INZ) is held off until the memory control flip-flop is set. In other words, the inhibit current is turned on during the same interval that the write current is on since both of these currents are controlled by the memory control flip-flop during single pulse operation.

There is some absence of overlap of the inhibit and write currents because of the long rise time of the inhibit driver, but not enough to write a binary ONE in a ZERO location. Furthermore, the CSIP pulse clears both the inhibit and memory control flip-flops, thereby turning off the respective drive currents a few milliseconds after a ONE is sensed in the strobe plane. The binary ONE sensed in the

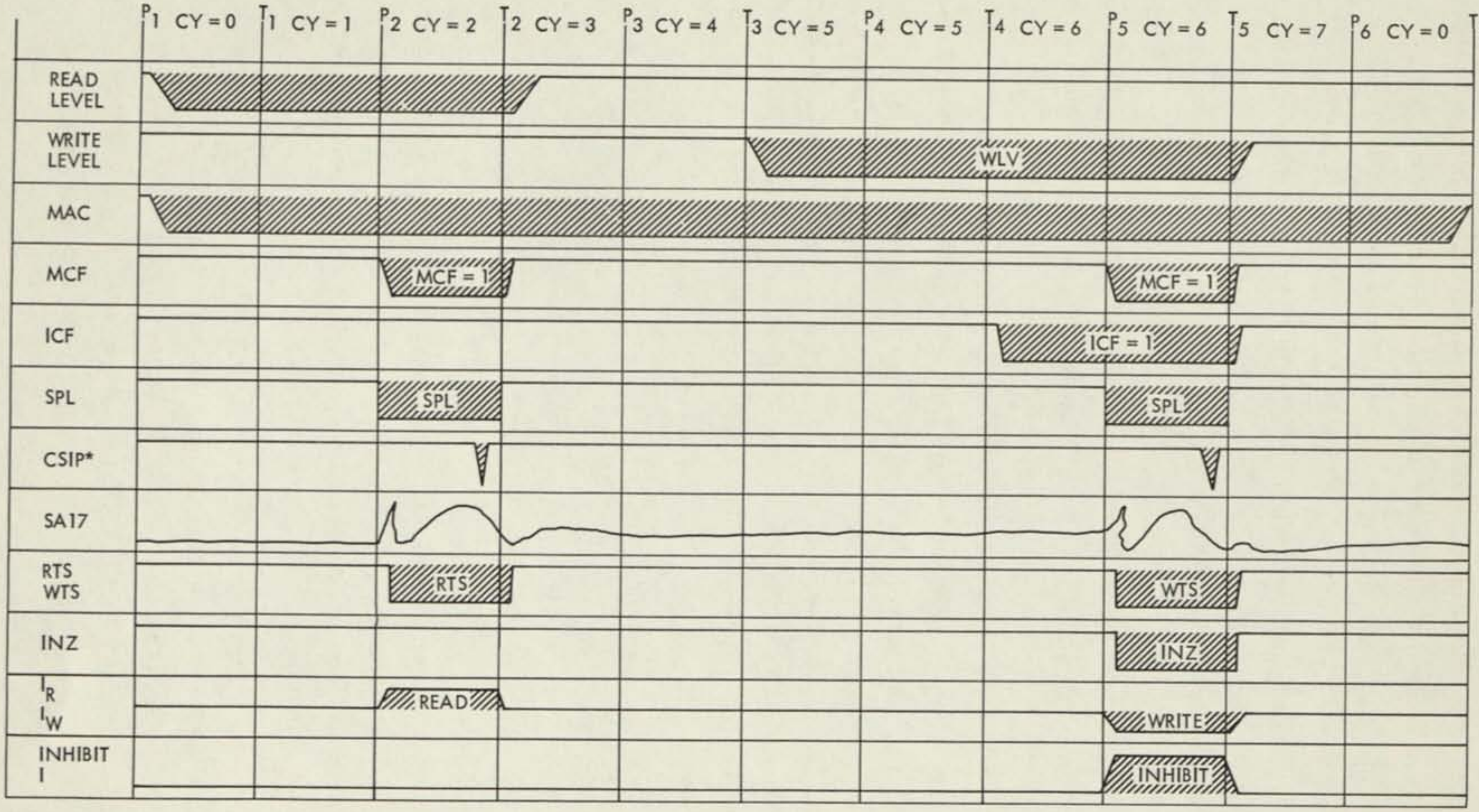


Figure 2-34. Single Pulse Timing Cycle

strobe plane during the write cycle fails to generate the unwanted strobe pulse (SBA*) due to the absence of the read timing signal (RTS).

2.7.2 Console Switch Interlocks

In the OLCU a magnetic interlock is used to prevent more than one switch in a group being operative at the same time. A logic diagram of such a group of switches is illustrated in Figure 2-35. L1, L2, L3, L4 and LN are magnetic "hold" coils mounted on switches 1 through N. The coils, when energized by the operation of their respective switches, hold the switch closed until the coil is de-energized. On each switch there are mounted auxiliary contacts, one normally-open contact and one or more normally-closed contacts to control energizing or de-energizing the hold coil. Assume switch 2 is pressed. The normally-open contact on switch 2 closes and energizes hold coil L2 by providing a circuit from +24 volts to ground. At the same time, the two normally-closed contacts open the circuit to coil from ground and the circuit from +24 volts to coils L3 through LN. If switch 3 is pressed, the normally-closed contact on this switch opens the ground circuit to coil L2 which opens switch 2. At the same time, the normally-open contact on switch 3 closes, energizing coil L3, and the second normally-closed switch opens the circuit from +24 volts to coils L3 through LN. In a similar manner, any switch pressed interrupts the circuit between +24 volts and ground through all other switch hold coils and energizes just the hold coil on the switch pressed. It is evident after tracing this diagram that only one switch can be activated at a time and pressing another switch button releases the switch active at that time.

The register setting keys are also held in contact magnetically, as shown on Figure 2-36. Any one or more of the switches may be operated at the same time and the auxiliary normally-open switches will energize their respective hold coils. A normally-closed pushbutton switch in series with the ground return from all coils serves as a reset switch for all coils.

The advantages of using magnetic hold and interlock switches introduce a higher reliability level than mechanical holds and interlocks and permit faster setting and resetting of register keys. A much greater number of keys can be

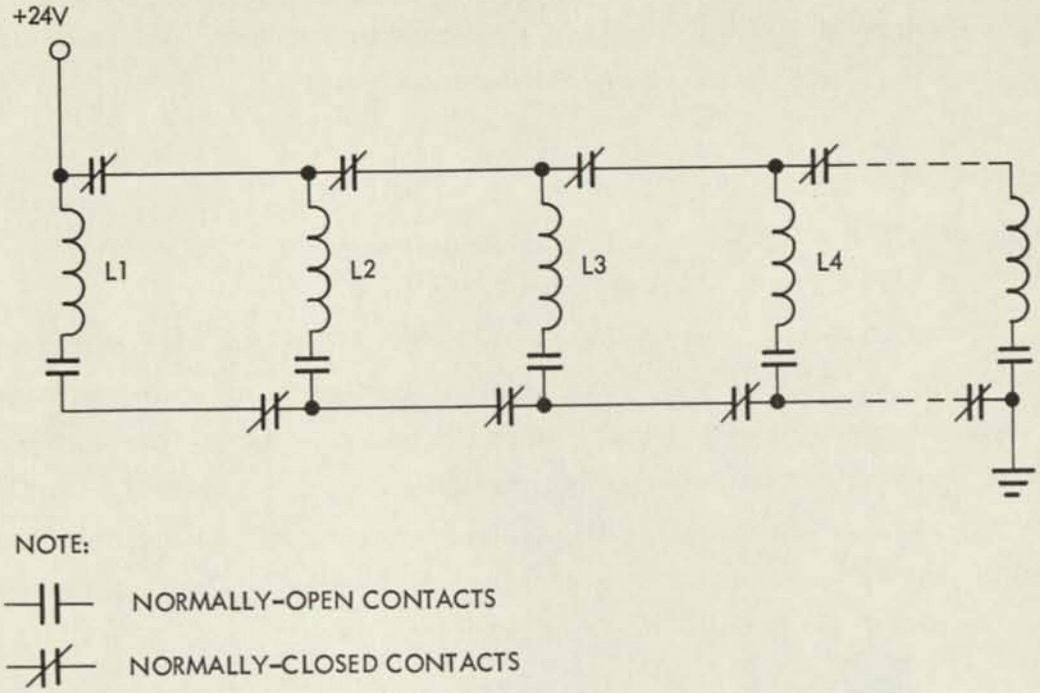


Figure 2-35. Magnetic Switch Interlock Logic Diagram

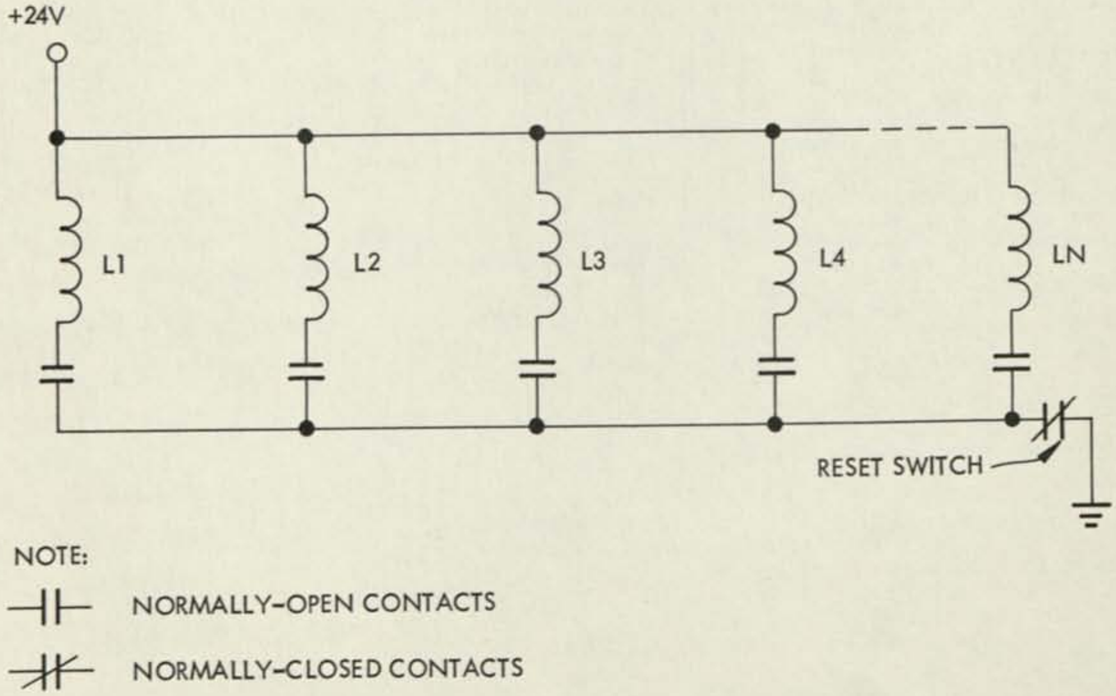


Figure 2-36. Register Selection Key Logic Diagram

reset simultaneously magnetically than by mechanical means. An indicator lamp built into each switch indicates which coil is energized.

2.7.3 Half Register Driver

2.7.3.1 General

Many OLCU signals must drive from 12 to 25 base loads. In the interests of economy a half register driver has been designed to provide this driving capacity. This avoids uneconomical use of the full register driver with its increased load-driving capacity. The half register driver circuit is similar to the full register driver circuit. The output, however, is driven from a single cascode circuit rather than from two such circuits as is the arrangement in the full register driver.

2.7.3.2 Theory of Operation

The half-register driver, as shown in Figure 2-37, is comprised of three sections: a) cascode output stage, b) power-inverter amplifier, and c) input inverter. The power inverter amplifier, Q2, is designed such that it will supply sufficient drive to the emitter follower, Q3, even under worst-case loading conditions (25 base loads). The input inverter, Q1, which drives the power inverter, Q2, and the inverter portion of the cascode, Q4, is required to amplify the input signal and to provide the proper signal sense to the output cascode.

When the input inverter, Q1, is turned on with a negative input level, the cascode inverter and power are both turned off. The off voltage of the power inverter collector is determined by the clamping action of the base-collector diode of Q3 and is typically -4.5 volts. Therefore, the cascoded emitter follower, Q3, is turned on with its base collector diode forward biased by means of a negative input level to Q1.

With a ground level input it is evident that the cascode inverter, Q4, is turned on and the emitter follower turned off. The bi-directional drive provided by the cascode output circuit is an excellent means of charging and discharging load capacitance and is more efficient than a simple emitter follower.

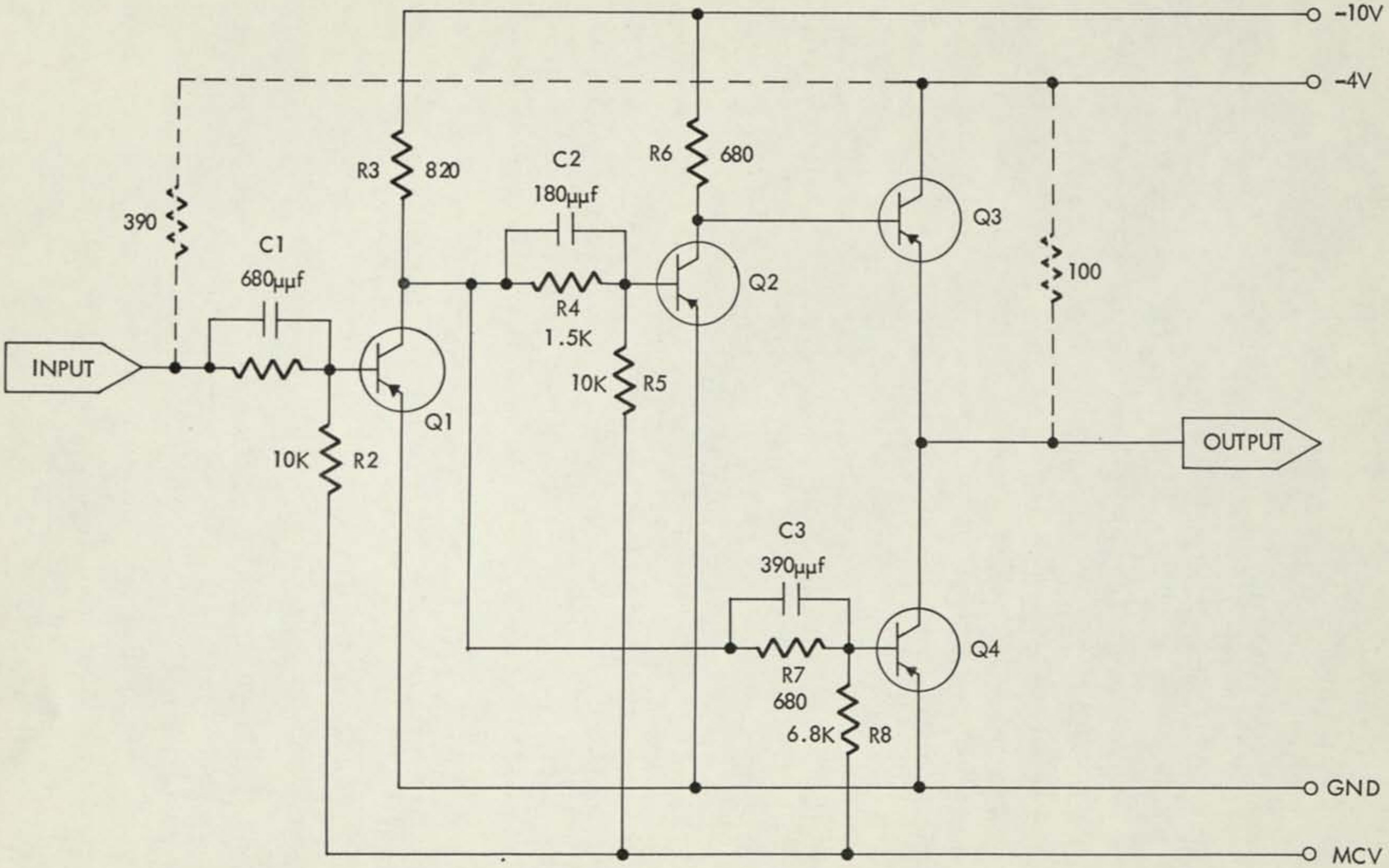


Figure 2-37. Half-Register Driver

2.7.3.3 Circuit Specifications

2.7.3.3.1 Function

The circuit function is to supply binary level loads to meet the following requirements:

a) Output Sense:	Same as input
b) Input Load:	One standard base
c) Output Load:	25 standard bases (maximum)
d) Delay Allowance:	15Ø microseconds

2.7.3.3.2 Special Wiring Rules

Certain special requirements must be met in wiring the half register drivers which are listed as follows:

a) To reduce crosstalk, the lead from the driver must be coaxial cable if it is over six (6) inches long, including the etch.

b) Short cables (less than four feet) may be "teed" off a longer one.

c) The output line should be terminated at the furthest point with 1ØØ ohms to -4 volts.

2.7.3.3.3 Operating Temperature Range

The required operating temperature range extends from minus 3Ø degrees to plus 65 degrees centigrade.

2.7.3.3.4 Power Requirements

The power requirements are listed below:

DC Supply	Maximum Current
-1ØV	12.8 Ma
-4V	49.Ø Ma
-4.4V	2.Ø Ma

2.7.3.3.5 Marginal Checking

The +4 volt supply serves for marginal checking of the half register driver. A typical circuit with typical transistors will fail at +16.5 volts and -2.4 volts. However, the upper marginal test voltage should be limited to +12 volts to preclude the possibility of damaging the cascode output transistors. This possibility exists if the upper marginal limit of Q4 exceeds that of Q2 when the output is Ø volts, since Q3 and Q4 would be shorted to -4 volts.

With low beta and high I_{co} transistors, the circuit will fail at approximately +10 volts and -Ø.5 volts, respectively.

2.7.3.4 Worst-Case Considerations

2.7.3.4.1 Cascode Output Power Dissipation

The major problem with a cascode output is that during commutation Q3 and Q4 are short-circuited to -4 volts for a small period of time. During this interval, peak currents of 13Ø milliamperes have been measured, and although the duration of the commutating transient is only on the order of 1ØØ to 16Ø microseconds, the peak power incurred is such that it adds appreciably to the average collector dissipation at the higher frequencies.

It was determined that the worst-case transient dissipation occurred with a fast-switching transistor for the inverter and a slow-switching transistor for the emitter follower of the cascode.

A group of transistors were tested for beta at Vc = 1 volt, I_c = 40 ma, and f = 5 mc. A maximum beta of 3Ø and a minimum beta of 1Ø were found and employed in the cascode output. The results of this test are tabulated below:

TABLE 2-10. TRANSIENT DISSIPATION IN Q3 AND Q4

	(Ø to -4)	(-4 to Ø)	$P_1\left\{\begin{matrix}T_1\\ \emptyset\end{matrix}\right.$	$P_2\left\{\begin{matrix}T_2\\ \emptyset\end{matrix}\right.$	P_{aver}	P_{aver}
	T_1	T_2			1 mc	5ØØ kc
INV	16Ø mμs	1ØØmμs	16.7 mw	1Ø5 mw	17.25 mw	11.23 mw
EF	16Ø mμs	1ØØmμs	67.5 mw	48 mw	19.18 mw	12.2 mw

Figures 2-38 through 2-41 are graphs depicting some of the results of the breadboard tests.

2.7.3.4.2 Output Inverter

The primary function of the output inverter is to provide a low resistance path to ground to discharge the load capacitance. Q4, as shown on Figure 2-37, is driven by the input inverter, Q1, and is on when the input is a binary zero level. The output collector load for Q4 is the 1ØØ ohm termination resistor to -4 volts. Since the input inverter, Q1, drives both the power and output inverter, the minimum base on current (I_{bon}) to Q4 is:

$$I_{bon}\ \text{min} \simeq \frac{V_c Q1_{nom} - Vbe}{R7} - \frac{MCV(1.1) + Vbe}{R8(\emptyset.9)} = 4.5 \text{ ma}$$

Where $V_c Q1_{nom} = -3.9V$,

Maximum Collector Current $I_{c\ max} = \dfrac{4.\emptyset(1.1) - Vce}{1\emptyset\emptyset(\emptyset.9)}$ sat = 47.8 ma

Therefore, the maximum required circuit beta:

$$B_{ckt\ max} = \frac{I_{c\ max}}{I_{bon\ min}} - 1\emptyset.6$$

To minimize circuit delay and prevent over-dissipating, the cascode output circuit, C3, as shown on Figure 2-37,must be of sufficient magnitude to assure sweeping all excess minority carriers out of Q4's base region with minimum storage under worst case conditions.

If the maximum base on current into Q4 is 6.2 milliamperes and $K's_{max}$ = 16Ø mμsec., then the charge that must be swept out of the base region before Q4 can turn off is $I_{b\ max}$ (k's) = 990 μμ coulombs, assuming that the entire base current is excess (B = ØØ). Under conditions of maximum on drive, the voltage across C3 is 3.7 volts.

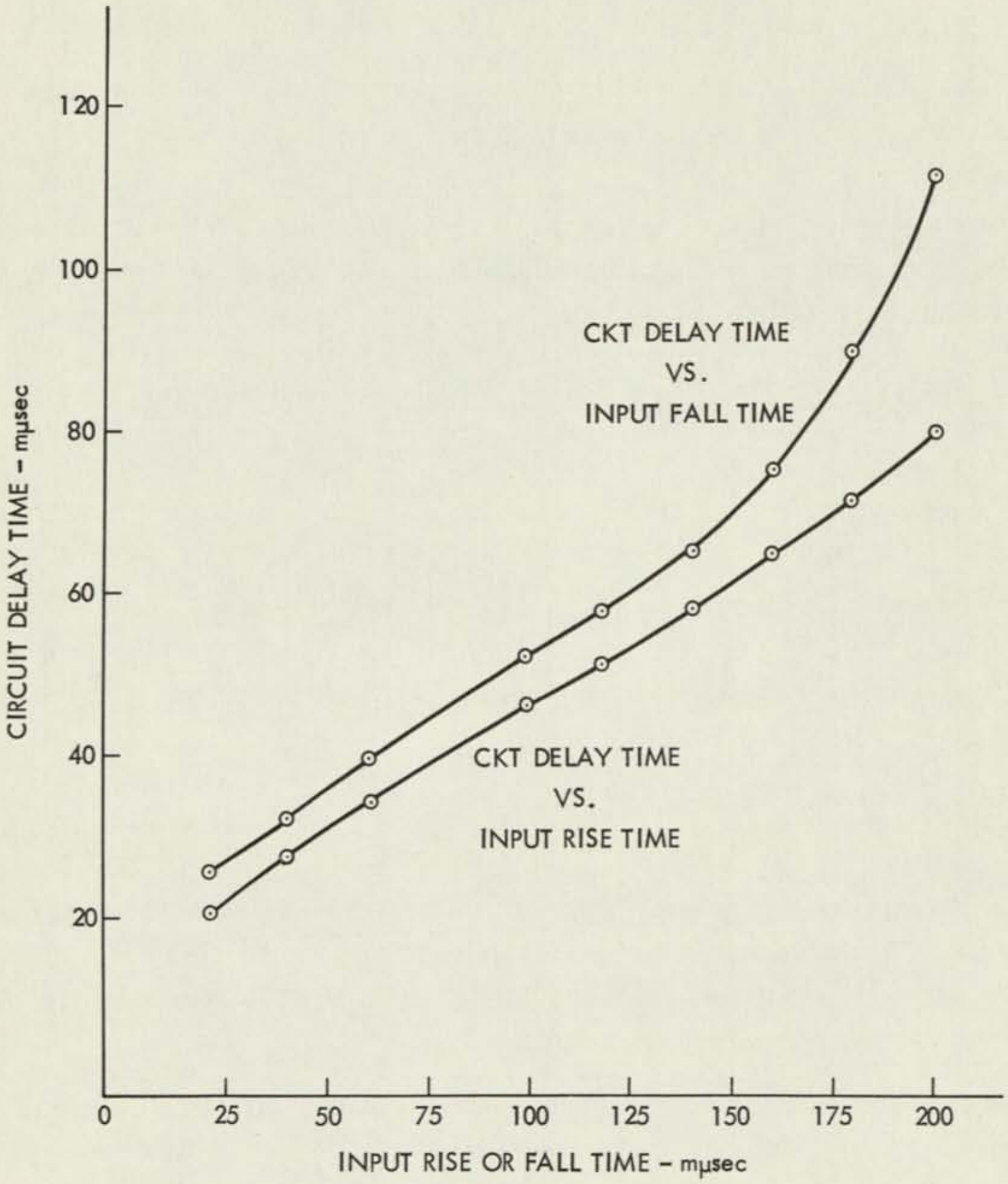


Figure 2-38. Circuit Delay Time vs Input Rise or Fall Time

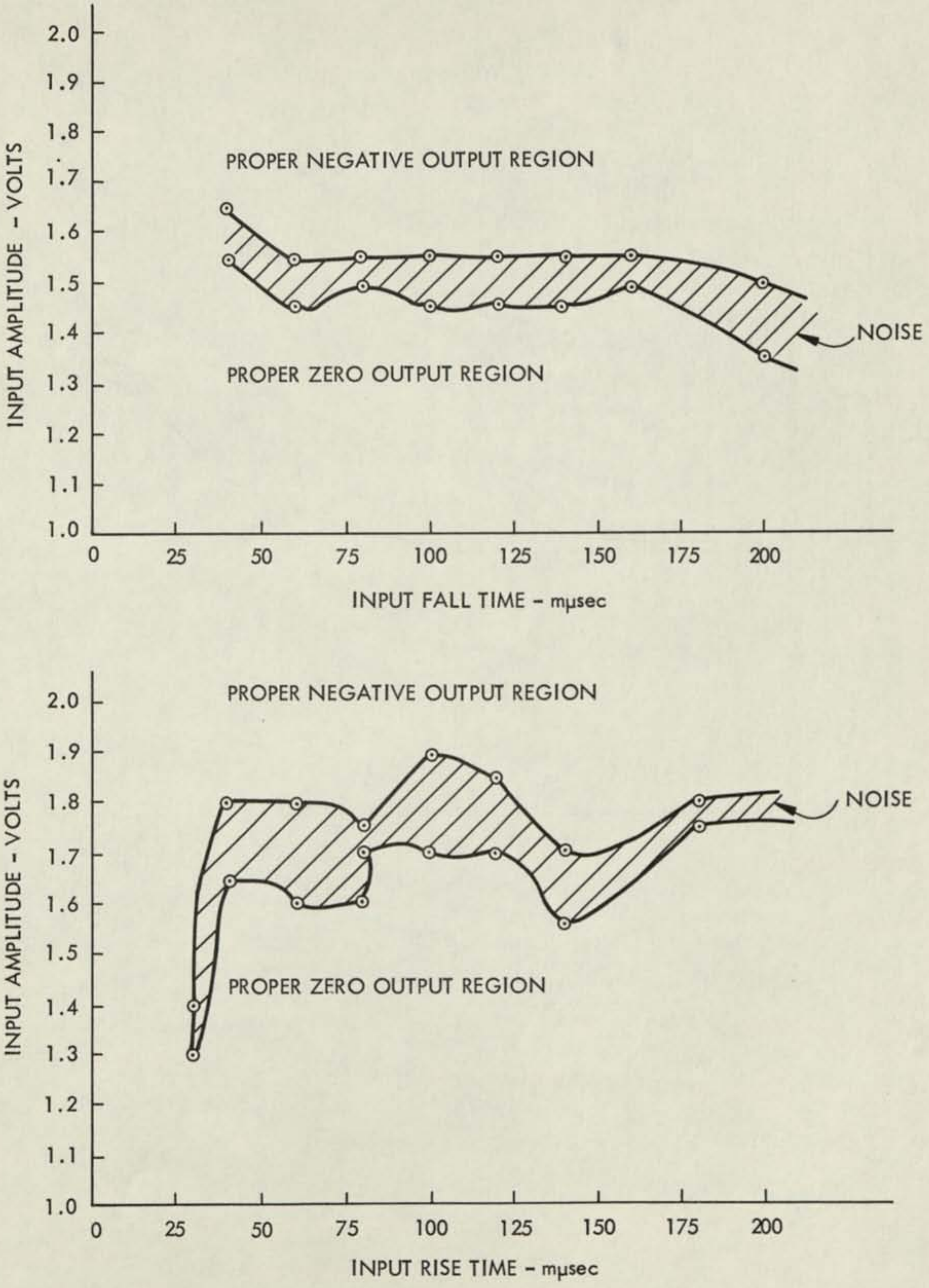


Figure 2-39. Input Amplitude vs Input Rise and Fall Time

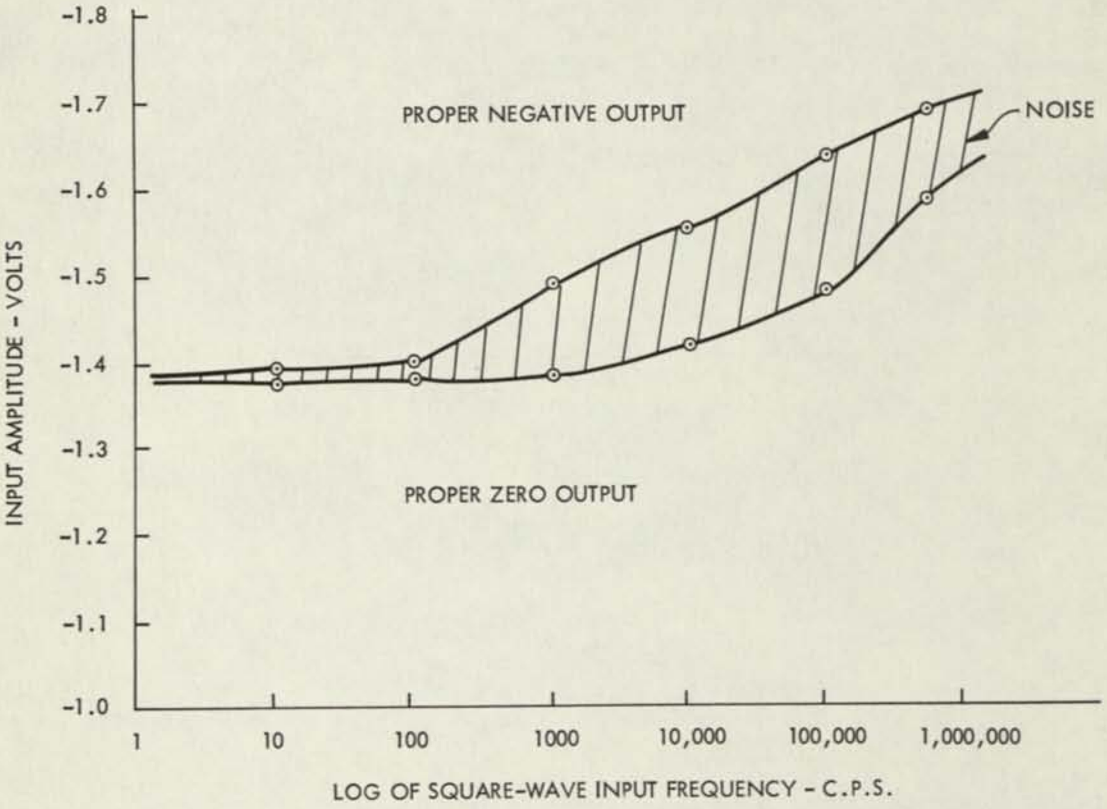


Figure 2-40. Input Amplitude vs Log of Input Frequency

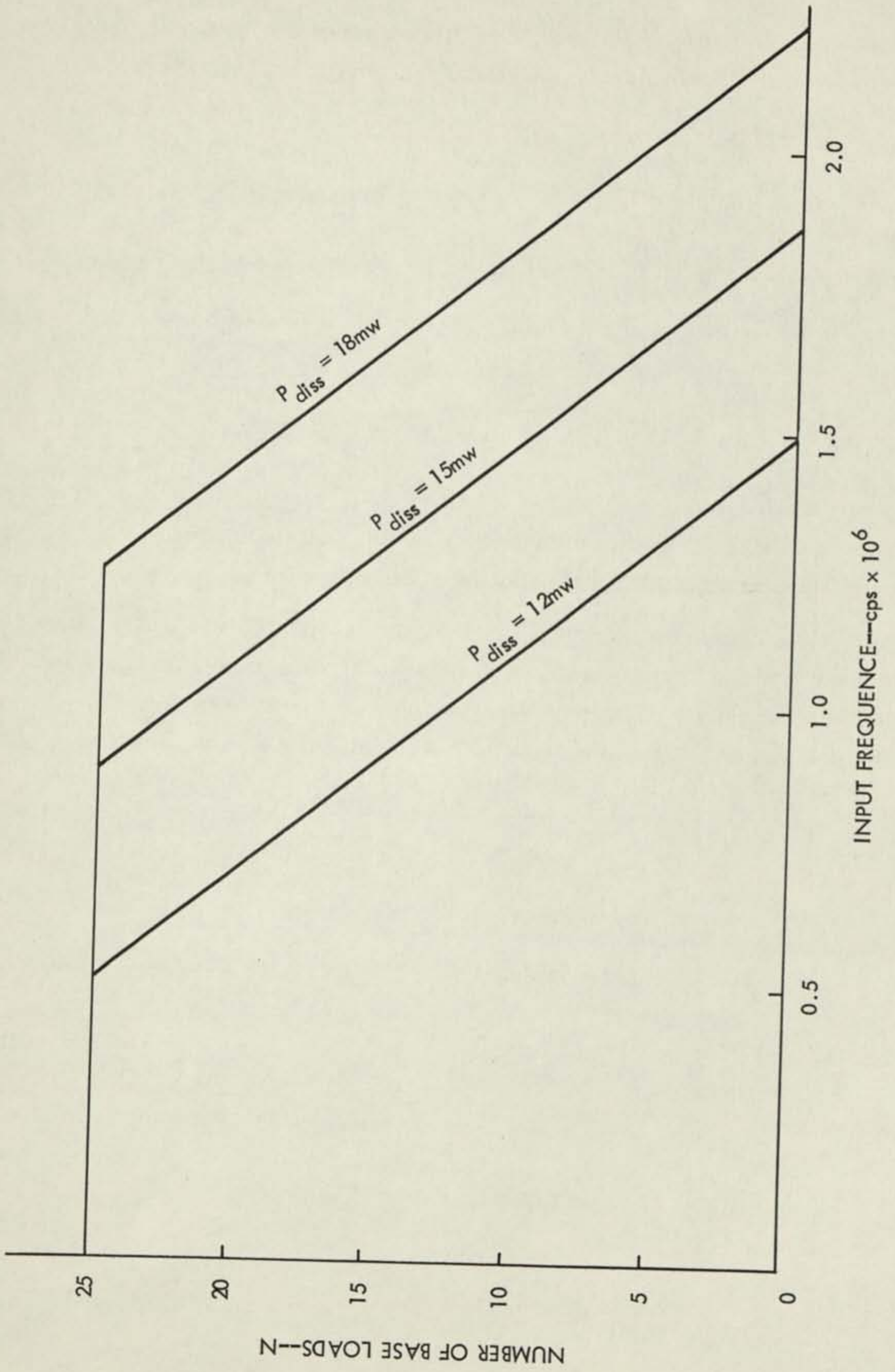


Figure 2-41. Recommended Load Derating vs Input Frequency

Assuming 1ØØ mμsec to sweep the entire charge out of Q4 and an average voltage across C3 of 2 volts during this time, the maximum charge taken by R7 is:

$$(i)(t) = \frac{(2)(1ØØ)(1Ø^{-12}}{(.68Ø)(.9)} = 328\ \mu\mu\ \text{coulombs}$$

Therefore, C3 must supply 33Ø + 99Ø = 132Ø μμ coulombs under its worst tolerance of 2Ø%

$$\text{or } Ø.8(C3) = \frac{132Ø\ \mu\mu\ \text{coulombs}}{3.7\ \text{volts}} = 357\ \mu\mu f,$$

and the nearest standard value = 39Ø μμf. When Q4 is turned on, the RC time constant is

$$\frac{(82Ø)(68Ø)}{82Ø + 68Ø}(39Ø \times 1Ø^{-12}) = Ø.145\ \mu\text{sec.}$$

Since the switch is on for a minimum of 1 μsec, $\frac{1.Ø}{0.145}$ = 6.9 time constants are allowed, thus assuring complete charge before the switch is turned off. Consequently, repetition rate sensitivity should not be a problem.

The average turn-on transient dissipation integrated over 1ØØ mμsec was found to be 1Ø5 mw. In turning off, there was no appreciable current surge due to "shorting" and the average power dissipation was found to be 16.7 mw integrated over 16Ø mμsec. The integrated dissipation over a complete period is then:

$$P_{\text{diss aver}} = \frac{P_{\text{on trans}} + P_{\text{on dc}} + P_{\text{off trans}} + P_{\text{off dc}}}{T}$$

a) For 1 mc

$P_{\text{on trans}}$	= (1ØØ)(1Ø5)	= 1Ø5ØØ mw - mμsec
$P_{\text{on dc}}$	= (5Ø)(Ø.2)(5ØØ - 1ØØ)	= 4ØØØ mw - mμsec
$P_{\text{off trans}}$	= (16.2)(16Ø)	= 26ØØ mw - mμsec
$P_{\text{off dc max}}$	= (4.4)(Ø.1)(5ØØ - 16Ø)	= 15Ø mw - mμsec
	TOTAL	= 1725Ø mw - mμsec

$$P_{\text{diss aver}} = \frac{1725Ø}{1ØØØ}$$

b) For 5ØØ kc

$$P_{aver} = \frac{(I_{co\ 5ØØ} + 9ØØØ + 26ØØ + 37Ø)\ \text{mw - m}\mu\text{sec}}{2ØØØ\ \text{m}\mu\text{sec}} = 11.23\ \text{mw}_{max}$$

The quiescent off stability of Q4 is dependent upon the ability of the off-bias source to supply the maximum I_{cbo}. Assuming a minimum base voltage of +Ø.Ø5 volts required to hold off Q4, the minimum circuit capability may be computed as follows:

$$Ico_{min\ ckt\ cap} = \frac{(Ø.9)(4.4) - Ø.Ø5}{(1.2)(6.8)} - \frac{Ø.Ø5 + Vce\ Q1max}{(Ø.68)(Ø.95)}$$

$$= 18Ø\ \mu a\ \text{min.}\ \ (\text{Max}\ I_{cbo}\ TI3Ø4 = 8\ \mu a)$$

2.7.3.4.3 Emitter Follower

The emitter follower worst-case loading is illustrated in Figure 2-42. The on drive to the emitter follower, Q3, is provided through R6, as shown on Figure 2-37. When the power inverter, Q2, turns off, its collector is clamped to approximately -4.5 volts maximum. Therefore, the minimum on-drive into the base of the emitter follower is as follows:

$$I_{b\ E.F.\ min} = \frac{1Ø(Ø.9) - 4.5}{(Ø.68Ø)(1.1)} = 6.3\ \text{ma}$$

The worst-case load the emitter follower sees is 25 base loads. If the maximum worst-case emitter current is:

$$I_{E.F.\ max} = \frac{(4.4 - V_{ce\ min}) - V_{be\ min}}{\frac{(2.2)(Ø.9)}{25}} = 5Ø\ \text{ma}$$

$$B_{E.F.\ ckt\ max} = \frac{I_{E.F.\ max}}{I_{B\ E.F.\ min}} = \frac{5Ø}{6.3} = 7.95$$

The average turn-on transient dissipation integrated over 16Ø mμsec was found to be 67.5 mw. In turning off, the transient dissipation integrated over 1ØØ was found to be 48 mw. The integrated power dissipation of Q3 may then be determined by the sum of the transient and dc dissipations averaged over one period.

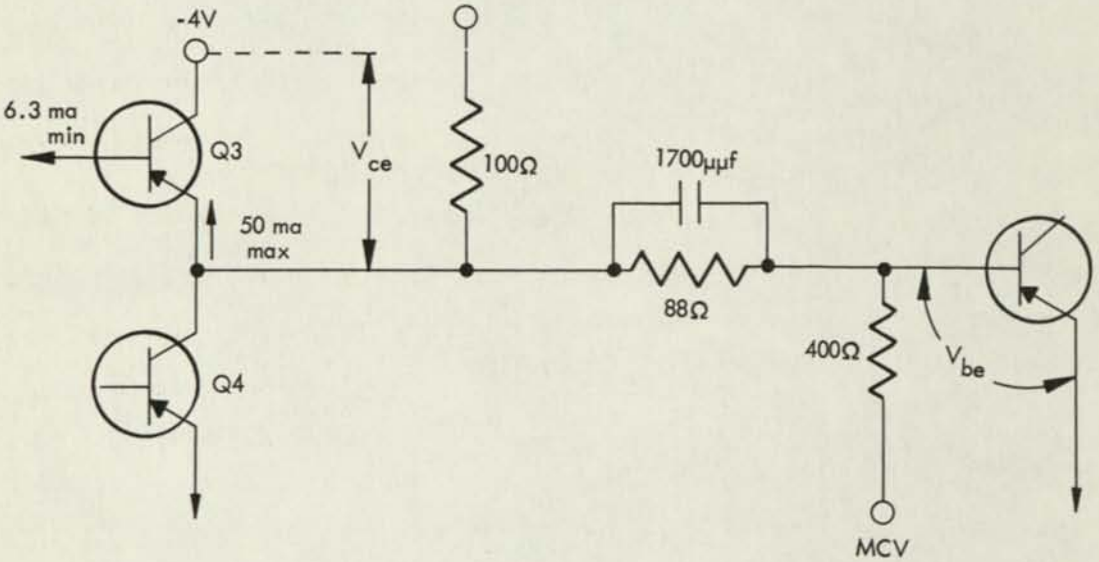


Figure 2-42. Emitter Follower Worst-Case Loading

a) For 1 mc

$P_{\text{trans on}}$ = (67.5)(16Ø) = 1Ø8ØØ mw - mμsec

$P_{\text{dc on}}$ = (5Ø)(Ø.2)(5ØØ - 1ØØ) = 34ØØ mw - mμsec

$P_{\text{trans off}}$ = (48)(1ØØ) = 48ØØ mw - mμsec

$P_{\text{dc off}}$ = (44)(Ø.1)(5ØØ - 1ØØ) = 176 mw - mμsec

TOTAL = 19176 mw - mμsec

$$P_{\text{diss aver max}} = \frac{19,176 \text{ mw - m}\mu\text{sec}}{1\text{ØØØ m}\mu\text{sec}} = 19.18 \text{ mw}$$

b) For 500 kc

$P_{\text{trans on}}$ = (67.5)(16Ø) = 1Ø8ØØ mw - mμsec

$P_{\text{dc on}}$ = (5Ø)(Ø.2)(1ØØØ - 16Ø) = 84ØØ mw - mμsec

$P_{\text{trans off}}$ = (48)(1ØØ) = 48ØØ mw - mμsec

$P_{\text{dc off}}$ = (4.4)(Ø.1)(1ØØØ - 1ØØ) = 4ØØ mw - mμsec

TOTAL = 244ØØ mw - mμsec

$$P_{\text{diss aver max}} = \frac{244\text{ØØ mw - m}\mu\text{sec}}{2\text{ØØØ m}\mu\text{sec}} = 12.2 \text{ mw}$$

2.7.3.4.4 Power Inverter

The power inverter amplifies the output of Q1 to a current level sufficient to drive Q3 under worst-case loading conditions. It is designed to operate with a circuit beta of 1Ø.

$$\text{If } I_{\text{bon min}} = \frac{V_{\text{cl nom}} - V_{\text{be}}}{\text{R4}} - \frac{(4.4)(1.1) + V_{\text{be}}}{(\text{R5})(\text{Ø}.9)} = 1.7 \text{ ma}$$

$$\text{and } I_{\text{c max}} = \frac{1\text{Ø}(1.1) - \text{Ø}.1}{(68\text{Ø})(\text{Ø}.9)} \cong 17.8 \text{ ma}$$

$$\text{then, } B_{\text{ckt max}} = \frac{17.8}{1.72} \cong 1\text{Ø}$$

The maximum I_{co} which may be supplied by the off bias source is computed as follows, assuming a minimum off V_{be} of Ø.5 volts:

$$Ico_{\text{min ckt cap}} = \frac{(4.4)(\text{Ø}.9) - \text{Ø}.\text{Ø}5}{(1\text{Ø})(1.22)} - \frac{(\text{Ø}.5 + \text{Ø}.15)}{(1.5)(\text{Ø}.8)} = 11\text{Ø}\ \mu a$$

If C2 as shown on Figure 2-37 is not large enough to pull Q2 out of storage, the emitter follower will remain off until Q2 finally turns off. Since the circuit delay would be increased by the additional amount of time required to turn off Q2, it is necessary to design C2 to accommodate the worst anticipated conditions.

If the maximum on current into the base of Q2 is 2.4 ma, then the minimum charge C1 must present to Q2 is

$$I_B\ (K's) = (2.4\ ma)(16\text{Ø}\ m\mu sec) = 385\ \mu\mu\ \text{coulombs}$$

Under these conditions where I_{bon} = 2.4 ma, the voltage drop across C2 is 3.9 volts. The charge consumed by R4, assuming 1ØØ mμsec for Q2 to turn off and an average voltage of two volts across C2, is

$$q_{R4\ max} = \frac{2(1\text{ØØ} \times 1\text{Ø}^{-9})}{(\text{Ø}.9)(1.5 \times 1\text{Ø}^{3})} = 148\ \mu\mu\ \text{coulombs}$$

Therefore:

$$q_{C1} = q_{R4} + q_{Q2} = 533\ \mu\mu\ \text{coulombs,}$$

$$C_{min} = (3.9)(\text{Ø}.8) = 533\ \mu\mu\ \text{coulombs,}$$

$$\text{or } C1_{min} = \frac{533 \times 1\text{Ø}^{-12}}{(3.9)(\text{Ø}.8)} = 17\text{Ø}\ \mu\mu f$$

Let C1 = 18Ø μμf

When Q2 turns on, the RC time constant is essentially $\left[\frac{(1.5)(\text{Ø}.82\text{Ø})}{1.5 + \text{Ø}.82}\right]$ $(18\text{Ø} \times 1\text{Ø}^{-12}) = \text{Ø}.\text{Ø}95\ \mu sec$. Since the switch is on for a minimum of 1 μsec $\frac{1.\text{Ø}}{\text{Ø}.\text{Ø}95} = 1\text{Ø}.5$ time constants are allowed, thus assuring the complete charging of C1.

2.7.3.4.5 Input Inverter

The input inverter is the same as a standard MOBIDIC inverter, with the exception of the collector load, 82Ø ohms instead of 1ØØØ ohms to -1Ø volts. The circuit is designed such that the maximum collector off voltage is under -5 volts and the circuit will function with a circuit beta of 1Ø.

2.7.4 Off-Line Control Equipment Power Requirements

The off-line control equipment power requirements are shown in Table 2-11. Units S6Ø2A and S6Ø2B are employed only as off-line equipment. The starred units S7Ø3, S7Ø4 and S721 are used both as off-line and on-line units. There are two S7Ø4-2 units employed in off-line use, and the total currents of both units are entered in the table. The total input power requirement is 2.9 kilowatts at 85% power factor.

TABLE 2-11. OFF-LINE EQUIPMENT POWER REQUIREMENTS

		VOLT - AMPERES										
Model Number	Quantity	-4	-1Ø	-2Ø	-1ØØ	+4.4	+15	+24	+5Ø	5 to 8 Volt MCV Amps	-28 Volt Amperes	Input Watts
S6Ø2A	1	6.59	4.16	5.578	Ø.139	Ø.742	3.75Ø	8.Ø	Ø.229	6.Ø		744.8
S6Ø2B	1	15.6	5.94			1.77		1.Ø		6.Ø		359.7
S7Ø3A & B*	1	6.5	7.9	4.3		2.6		1.Ø	Ø.Ø4	6.Ø		469.9
S7Ø4-2*	2	24.44	13.7			4.126		2.Ø	Ø.16	12.Ø		688.4
S721*	1	11.745	2.19			Ø.433			Ø.Ø4	6.Ø	9.5	657.5
Totals		64.875	33.89	9.878	Ø.139	9.677	3.75Ø	12.ØØ	Ø.429	36.Ø	9.5	292Ø.3

NOTE: Starred items are both on-line and off-line units

2.8 MECHANICAL DESIGN

The newly designed racks required for the off-line control system include the following listed units:

S6Ø2A	Off-Line Control Unit
S6Ø2B	Off-Line Control Unit
S721	Off-Line Device Switching and Buffer Unit

Each of these units is housed in a separate enclosure. The enclosures are similar in construction to standard MOBIDIC-type enclosures.

2.8.1 Off-Line Control Unit (S6Ø2A)

The off-line control unit S6Ø2A enclosure is shown in Figure 2-43. The front of the S6Ø2A enclosure is divided into three sections. The upper front door assembly covers the A1 row. Immediately below this door is the operator's panel. The lower front door assembly covers rows A5 through A9. Rack layout details are given in 2.8.1.1. Both doors are hinged and gasketed and fitted with locking-type handles. The operator's panel is hinged and gasketed and retained in closed position by captive-type screws. The lower door assembly is louvered and fitted with aluminum filters. A self-contained forced air cooling system is provided.

2.8.1.1 Rack Layout

The top row, A1, is used for subassemblies associated with the marginal check voltage system. The arrangement of these subassemblies is shown on Figure 2-3. The controls and indicators on these subassemblies are described in 2.6.2.1. The operator's panel occupies the space normally occupied in standard MOBIDIC-type enclosures by logic-package assemblies in rows A2, A3, and A4. The panel arrangement is shown in Figure 2-4. The controls and indicators on the operator's panel are described in 2.6.2.2 and 2.6.2.3. The position chosen for the operator's panel was based on the desire to place regularly-used controls and indicators approximately at eye level. Controls and indicators on this panel have been functionally grouped. The OLCU memory array with its associated wiring and connectors occupies part of row A5. The memory array is described in detail in 2.8.1.2. OLCU logic-package assemblies occupy the remainder of row A5 and all of rows A6, A7, A8, and A9.

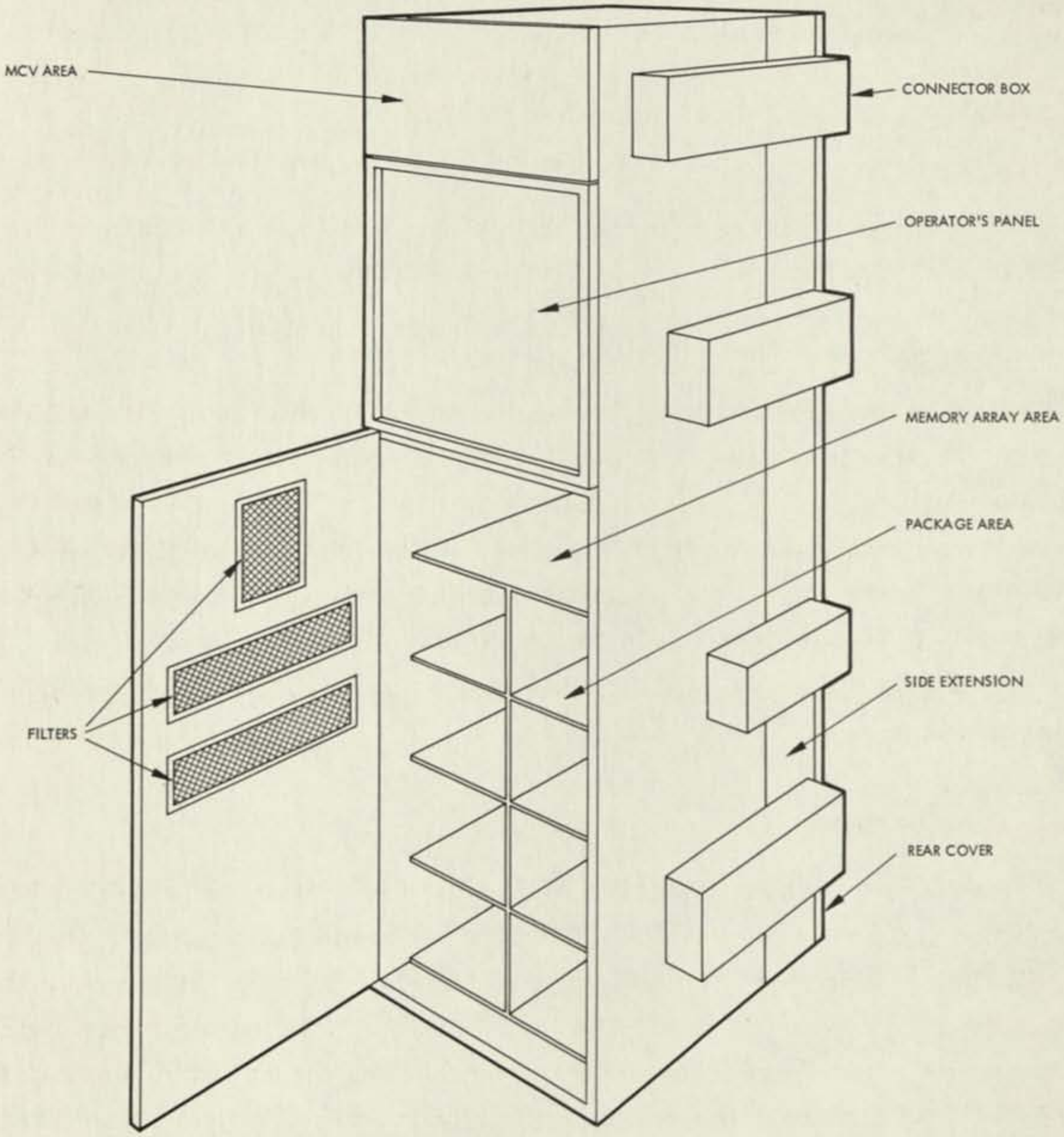


Figure 2-43. Off-Line Control Unit, S602A

2.8.1.2 Memory Array Assembly

The memory array assembly consists essentially of a frame, 9 memory planes, several component board subassemblies, associated wiring, and connectors. This assembly is slide-mounted for ease of maintenance and for four-side accessibility. The location of the memory array assembly in row A5 is at a good working height for maintenance and servicing.

The frame is of welded aluminum construction. It has been designed to support the memory planes and component board subassemblies and to protect them from shock, vibration, and mechanical damage during operation and servicing. A frame mockup has been tested to determine presence of resonant frequencies below 55 cycles per second. For this test the frame was assembled with dummy planes, mounted on a vibration table, and then vibrated successively in three mutually-perpendicular directions that are parallel respectively to the edges of the frame. The frequency range was from 10 to 55 cycles per second in 1 cycle-per-second increments. Careful inspection showed no evidence of mechanical failure as a result of the test. Test results indicated that there were no resonant frequencies below 55 cycles per second.

The 9 memory planes are each of double-matted construction. Each mat includes 336 magnetic cores, arranged in a 14 by 24 core matrix. The planes are assembled horizontally in the frame. Individual planes may be replaced when necessary without removal of adjacent planes.

Two component board subassemblies are attached to the sides of the frame, one on each side. These subassemblies are hinged to the frame. This arrangement allows them to be swung out for maintenance of items mounted on the frame side. Test points are provided on the outer side of each of these component boards. The boards themselves are made of glass-filled epoxy. A third component board is located directly behind the memory planes, in the center of the array. This board mounts four temperature-sensing thermistors.

The connectors are attached to the right rear of the memory array frame. The connectors are each fitted with jackscrews for engagement and disengagement. Cabling is neatly laced and clamped. An accordian-pleated service loop in the cable assembly permits memory operation in the extended position without the use of separate extender cables.

The memory array assembly packaging techniques reflect sound design, compactness, and simplicity without sacrifice of accessibility and maintainability.

2.8.1.3 Cooling

The S6∅2A enclosure is fitted with a centrifugal blower assembly mounted in a recessed section of the rear cover assembly, directly behind the memory array assembly. The blower is of the suction type. It is capable of delivering 155 cubic feet of air per minute (CFM) at 1" H_2O. An air vane switch at the exhaust end of the blower is connected electrically to two indicating lights, one is the blower failure indicator on the operator's panel, the other is on the marginal check voltage system stepping switch subassembly in the A1 row of the enclosure. These indicators light if the rate of movement of the exhaust air drops below a pre-set minimum. This arrangement provides protection for heat-sensitive components, particularly those associated with the memory.

Cooling air drawn in by the blower passes through louvers and filters in the lower front door assembly. The filters, which are of aluminum mesh, are washable. Air entering the enclosure in the area of the logic-package assemblies in rows A5, A6, A7, A8, and A9 travels past the logic-package assemblies in those rows and out through the connector panels. Openings are provided in the connector panels for this purpose. The size of the openings will be determined on the basis of results of an air balancing test carried out using a prototype cabinet. Air entering the enclosure in the area of the memory array in row A5 passes the array and leaves through an exhaust hole in the panel immediately behind the array. Air passing the logic-package assemblies and memory array is ductedto the blower betweenthe connector panels and the gasketed rear cover assembly of the enclosure.

2.8.2 Off-Line Control Unit (S6∅2B)

The off-line control unit S6∅2B enclosure is similar in design to that of the standard MOBIDIC converter enclosure. The unit is cooled by the sametype of blower as is used in the S6∅2A off-line control unit enclosure. The blower assembly is mounted directly on the rear cover assembly. An air vane switch is provided. This is connected electrically to two indicating lights, one is the blower failure indicator on the operator's panel on the S6∅2A unit, the other is on the marginal check voltage system stepping switch subassembly in the A1 row of the S6∅2B unit itself.

2.8.3 Off-Line Device Switching and Buffer Unit (S721)

The off-line device switching and buffer unit S721 enclosure houses the off-line paper tape punch and spooler assembly, the paper tape reader assembly, the marginal check voltage system subassemblies, and the logic-package assemblies associated with the device switching and buffer units. The enclosure is basically of standard MOBIDIC-type construction. The standard design has been modified, however, to permit installation of the paper tape punch and spooler and the paper tape reader. These have been mounted in the area occupied by the A1 through A5 rows in standard enclosures. The arrangement is similar to that of the S7∅2 enclosures. In these rows the shelves for the marginal check voltage system subassemblies and the logic-package assemblies have been omitted. Aluminum angle braces have been installed in that area to provide the added strength needed to support the paper tape punch and spooler and the paper tape reader. The paper tape punch and spooler have been installed in the area normally occupied by rows A1 and A2. These items have been slide-mounted for ease of maintenance and accessibility. The paper tape reader has been hinge-mounted in the area normally occupied by rows A3, A4, and A5. Row A6 is used for the marginal check voltage system subassemblies. Rows A7, A8, and A9 are used for logic-package assemblies in the device switching and buffer units.

The front door assembly islouvered and provided with filters in the area of the four lower rows (rows A6, A7, A8, and A9). Three washable aluminum mesh air filters are used. Two propeller-type ventilating fans are used to move the cooling air through the enclosure. These fans are mounted side by side on the rear cover assembly, in line with row A7. Each fan is rated at 12∅ cubic feet per minute at free discharge. Two indicating lights, one on the S6∅2A operator's panel and the other on the marginal check voltage system stepping switch subassembly in row A6 of the S721 enclosure, provide indication of fan failure.

2.9 EQUIPMENT OPERATION

2.9.1 General

Utility of the off-line control system depends to a certain extent upon the operator's knowledge of the system's capabilities. These capabilities are listed in summary form in Table 2-12. No attempt is made in this document to provide a detailed step-by-step operating procedure. Such information is within the scope of an instruction manual rather than that of a final design plan. A conversion summary is given in 2.9.2, Paragraphs 2.9.3 and 2.9.4 provide descriptions of two major operational features of the off-line control system-operation in the blockette mode, and error correction procedures. The general sequence of operations during a typical conversion cycle was described in paragraph 2.2. The controls and indicators used during OLCU operation are listed and their major functions were described on Tables 2-4 and 2-5.

2.9.2 Conversion Summary

A summary of the operation of the OLCU for various conversions is presented in Tables 2-13, 2-14, and 2-15. The following notes will assist understanding of these tables.

The term selected input - refers to the characters that are actually stored in the OLCU memory in one input cycle. If the blockette mode of operation is used, the term refers to the character actually selected. If paper tape is the input medium, data and control characters are never stored together in the memory. If data characters are being stored and a control character is read into the IBR, or vice versa, the input cycle is terminated immediately and the character in IBR is not sent out until the next output cycle. The selected input can never be larger than the OLCU memory capacity.

The OLCU is halted when an end-of-file condition is detected. An end-of-file condition occurs when an end-of-file mark is read from magnetic tape, a stop code is read frompaper tape, or when the last card is read from the card reader-punch when the hopper is empty. The last condition can only occur under manual control of the card reader-punch.

TABLE 2-12. OLCU INPUT-OUTPUT SUMMARY

		During Input Cycle		During Output Cycle		
Device	Mode Switch	R/C	F/H	C/R	H/F	Padding Charac.
IMT	BIY	∅	∅	∅	∅	∅
	ALF	∅	∅	X	∅	X
	HOL	∅	1	1	∅	∅
IPT	BIY	∅	∅	∅	∅	∅
	ALF	∅	∅	X	∅	X
	HOL	∅	1	1	∅	∅
ICR	BIY	∅	∅	∅	∅	∅
	ALF	1	∅	∅	∅	X
	HOL	1	∅	∅	1	∅
OMT	BIY	∅	∅	∅	∅	∅
	ALF	X	∅	∅	∅	1
	HOL	1	∅	∅	1	∅
OPT OLP	BIY	∅	∅	∅	∅	∅
	ALF	X	∅	∅	∅	∅
	HOL	1	∅	∅	1	∅
OCP	BIY	∅	∅	∅	∅	∅
	ALF	∅	∅	1	∅	∅
	HOL	∅	1	1	∅	∅

Machine action depends upon (1) the devices selected and (2) the mode selected.

Legend:

∅	indicates never done
1	indicates always done
X	indicates may or may not be done depending upon both the input and output devices
R/C	indicates a row to column conversion
C/R	indicates a column to row conversion
F/H	indicates a Fieldata to Hollerith conversion
H/F	indicates a Hollerith to Fieldata conversion

TABLE 2-13. OLCU SUMMARY: MAGNETIC TAPE INPUT

Output MODE	CARD PUNCH	MAGNETIC TAPE	LINE PRINTER	PAPER TAPE PUNCH
BINARY	I. 168 FIELDATA characters are required at the input to produce a complete output card. Fourteen FIELDATA characters are required to punch a row. The first four bits of each row are padding bits and are not punched. The cards are punched as if the input characters are coming from MOBIDIC on-line. II. An integral number of cards are always punched for each input block. If necessary, the last card of a block is padded with blank rows. III. Contents of control blocks are not punched.	I. Output is identical to the selected input.	I. Format as set up on the line printer buffer plug board and tab. plugs. II. Contents of control blocks are not sent to the line printer. III. An end of file block causes feeding to the end of the page and stops the printer.	I. Output is identical to the selected input. II. End of file block produces a stop code on the output.
ALPHA-NUMERIC	I. Two FIELDATA characters are required to punch a column. There are 168 characters required to produce a complete output card. The first eight characters per card are the padding characters and are not punched. II. An integral number of cards are always punched for each input block. If necessary, the last card of a block is padded with blank columns. III. The contents of control blocks are not punched.	I. Output is identical to the selected input, except that, if necessary, each block is padded to the nearest multiple of six characters.	SAME	SAME
HOLLERITH	I. 84 FIELDATA characters are required at the input to produce a complete output card. Since the first four characters of each card of information are padding characters, they are not punched. II. An integral number of cards are always punched for each input block. If necessary, the last card of a block is padded with blank columns. III. The contents of control blocks are not punched. IV. The input data alarm is set when the input character does not have a Hollerith equivalent.	OPERATOR ERROR	SAME	SAME

TABLE 2-14. OLCU SUMMARY: PAPER TAPE INPUT

Output MODE	CARD PUNCH	MAGNETIC TAPE	LINE PRINTER	PAPER TAPE
BINARY	I. 168 FIELDATA characters are required at the input to produce a complete output card. Fourteen FIELDATA characters are required to punch a row. The first four bits of each row are padding bits and are not punched. The cards are punched as if the input characters were from MOBIDIC on-line. II. Punching of a card is terminated by any control character at the input. If necessary, a card is padded with blank rows. III. Control characters are not punched.	I. Output is identical to the selected input.	I. Format as set up on the line printer buffer plug board and tab plugs. II. Control characters are not sent to the line printer. III. Stop code causes feeding to the end of the page and stops the printer.	Output is identical to the selected input.
ALPHA-NUMERIC	I. 168 FIELDATA characters are required at the input to produce a complete output card. Since the first eight characters of each card of information are padding characters, they are not punched. II. Punching of a card is terminated by any control character at the input. If necessary, a card is padded with blank columns. III. Control characters are not punched.	I. Output is identical to the selected input, except that, if necessary, each block is padded to the nearest multiple of six characters. II. Stop Code generates on end of file block.	SAME	SAME
HOLLERITH	I. 84 FIELDATA characters on the input produce a complete output card. Since the first four characters of each card of information are padding characters, they are not punched. II. Punching of a card is terminated by any control character at the input. If necessary, a card is padded with blank columns. III. Control characters are not punched. IV. Input data alarm is set when the input character does not have a Hollerith equivalent.	OPERATOR ERROR	SAME	SAME

TABLE 2-15. OLCU SUMMARY: CARD READER INPUT

Output MODE	MAGNETIC TAPE	LINE PRINTER	PAPER TAPE
BINARY	I. Operator option of one, two, or three cards per block. II. Fourteen FIELDATA characters are generated per row. The first four bits of each row are padding bits which were not on the card. Each card produces 168 characters on the tape. The format on the tape for each card is the same as would be read by MOBIDIC from a card reader on-line. III. Reading of the last card with the hopper empty will generate an end of file block.	I. 168 FIELDATA characters are always sent to the line printer for each card. The first four bits of every fourteenth character are padding bits which were not on the card. II. Format as set up on the line printer buffer plug board and tab plugs. III. Reading of the last card with the hopper empty causes feeding to the end of the page and stops the line printer.	I. 168 FIELDATA characters are always punched for each card. The first four bits of every fourteenth character are padding bits which were not on the card. The format on the tape for each card is the same as would be read by MOBIDIC from a card reader on-line. II. Reading of the last card with the hopper empty will cause punching of a stop code.
ALPHA-NUMERIC	I. Operator option of one, two, or three cards per block. II. The first eight characters on the tape for each card are padding characters which were not on the card. Each card produces 168 characters on the tape. III. Reading of the last card when the hopper is empty will generate an end of file block.	I. 168 FIELDATA characters are always sent to the line printer for each card. The first 8 characters per card are padding characters which were not on the card. II. Format as set up on the line printer buffer plug board and tab plugs. III. Reading of the last card with the hopper empty causes feeding to the end of the page and stops the line printer.	I. 168 FIELDATA characters are always punched for each card. The first eight characters on the tape for each card are padding characters, which were not on the card. II. Reading of the last card with the hopper empty will cause punching of a stop code.
HOLLERITH	I. Operator option of one, two, or three cards per block. II. 84 FIELDATA characters per card are written. The first four characters on the tape for each card are padding characters which were not on the card. III. Reading of the last card with the hopper empty will generate an end of file block.	I. 84 FIELDATA characters are always sent to the line printer for each card. The first four characters per card are padding characters which were not on the card. II. Format as set up on the line printer buffer plug board and tab plugs. III. Reading of the last card with the hopper empty causes feeding to the end of the page and stops the printer.	I. 84 FIELDATA characters are always punched for each card. The first four characters on the tape for each card are padding characters which were not on the card. II. Reading of the last card with the hopper empty will cause punching of a stop code.

2.9.3 Operation in the Blockette Mode

The blockette mode of operation provides means of selecting portions of the input data for conversion during an OLCU cycle. This feature can be used for both magnetic and paper tape input. The blockette MARK and KEY switches on the operator's panel are used only when the blockette mode of operation is used.

When paper tape is the input medium, only the MARK switches are used. The blockette mark character will indicate the beginning and end of the data to be converted during one complete in-out cycle. When the blockette mode is selected, characters are read into the OLCU memory until either the memory is full or a character identical to the mark character as determined by the setting of the MARK switches has been read. Reading of a mark character caused the input operation to stop and the output operation to start. The mark character is not stored in memory and does not appear in the output. When converting from paper tape to magnetic tape, the blockette mode allows preparation of variable length blocks on magnetic tape.

When the input is magnetic tape, both the MARK and KEY switches are used. When an input cycle is started in the blockette mode, the characters from the magnetic tape are searched until one identical to the character indicated by the setting of the MARK switches has been found. Until one is found, characters are not stored in memory. When a mark character is found, the next character is compared with the character indicated by the setting of the KEY switches. If the next character is the same as the character determined by the setting of the KEY switches, all succeeding characters are stored until either the next mark character is found, or the block ends, or the OLCU memory is filled. If a mark character is decoded, the next character is examined to see if it is the same as the key character indicated by the setting of the KEY switches. If the key characters are the same, succeeding characters are stored in memory. If a key character is not read, no characters are stored in memory until the next mark character has been read and the process is repeated. Thus, all characters from a block between a mark and key character which are the same as those indicated by the MARK and KEY switches and another mark character which is the same and a key character which is different are stored in the OLCU memory. The total number of

If 1st char is not = to Mark in OLCU, Data will be lost until 1st Mark is found.

characters to be stored is restricted, however, by the size of the memory. The mark and key characters are not read into the memory and do not appear on the output medium.

For each bit of the mark and key characters, a ONE, ZERO, or neither, can be selected by the MARK and KEY switches. When neither a ONE nor a ZERO is selected, that bit position of each incoming character is not examined and the content of that bit position does not matter insofar as decoding selected mark and key characters is concerned. This feature allows selection of specially-classfied blockettes.

Use of the blockette mode with magnetic tape as input allows selection of portions of the input blocks for conversion. This feature provides specialized searching capacity and allows on-line preparation for off-line conversion of magnetic tapes with blocks longer than those which can be handled in one conversion cycle of the OLCU. For example, when an off-line printing operation is to be done, long blocks which contain several documents each can be prepared on-line on magnetic tape. This effects a saving in computer time. One document per block can then be printed off-line for each pass of the input tape through the OLCU.

2.9.4 Error Correction Procedures

The error control features of the OLCU are so designed that, for the types of errors which are expected to be most frequent,suitable error correction procedures can be followed. Care has been taken to make the correction procedures simple and to use common procedures for different devices and media. These procedures are summarized here. The auxiliary card buffer, described in Section IV, is used when checking cards.

1. PAPER TAPE-PARITY ERROR (INPUT DATA ALARM)

   a. Machine halts with error character displayed in the INPUT BUFFER REGISTER.

   b. By looking at the tape reader the operator can see about 5 characters on either side of the one in error and if possible, can decide what the proper character should be.

c. Set up the desired character in the INPUT SWITCH REGISTER (ISR).

d. Press the READ IN switch which places the contents of (ISR) into the INPUT BUFFER REGISTER.

e. Press START which clears the error alarm and continues the operation.

2. MAGNETIC TAPE-READ ERROR (MEMORY PARITY ALARM-INPUT DATA ALARM)

a. Machine halts at the end of error block.

b. Press Input BACKSPACE - this moves the operation back to the beginning of the error block.

c. Press START - this clears the error alarm, rereads the block, and continues the operation

3. CARD ERROR - READ CHECK ERROR (MEMORY PARITY ALARM-INPUT DATA ALARM)

a. Machine halts after bad card is checked.

b. Empty input hopper of cards.

c. Clear cards from the card machine (3 cards).

d. In the case of paper tape output pick up the last 5 cards from the stacker and place in the input hopper. In the case of magnetic tape output pick up N + 2 cards from the stacker where N = the number of cards per block and replace in the input hopper.

e. Replace unread cards to input hopper.

f. Reset the card machine.

g. Press START on the OLCU which continues the operation.

4. MAGNETIC TAPE - WRITE ERROR (OUTPUT DATA ALARM)

a. Machine halts after error block.

b. Press output BACKSPACE which sets the operation at the beginning of the bad block.

c. Press START which rewrites the block and continues the operation.

d. If another alarm is obtained, it may be due to a bad spot on the tape.

e. Press output BACKSPACE again.

f. Press ERASE which clears the block of tape.

g. Press START which continues the operation.

5. CARD ERROR - READ AFTER PUNCH (OUTPUT DATA ALARM)

a. Machine halts after checking bad card.

b. Remove blank cards from input hopper.

c. Clear out card machine (3 cards).

d. Pick up the last four cards in the output stack and discard the top three.

e. Place the fourth card in the input hopper and then replace all the blank cards.

f. Reset the card machine.

g. Press START which continues the operation.

MEMORY PARITY errors with paper tape input cannot be corrected. Also, there can be no check by the OLCU of the output going to paper tape or line printer. If at any time one desires to ignore an error and continue after the OLCU has stopped the START switch will clear the error alarm and continue.

2.10 SYSTEM OPERATION

The off-line control system has been designed to operate within and in conjunction with the entire MOBIDIC 7A data processing system. Several aspects of the inter-system relationships are discussed in paragraphs 2.10.1 through 2.10.2.

2.10.1 In-Out Device Sharing

The off-line control system and the central computer share certain in-out devices. These devices are: the two card reader-punches, two of the magnetic tape transports, and the line printer. An interlock has been provided to assure proper operation of shared in-out devices i.e., in-out devices capable of either on-line or off-line use. The interlock permits uninterrupted operation, either on-line or off-line, once a shared device has been selected for one or the other type of operation. If a program in the computer addresses a shared device already in off-line use a device alarm signal is sent to the computer.

There are five toggle switches and five indicators on a status panel above the console in van 1. A toggle switch and indicator are associated with each shared device. The switches each have a RELEASE and a SELECT position. The indicators light to signal that a shared device (all of which are in van 2) is being used off-line.

The computer operator in van 1, when preparing to run a given program, must first check the indicators on the status panel to determine the status of the shared devices. This must be done so that a shared device already operating off-line will not be selected for on-line operation during running of the program. If a desired device is not in off-line use, the toggle switch associated with that device should be set to the SELECT position. Once this has been done, the device is reserved for on-line use and it is, therefore, unavailable for off-line use until released by the computer operator. It is released by setting the associated toggle switch at the RELEASE position. If the computer operator in van 1 attempts to select a shared device which is already in use off-line (as signaled by the appropriate indicator on the status panel), a device alarm indicator on the console lights and the computer halts as soon as that device is addressed during running of the program.

The computer operator in van 2 is also able to select shared devices for off-line operation without fear that such operation will be disturbed by computer attempts to use the selected device on-line. There is a separate switch on the OLCU operator's panel for each type of in-out device (magnetic tape, paper tape, card reader during input; magnetic tape, paper tape, card punch, line printer during output) which can operate with the OLCU. When a shared device has been selected for off-line use by appropriate switch operation, that device is reserved for off-line use. A three-bit address is generated when a shared device is reserved. This address, which is decoded at the device switching unit associated with the reserved device, prevents the computer from attempting to operate that device on-line. If the operator in van 2 attempts to select for off-line operation a shared device which has already been reserved for on-line operation, a device alarm indicator on the OLCU operator's panel lights when an attempt is made to carry out a conversion involving use of that device.

2.10.2 OLCU Addressing

The addresses used by the OLCU when addressing shared devices during off-line operation are, with one exception, independent of those used by the computer during on-line operation. The exception is in the case of the shared magnetic tape transports. Each such transport has one address which is used during either on-line or off-line operation. This arrangement permits use of a single address switch on the transport for both on-line and off-line addressing. It should be noted that the address of a shared magnetic tape transport can be changed by manual operation of the address switch.

The OLCU is capable of addressing in-out devices as follows: input magnetic tape transport, output magnetic tape transport, input paper tape reader, output paper tape punch, input card reader-punch, output card reader-punch, and output line printer.

The OLCU bus is similar to the converter bus. Thus, it contains six dual bus lines for device addressing. In the case of the OLCU bus, however, only three of these dual lines (busses 20, 21, and 22) are actually used for transmittal of device address bits. The first three address bus lines (busses 17, 18, and 19) are used for device selection and reservation. As soon as a shared device has been selected for off-line use, the level of these three bus lines is raised (to a logical ONE). The raised levels are decoded by the device switching unit (DSU) associated with the selected device to provide a "reserved" signal. This signal is used to prevent on-line use of that device by the computer. This signal is also used to light the appropriate indicator on the status panel in van 1. The device switching unit associated with a selected device is not actually enabled until the remaining three device address bits have been gated ON by the go flip-flop (GO) in the OLCU.

With the exception of the magnetic tape transports, the off-line addresses of all devices used with the OLCU are fixed. The OLCU sends out one fixed address for the input magnetic tape transport and a different fixed address for the output magnetic tape transport. Either of the two transports can be made the input or the output transport by suitably setting the address switches provided on the S721 enclosure. Either switch can be set to either a magnetic tape transport input or output address.

Two card reader-punches are included in the computer. Normally, both are connected in the complete MOBIDIC 7A system, one assigned to on-line operation, the other to off-line operation. In such an arrangement either of the card reader-punches can be used on-line or off-line if the other is inoperable (e.g., during maintenance). If only one card reader-punch is connected in the MOBIDIC 7A system, that device must be shared. On-line or off-line operations may, of course, be delayed when a card-reader punch must be shared.

At times it may be desirable to connect both card reader-punches for on-line operation. Each card reader-punch then has its own address. Provision has been made, however, for addressing one card reader-punch by either address. This feature is a programming convenience for use when only one of these devices is operable. A switch has been installed on each of the card reader-punch buffer units for this purpose.

During off-line card reader-punch operation the OLCU sends out one fixed card reader-punch address. Switches are provided to prevent this one address from reaching both card reader-punches in the event that both such devices are connected for off-line operation. These switches, one on each card reader-punch buffer unit, are so interlocked that both cannot be energized at the same time. The particular card reader-punch to be used off-line is selected manually by operating the switch on the appropriate buffer unit. Operation of the switch enables the off-line DSU associated with the selected device.

2.11 MARGINAL CHECK VOLTAGE SYSTEM

2.11.1 General

The marginal check voltage (MCV) system used in MOBIDIC 7A is the same as that used in other MOBIDIC computers with the exception of the method used to control application of the MCV to equipment in van 2. In that van the MCV is applied under manual control instead of automatically under computer control. This arrangement is necessary since the OLCU, which is used in controlling MCV application, is not a stored-program device. The only equipment changes required were the use of different stepping switch subassemblies in van 2 cabinets. The stepping switch assemblies for manual and automatic control, although physically interchangeable, are wired differently.

Application of the MCV to equipment in van 2 is regulated by the MCV controls included in the off-line control S602A unit. The MCV controls and associated indicators are shown on Figure 2-3 and described in paragraph 2.6.2. These controls and indicators are used to turn on MCV power, adjust the MCV level, and control MCV switching. The MCV controls can be used to regulate marginal checking of the OLCU and of all the device switching and buffer units in van 2 with the exception of the on-line device switching units (DSUs). These DSUs operate on-line with the central computer. The MCV is applied to these on-line DSUs under manual control from the OLCU. However, their operation during application of the selected MCV is checked by the central computer.

During marginal checking of the equipment in van 2 a ripple pattern is first read into the OLCU and then written out. This is done while the marginal check voltage is applied, one row at a time, to the equipment being used. The output is monitored to determine whether an error occurred. By repeating the sequence each of the equipments can be checked. When an error occurs, the sequence is repeated while the marginal check voltage is applied, one logic-package assembly at a time, to the logic-package assemblies in the row in which the error was detected. This arrangement permits accurate determination of the individual logic-package assembly in which a failure has been forced by MCV application.

2.11.2 Operational Control

The marginal check voltage control circuits are shown on Figure 2-44. When the RACK SELECTOR switch is at the OFF position MCV power is turned off and the stepping switches are at the reset position. MCV power is turned on by rotating the RACK SELECTOR switch from the OFF position to the number corresponding to the cabinet to be checked. When the MCV power is turned on the MCV POWER ON indicator lights. At the same time, the RESET indicator in the MCV subassembly in the rack to be checked lights to inform the operator that the stepping switch for that rack is in the reset, neutral, position.

The GAIN-LEAKAGE switch is set at the GAIN position when testing for low-gain transistors. It is set at the LEAKAGE position when testing for high-leakage transistors. In MOBIDIC circuits the +4.4 volt bias supply for the PNP transistors is varied to determine marginal transistors. Decreasing the bias voltage from +4 volts to ∅ volts simulates high leakage conditions while increasing the bias voltage from +4 volts to +12 volts simulates low gain conditions.

The MAN ADJ - AUTO ADJ (manual adjust - automatic adjust) switch controls the method of setting the marginal check voltage level. When this switch is set at the AUTO ADJ position the voltage level is fixed at a preset value. When it is set at the MAN ADJ position, the operator can adjust the level by using the potentiometer provided for manual voltage control. The MCV TEST LEVEL meter indicates the voltage level applied to the circuits during marginal checking. The meter is so connected that is reads full scale at 12 volts during the gain test and at 4 volts during the leakage test. Calibration is in terms of percent of full-scale value.

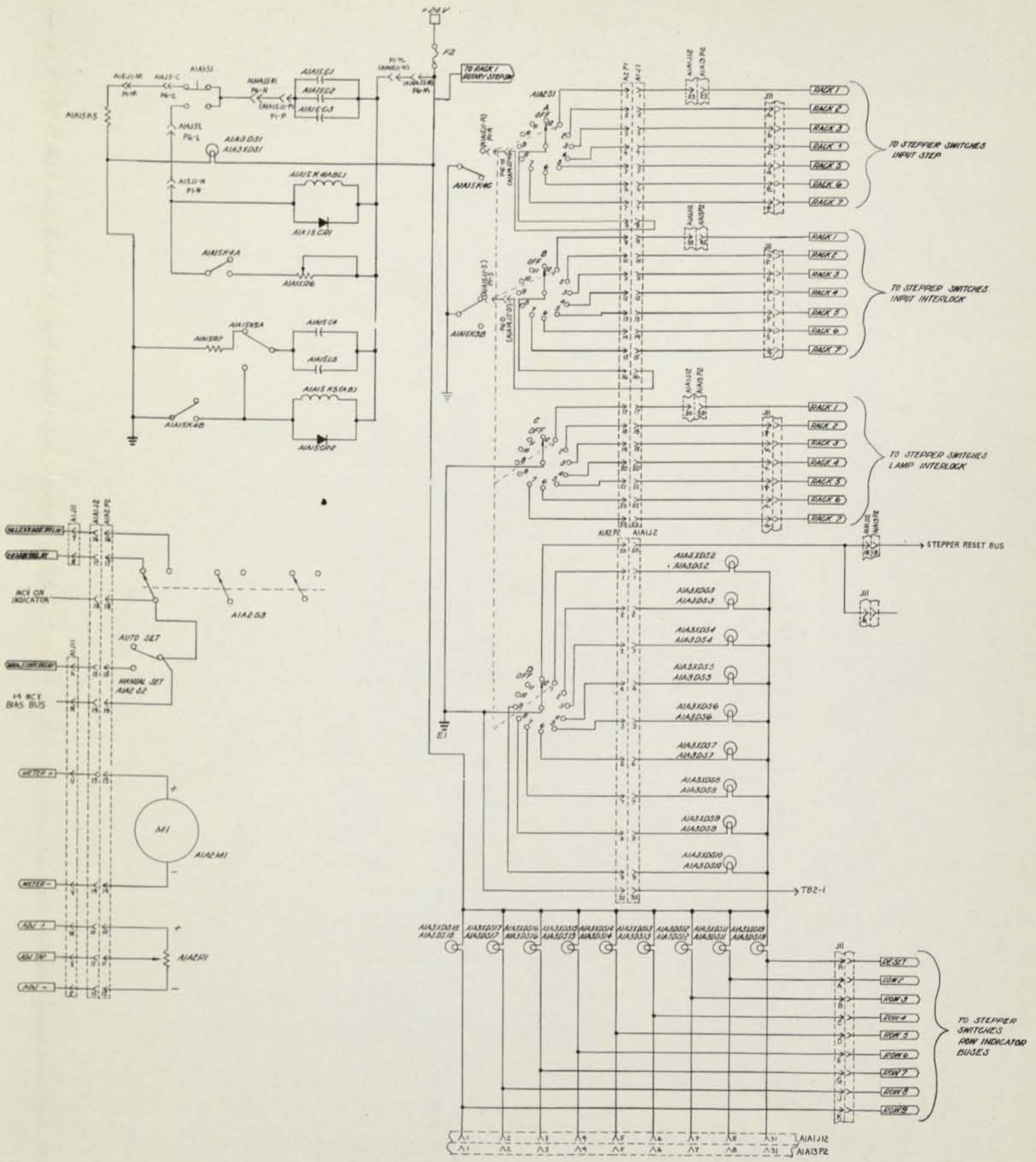


Figure 2-44. MCV Control Circuits

A special timing sequence is required during MCV application. This sequence must deliver a timed input step pulse to the stepping switch coil and open the function selection circuit to protect the stepping switch contacts. Proper timing of this sequence is provided by a relay-capacitor-diode circuit. This circuit is shown on Figure 2-44. The sequence is initiated by operation of the ROW STEP switch on the cabinet to be checked. The input step pulse is approximately 30 milliseconds in duration. The function selection circuit is opened at the same time that the input step pulse starts and it remains open for approximately 30 seconds longer than the input step pulse. During marginal checking of a given cabinet the stepping switch is advanced once for each time that the ROW STEP switch is operated. The stepping switch does not actually advance, however, simultaneously with the input step pulse. The advance takes place during the time the function selection circuit is open. In effect, therefore, the MCV power is held off while changing from row to row during application of the MCV. Indicators on the MCV subassemblies, lighted by operation of stepping switch contacts, display the number of the row being checked.

The basic function of the RACK SELECTOR switch is to switch the marginal check voltage from one cabinet to another. It also performs several additional functions, as follows:

1. Switch Deck A - transmits the input step pulse to the proper stepping switch.
2. Switch Deck B - transmits an input interlock to the proper stepping switch to open the MCV power circuit when stepping from one position to another.
3. Switch Deck C - provides a lamp interlock to the appropriate stepping switch in order to turn on the proper row indicator lamp.
4. Switch Deck D - directly lights the rack indicator lamps to indicate the cabinet being checked. In position 1, it resets all of the stepping switches.

2.12 OLCU ABBREVIATIONS

A list of the abbreviations used in reference to the OLCU follows:

Abbreviation	Description
A	Control flip-flop used with the sequence counter (SQ) for obtaining an odd-microsecond cycle time for magnetic tape timing.
ACB	Signal from the auxiliary card buffer indicating that it is transmitting data to the OLCU. Sent over bus line 13.
ACXB	Allow count in the XB-counter.
ADR1-6	Address lines used to turn on the device switching units (DSUs).
Alarm	Indicates that a data alarm has occurred.
ALF	Signal from the operator's panel: ALPHA-NUMERIC mode.
ALFW	Mode selection switch on the operator's panel: ALPHANUM (alphanumeric) mode.
AMFC N	Auxiliary function M. The memory buffer register (MBR) contains a Hollerith character for which the corresponding Fieldata character has a ONE in the Nth bit position.
AMFC1	Auxiliary function M. The memory buffer register (MBR) contains a Hollerith character for which the corresponding Fieldata character has a ONE in the first bit position.
A1FC1	Auxiliary function 1. The memory buffer register (MBR) contains a Hollerith character for which the corresponding Fieldata character has a ONE in the first bit position.
A1MAC	Add one to the memory address counter.
A1SP	Auxiliary function 1. The memory buffer register (MBR) contains a special Hollerith character.
A1ST*	Add one to ST-counter pulse.

Abbreviation	Description
A1XB	Add one to the XB -counter of the memory address counter.
A1YB	Add one to the YB-counter of the memory address counter.
A2FC1	Auxiliary function 2. The memory buffer register (MBR) contains a Hollerith character for which the corresponding Fieldata character has a ONE in the first position.
A2SP	Auxiliary function 2. The memory buffer register (MBR) contains a special Hollerith character.
A3SP	Auxiliary function 3. The memory buffer register (MBR) contains a special Hollerith character.
B	Control flip-flop used with the sequence counter (SQ) for obtaining an odd-microsecond cycle time for magnetic tape timing.
BACK	Signal which indicates that a backspace magnetic tape operation is to be performed.
BIM	Backspace input magnetic tape control flip-flop.
BIW	Mode selection switch on the operator's panel: BINARY mode.
BIY	Signal from the operator's panel: BINARY mode.
BKT	The character in the input buffer register (IBR) is a blockette mark.
BKTW	Mode selection switch on the operator's panel: BLKT (blockette) mode.
BLE	End-of-block mark.
BLS	Start-of-block mark.
BOM	Backspace output magnetic tape control flip-flop.

Abbreviation	Description
BSI	Signal from the operator's panel indicating that the INPUT BACKSPACE switch has been energized.
BSIW	The INPUT BACKSPACE switch on the operator's panel.
BSO	Signal from the operator's panel indicating that the OUTPUT BACKSPACE switch has been energized.
BSOW	The OUTPUT BACKSPACE switch on the operator's panel.
Cancel	Prevents sending the received character, which is in the input buffer register (IBR) to the memory buffer register (MBR).
CB1 - CB4	Signals from the operator's panel which indicate the number of cards to be processed per block.
CB1W - CB4W	Set of switches on the operator's panel; used to select the number of cards to be processed per block.
CHECK	Flip-flop used to indicate that the first card has been read and that checking can start.
CIBR N	Complement bit N of the input buffer register (IBR).
CINI	Character in the input buffer register (IBR) signal sent from external control to internal control.
CINO	Character in the output buffer register (OBR) signal sent from internal control to external control.
CLE	Signal which indicates that a clear operation is to be performed.
CLOCK*	Pulse at a one-megacycle rate, originating from the clock generator whenever OLCU power is turned on.

Abbreviation	Description
CLR	Signal from the operator's panel indicating that the CLEAR switch has been energized.
CLRDC	Clear internal control.
CLRIBR	Clear the input buffer register (IBR).
CLROBR	Clear the output buffer register (OBR).
CLRSYN	Clear external control signal.
CLRWAR	Clear the write amplifier signal to the magnetic tape transport.
CLW	The CLEAR switch on the operator's panel.
CLXA	Clear the XA-counter.
CLXB	Clear the XB-counter.
CLYA	Clear the YA-counter.
CLYB	Clear the YB-counter.
CMBR N	Complement bit N of the memory buffer register (MBR).
CON	Flip-flop which is set whenever a control character is received during an input operation.
Count	Signal generated by the external control. Causes the memory address counter (MAC) to be incremented by ONE every two microseconds.
CSIP	A level used in memory to clear the memory control (MCF) and inhibit control (ICF) flip-flops.
CSY	Flip-flop used to clear external control.
CY	Cycle counter.
DSU	Device switching unit.
DVA	Device alarm.

Abbreviation	Description
DXBØ	The decoded output of the XB-counter equals ZERO (Ø).
EAM	Electric accounting machine.
EBM	The output buffer register (OBR) contains an end-of-block mark.
EDP	Electronic data processing.
EHT	Signal from the operator's panel indicating that the EMERG HALT (emergency halt) switch has been energized.
EHTW	The EMERG HALT (emergency halt) switch on the operator's panel.
EMPTY	Indicates that no characters have been accepted from the magnetic tape during an input operation.
END	Flip-flop which is set when the last card has been read.
EOF	End-of-file mark.
EOTF	End-of-tape control flip-flop.
EOTW	End-of-tape warning signal from the magnetic tape unit.
ERASE	Flip-flop used to indicate an erase operation on magnetic tape is to be carried out.
ERS	Signal from the operator's panel indicating that the ERASE switch has been energized.
ERSW	The ERASE switch on the operator's panel.
FCO N	The Nth bit of the Fieldata code.
FIN	Finished-reading signal sent from the read synchronizer.
FINI	Finished with the input buffer register (IBR) signal sent from internal control to external control.

Abbreviation	Description
FINO	Finished with the output buffer register (OBR) signal sent from external control to internal control.
FP	Fieldata parity bit as generated by the Hollerith to Fieldata code converter.
FRC	Forward control flip-flop used to drive bus line 1Ø.
GO	Flip-flop used to gate signals from external control to the devices.
H	Halt flip-flop. The machine is halted when the halt flip-flop is set (ONE state).
HAØ1	Memory buffer register (MBR) bits 12, 13, or 14 contain a ONE. This information is used in the Hollerith to Fieldata code converter.
HAØ2	Memory buffer register (MBR) bits 9, 1Ø, or 11 contain a ONE. This information is used in the Hollerith to Fieldata code converter.
HAØ3	Memory buffer register (MBR) bits 4, 5, or 6 contain a ONE. This information is used in the Hollerith to Fieldata code converter.
HAØ4	Memory buffer register (MBR) bits 1, 2, or 3 contain a ONE. This information is used in the Hollerith to Fieldata code converter.
HC N	The input buffer register contains a Fieldata character for which the corresponding Hollerith character has a ONE in bit position N.
HLT	Signal from the operator's panel indicating that the HALT switch has been energized.
HLTW	The HALT switch on the operator's panel.
H/N	Octal designation for the decoded sequence counter (SQ) bits 4 to 6.

Abbreviation	Description
HOLW	Mode selection switch on the operator's panel: HOLL (Hollerith) mode.
HPH	Parity bit for Hollerith bits 7 through 12. The input buffer register (IBR) contains a character for which the corresponding Hollerith character has an odd number of ONEs in the upper (high-order) 6 bit positions.
HPL	Parity bit for Hollerith bits 1 through 6. The input buffer register (IBR) contains a character for which the corresponding Hollerith character has an odd number of ONEs in the lower (low-order) 6 bit positions.
IBR	The input buffer register.
ICC	Type of operation signal: input operation, column card format. INPUT. (INCL + INHOL)
ICD	Type of operation signal: input operation card reader. INPUT. INCD
ICF	Inhibit control flip-flop.
ICR	Type of operation signal: input operation, row card format. INPUT. INCROW
ICXB	Inhibit carry into the XB-counter.
IDTF	Input data error flip-flop.
IDVA	Input device alarm flip-flop.
IGT	Flip-flop used to gate information on the input data lines into the input buffer register (IBR).
IHCY	Inhibit sequencing of the cycle counter (CY).
IHP	Inhibit parity signal, used when 5-channel paper tape is read. Sent over bus line 15.
IHSQ	Inhibit sequence counter (SQ) signal, used to hold the current state of the sequence counter.
IHST	Inhibit the count in the ST-counter.

Abbreviation	Description
IMT	Type of operation signal: input operation, magnetic tape. INPUT. INMT
INCD	Signal from the operator's panel: CARD READ (card reader) input device. Also known as INCR.
INCL	Signal from the operator's panel: the input data format is non-Hollerith coded column cards. INCRW. ALFW
INCR	Signal from the operator's panel: CARD READ (card reader) input device. Also known as INCD.
INCROW	Signal from the operator's panel: the input data format is row cards. INCRW. BIW
INCRW	Device selection switch on the operator's panel: CARD READ (card reader) input.
INHOL	Signal from the operator's panel: the input data format is Hollerith coded cards. INCRW. HOLW
INMT	Signal from the operator's panel: magnetic tape input device.
INMTW	Device selection switch on the operator's panel: MAG TAPE (magnetic tape) input.
INPT	Signal from the operator's panel: PAPER TAPE input device.
INPTW	Device selection switch on the operator's panel: PAPER TAPE input.
INPUT	Signal which indicates that an input operation is to be performed.
INZ	Inhibit level.
IPAR	The input buffer register (IBR) contains a character with even parity.
IPT	Type of operation signal: input operation, paper tape reader. INPUT. INPT

Abbreviations	Description
ISA	Inhibit sense amplifier.
ISR	The input switch register on the operator's panel.
ISR1 - ISR8	Output signals from the input switch register (ISR).
ISR1W - ISR8W	Input switch register (ISR) switches on the operator's panel.
K1Ø - K71	Output signals from the blockette KEY switches.
KEY	The input buffer register (IBR) contains a character selected by the blockette KEY switches.
KEYØ1W - KEY17W	The blockette KEY switches on the operator's panel.
LAST	Flip-flop which is set when no more information is available (e. g., end of tape) and the output medium is cards. Its setting indicates that there is still one more card to be checked.
LBLS	The input buffer register (IBR) contains a start-of-block mark.
LBS	The input buffer register contains either a start-of-block mark or a start-of-control-block mark. LBLS + LSCB.
LCARD	A signal from the auxiliary card buffer, received when the last card in the card reader is being read. Send over bus line 12.
LDEL	The input buffer register (IBR) contains the paper tape format code for "code delete".
LEB	The input buffer register (IBR) contains an end-of-data or end-of-control-block mark.
LEBM	The input buffer register (IBR) contains an end-of-block, end-of-file, or end-of-control-block mark.
LEOF	The input buffer register (IBR) contains an end-of-file mark.

Abbreviations	Description
LPTF	The input buffer register (IBR) contains the paper tape format code for "blank tape".
LROW	The last row of a card is being punched. Received on bus line 16.
LSCB	The input buffer register (IBR) contains a start-of-control-block mark.
LSTP	The input buffer register (IBR) contains the paper tape format code for "stops".
M1Ø - M71	Output signals from the blockette MARK switches on the operator's panel.
MAC	Memory address counter.
Master Control	Another term for program control.
MAX	The memory is full.
MBR	Memory buffer register.
MCF	Memory control flip-flop.
MCV	Marginal check voltage.
MEB	The character in the upper (high-order) bit positions of the memory buffer register (MBR) is an end-of-block mark.
MEM1	A mismatch or end-of-block character has been detected in the memory buffer register (MBR).
MHPAR	Bits 9 through 16 of the memory buffer register (MBR) contain odd parity.
MIS	The character in the input buffer register (IBR) is not of the same class (data or control) as was the previously-accepted character.
MISM	The character in the upper (high-order) bit positions of the memory buffer register (MBR) is not of the same class (data or control) as is specified by the state of the CON flip-flop.

Abbreviations	Description
MLPAR	Bits 1 through 8 of the memory buffer register (MBR) contain odd parity.
MPE	Memory parity error recording flip-flop.
MRKØ1W - MRK17W	The blockette MARK selector switches on the operator's panel.
MXA5	The XA-counter holds a count of 5.
NCB	Signal indicating the number of cards to be processed per block, derived from the four cards-per-block switches on the operator's panel.
NHE	Signal from the operator's panel indicating the non-halt on error mode of operation.
NHEW	The IGNORE DATA ERROR (non-halt on error) switch on the operator's panel.
OBR	Output buffer register.
OCC	Type of operation signal: output operation, column card format. OUTPUT. (OUTCL + OUTHOL)
OCP	Type of operation signal: output operation, card punch. OUTPUT. OUTCP
OCR	Type of operation signal: output operation, row card format. OUTPUT. OUTGROW
ODTF	Output data alarm flip-flop.
ODVA	Output device alarm flip-flop.
OGT	Flip-flop used to gate the information in the output buffer register (OBR) onto the data lines.
OLCU	The off-line control unit.
OLP	Type of operation signal: output operation, line printer. OUTPUT. OUTLP
OLPT	Type of operation signal: output operation, line printer or paper tape punch. OLP + OPT
OMT	Type of operation signal: output operation, magnetic tape. OUTPUT. OUTMT

Abbreviations	Description
OPERROR	Signal indicating that the switches on the operator's panel have not been properly set up.
OPT	Type of operation signal: output operation, paper tape punch. OUTPUT.OUTPT
OUTCL	Signal from the operator's panel: the output data format is non-Hollerith column cards. OUTCPW.ALFW
OUTCP	Signal from the operator's panel: CARD PUNCH output device.
OUTCPW	The device selection switch on the operator's panel: CARD PUNCH output.
OUTGROW	Signal from the operator's panel: the output data format is row cards. OUTCPW.BIW
OUTHOL	Signal from the operator's panel: the output data format is Hollerith-coded cards. OUTCP.HOLW
OUTLP	Signal from the operator's panel: line printer output device.
OUTLPW	The device selection switch on the operator's panel: LINE PRINT (line printer) output.
OUTMT	Signal from the operator's panel: magnetic tape output device.
OUTMTW	Device selection switch on the operator's panel: MAG TAPE (magnetic tape) output.
OUTPT	Signal from the operator's panel: paper tape output device.
OUTPTW	Device selection switch on the operator's panel: PAPER TAPE output.
OUTPUT	Signal which indicates that an output operation is to be performed.
p*	A 5ØØ KC timing pulse, alternates with t-pulse.
PC	The program counter, used in program, or master, control.

Abbreviations	Description
PTF	Transfer from the memory buffer register (MBR) to the output buffer register (OBR) in paper tape format.
Q	A one megacycle timing pulse. Coincides with p- and t-pulses.
RC	Read control counter, also known as the ST-counter.
RCF	Read control flip-flop, used to drive bus line 11.
RDYD	Ready line input to external control from the device, received on bus line 23.
RDYF	Ready flip-flop.
RI	Signal from the operator's panel that the READ IN switch has been energized.
RIW	The READ IN switch on the operator's panel.
RLV	Read level sent to memory control.
RTS	Read timing level.
RUN	Signal from the operator's panel indicating that the RUN mode of operation has been selected.
RUNW	The RUN switch on the operator's panel.
RVF	Reverse control flip-flop, used to drive bus line 9.
SA N	Output of sense amplifier N.
SA	Sense amplifier.
SBA*	Strobe pulse.
SBM	The output buffer register (OBR) contains a start-of-block or start-of-control-block mark.
SCB	Start-of-control-block mark.
SCL	Signal from the device switching unit (DSU) indicating that the selected device operates according to Signal Corps Technical Requirements SCL-1986, received on bus line 25.

Abbreviations	Description
SC N	Control flip-flop N used in internal control to indicate states of internal control.
SCY	Signal from the operator's panel indicating that the SINGLE CYCLE mode of operation has been selected.
SCYF	Single cycle control flip-flop.
SCYW	The SINGLE CYCLE mode switch on the operator's panel.
SEC	Control flip-flop in internal control which usually indicates which half (high-order or low-order bit positions) of the memory buffer register (MBR) is to be used.
SHMBR	Shift memory buffer register (MBR).
SIP	Signal from the operator's panel indicating that the SINGLE PULSE mode of operation has been selected.
SIPW	The SINGLE PULSE mode switch on the operator's panel.
SKW	The ST-counter has counted the necessary delay from the receipt of the synchronizing bit (during magnetic tape input). Insures that a complete character has been received in the input buffer register (IBR).
SLCB	The input buffer register (IBR) contains a start-of-control-block mark.
SLIBR	Shift left the contents of the input buffer register (IBR).
SLOBR	Shift left the contents of the output buffer register (OBR).
SP	Flip-flop used in single pulse control.
SPL	Single pulse level used in memory control.
SQ	Sequence counter, a binary counter used in external control.

Abbreviations	Description
ST	Sequence counter used in the read synchronizer, also known as the read control counter (RC).
STBD	A 2μsec strobe from the device. Received on bus line 24 (INBUS 24).
STBF	Strobe flip-flop used to drive bus line 24.
STOP	The code for "stop" in paper tape format.
STR	Signal from the operator's panel indicating that the START switch has been energized.
STRE	Start read signal which causes the read synchronizer ST-counter to leave the standby state.
STW	The START switch on the operator's panel.
SXBL	Select card location in memory indicated by (XB -1) for the current memory operation.
Synchronizer	Another term for external control.
SYNCH1	Synchronizing signal received from the paper tape punch. Received on bus line 12.
SYNCH2	Synchronizing signal received from the paper tape punch. Received on bus line 13.
S1YA	Subtract 1 from the YA-counter.
t	A 5ØØ KC timing pulse. Alternates with a p-pulse.
TCM	Transfer from the code converter to memory.
TCO	Transfer from the code converter to the output buffer register (OBR).
TIM	Transfer from the input buffer register (IBR) to memory.
TL1 - TL9	Timing levels 1 through 9. The memory address counter (MAC) contains the various counts required for magnetic tape timing.
TLM	The memory address counter contains its highest possible count.

Abbreviations	Description
TMO	Transfer from the memory buffer register to the output buffer register.
VY	Signal from the operator's panel which indicates that the verify mode of operation has been selected.
VYW	The VERIFY mode switch on the operator's panel.
WCF	Write control flip-flop used to drive bus line 11.
WCHK	The with-check signal which indicates that a checking operation is to be performed on the cards.
WLV	Write level sent to the memory control.
X1FC	Memory buffer register (MBR) bits 1 through 6 and 9 through 11 are each ZERO. Used in the Hollerith to Fieldata code converter.
X2FC	Memory buffer register (MBR) bit 12 or 13 is a ONE. Used in the Hollerith to Fieldata code converter.
XA	A modulo-6 counter in the memory address counter.
XB	A modulo-4 counter in the memory address counter.
XBXM	XB = XM. The final card location in memory for the current cycle is being addressed by the memory address counter (MAC).
XG	Flip-flop used in single pulse control.
XM	Register used to store card count when checking. cards.
YA	A modulo-7 counter in the memory address counter.
YB	A modulo-2 counter in the memory address counter.
$W	Time delay relay used with the switches on the operator's panel.

SECTION III

LINE PRINTER AND LINE PRINTER BUFFER

3.1 LINE PRINTER

The Shepard High-Speed Printer has no carriage as in the more conventional typewriters. Instead, there is one print hammer for each column of the page. The type face is on a drum which rotates at 900 rpm. Across one row of the drum are 120 "A"s, across the next row are 120 "B"s, etc., until, when we have gone around the entire periphery of the drum, all 64 characters have been found.

During the printing operation, the paper is positioned so that the line to be printed is opposite the print hammers. The drum rotates continuously.

As an example, let us assume that the row of "A"s on the drum is opposite the print hammers and it is desired to print "A"s in the 11th, 35th, and 97th columns of the line now being printed. If the 11th, 35th, and 97th hammer solenoids are impulsed, the hammer strikes the paper through the ribbon. The embossed "A"s on the print drum behind the paper are then transferred to the line of the page in columns 11, 35, and 97; then, the hammers return to rest while the drum continues to rotate. When the "B" row of the drum moves into position under the line of the page, the print hammers for the columns where "B"s are desired, if any, are then impulsed. This continues until all the characters on the periphery of the drum have passed under the print hammers; that is, until the drum has made one complete revolution. Of course, due to the finite travel time taken by the hammer to move from its rest position to the impact point, the actual impulsing must occur a finite time before the row of characters reaches the proper position.

simultaneously

In the MOBIDIC 7A Line Printer, 64 characters are provided around the periphery of the drum, one for each of the 64 possible Fieldata codes. Those codes which are not normally printing codes (for example, space, tabulate, etc.) were assigned special unique characters for use in the Verbatim Print-Out Mode, to be described later. The 64 characters are arranged in numerical order of their corresponding Fieldata code. **Refer to Table 3-1**. The 64 Six-Bit Fieldata Character Codes.

TABLE 3-1. THE 64 SIX-BIT FIELDATA CHARACTER CODES

The 64 six-bit Fieldata character codes are as follows. Where a code is not a printing character, its normal interpretation is given first, followed by the special character assigned to it for the Verbatim Print-Out Mode of operation.

000 000	Master space, ≈
000 001	No action, ∼
000 010	No action, ✱
000 011	Tabulate, ◇
000 100	Carriage return, ≺
000 101	Space, Δ
000 110	A
to	
011 111	Z
100 000	)
100 001	-
100 010	+
100 011	<
100 100	=
100 101	>
100 110	_
100 111	$
101 000	*
101 001	(
101 010	"
101 011	:
101 100	?
101 101	!
101 110	,
101 111	Stop, ⊕
101 000	0
to	
111 001	9
111 010	'
111 011	;
111 100	/
111 101	.
111 110	□
111 111	No action, <—

A special track on the printing drum sends out sync pulses for each character position of the drum. By simply counting these pulses with an ordinary binary counter, the Fieldata code for the characters is obtained in the order in which they appear under the hammers. Another sync channel on the print drum sends out a pulse once per revolution to keep the drum and counter in synchronism.

When the drum has completed one complete revolution, the printing of the line is complete; and we may send a signal to the printer to feed the paper. The paper will feed continuously while this signal is on. The printer will send out a pulse for every line fed, which also is sent to a counter so that the buffer may keep track of the vertical position of the paper. When the required number of line feeds has been completed, it may be stopped by simply terminating the signal from the buffer. A third signal will issue from the printer until the paper has had time to come to a stop. This will be used by the buffer to inhibit the printing of the next line until the paper is at rest.

3.2 LINE PRINTER BUFFER

The Shepard High-Speed Line Printer used in MOBIDIC 7A simultaneously prints on 120 columns of a line. Since the data to be printed comes from the In-Out Converter of the MOBIDIC system in the form of characters, of 6 bits each, it is evident that some buffering device is needed. In addition, some of the characters are for control purposes, and some means must be provided to transform these into suitable control signals for the printer.

The High-Speed Line Printer, with its buffer unit, must be able to perform in various modes of operation. First, the so-called Programmed Line-Feed mode requires that the first character received for each line of print must be treated as a special Line-Feed Character. Before the line is printed, the printer must feed the number of lines denoted by the 6 bits of this character.

The second possible mode of operation is the Automatic Line-Feed mode, where all characters received are treated as printing or control characters. After the line is printed, a single or double line feed is always performed.

The third possible mode of operation is the Verbatim Print-Out Mode. Here, all characters received are printed, and none are treated as control characters.

Since there is no possibility of a TAB, SPACE, or CARRIAGE RETURN code, the only format control possible is through the 120-to-120 plugboard between the buffer and the printer. Each line is followed by a double line feed.

The 120-to-120 plugboard is a means of juxtaposing the connections between the 120 columns of the memory and those of the printed page. The characters read in are stored sequentially in the buffer memory. The plugboard allows us automatically, without the need for SPACE or TAB characters, to put the printed data in spaced columnar form, and even to rearrange the order of the columns at will. SPACE and TAB codes will operate in the memory similarly to the way they do on an ordinary typewriter.

Just as on a typewriter, the receipt of a TAB code causes the column selector counter (PA) to alter its selection to the next higher column previously chosen as a "tab" position. The choice of the tab positions must be made manually by an operator.

It is possible to select any 15 memory locations as tab positions. There are 120 TAB PLUGS, each wired to specify one of the memory locations from 2 to 120, and the imaginary 121st. The latter is used to force the buffer to sense the "end-of-line" or "carriage return" condition.

Each plug has marked, on its back, the number corresponding to the memory location it represents.

To receive these plugs, there is a row of 15 sockets. The 15, or less, selected plugs are inserted into the sockets so that the labels are in increasing numerical order from left to right. As tab codes are received, the tab plugs are selected in turn to specify the memory location to which the buffer tabulates. Thus, if the tab plugs read, in order, 22, 33, 57, 79, 99, the first tab of the line is to column 22 of the memory. The second is to 33, etc. If, for example, there were 24 characters received after the second tab, the next character should go into column 57, the location of the next tab. If the next input character were the TAB code, we would ignore column 57, and tab to column 79; hence, if the plugs are arranged in increasing order, every tab code would cause the tabulation to the next higher memory column to the one we were about to use. IF NO PLUG IS LOCATED IN THE SOCKET WHOSE TURN IT IS, A TAB CODE WILL SET THE BUFFER ALARM.

If a sixteenth tab code is received for any line, the tabulation takes place to the imaginary 121st memory location, to force the buffer to the "end-of-line" condition.

The reading in of data to the memory will be terminated by a CARRIAGE RETURN or a STOP code. The former will just stop the reading-in process and start the print-out and line-feed phases. The STOP code, on the other hand, will result in the printing operation being followed by a paper feed to the top of the next page, and then a halt until the operator presses the MANUAL RESET button. This last feature should be useful for giving the operator instructions. When he has finished carrying them out, he presses the button, and the printer resumes operation.

A sixteenth socket will be present among the controls, which will take the same plugs as the 15 tab sockets. Putting a plug in this socket will effectively "shorten" the length of the buffer memory to the number of positions indicated on the plug less one. In other words, inserting the plug labelled 75 into this socket will make position 74 the last one of the memory.

3.2.1 Line-Printer Buffer Block Diagram

3.2.1.1 Memory Plane

Figure 3-1 is the block diagram of the Line-Printer Buffer. The memory consists of a 16 by 64 core plane which is wired to perform the following functions:

1. Write the contents of the static In-Out register (ICR and OCR combined) into the column specified by the column counter (PA).
2. Read the contents of the column specified by the column counter (PA) into the 14-bit section of the in-out static register (ICR). Note that it is impossible to read from the memory into OCR, anything present in the bottom two rows of the memory during this type of reading would be discarded.
3. Read the contents of 120 of the 128 cores in the bottom two rows of the memory to 120 output lines.

3.2.1.2 In-Out Static Register (ICR)

Data received from the input busses is stored temporarily in the 14-bit Static Register, ICR. Each character occupies 7 bits of the register, the 7th

NOTE:

C* REPRESENTS A BIT FOR BIT COMPARATOR

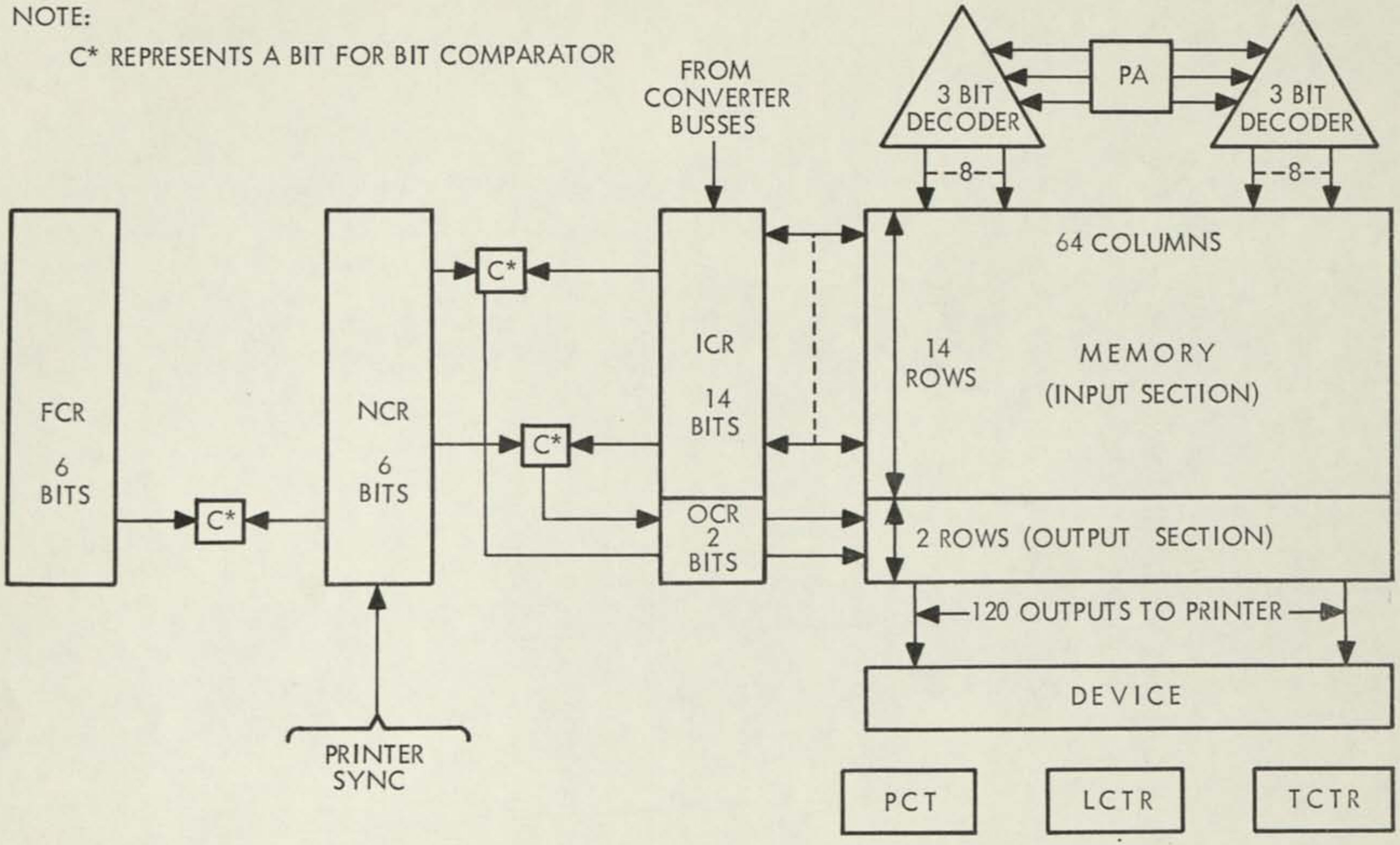


Figure 3-1. Block Diagram of the Line Printer Buffer

(control) and the 8th (parity) bit of the incoming character having been combined into a single parity bit. The 14 rows of the memory are treated as two 7-row sections. The top 7 rows of the 64 columns hold the first 64 memory positions while the next 7 rows hold positions 65 and up. In other words, positions 1 and 65 of the memory are physically in the first column (position 1 in rows 1 through 7, position 65 in rows 8 through 14). Likewise, positions 2 and 66 are in column two, etc. If the character is to be stored in one of the first 64 positions of the memory, it is kept in the upper 7-bit positions of the ICR into which it was read and, hence, goes into the upper section of the memory. If the character is to be stored in the 65th, or later, position of the memory, it is shifted from the bit positions 1-7 to bit positions 8-14 before being written into the memory. When a column of the memory is read out, therefore, two character positions are written into the Static Register. The role of the OCR section of the register will be treated later.

3.2.1.3 Next Character Register (NCR)

A sync pulse comes in from the printer for every character which passes under the print hammers. In order to keep track of which character is under the hammers, therefore, it is necessary to continually count these pulses. This is done in the NCR (Next Character Register) Counter.

When in the process of printing the information on the paper, we compare the present contents of the NCR with each character position of the memory. Where the identical character code is found, it is cleared from the memory, and a "1" is stored in the corresponding bit position of the Output Section of the memory (the last two rows); hence, when all 64 columns of the memory have been read out, compared, and written back in, the Output Section contains a pattern of 0's and 1's. Each "1" indicates a memory character position which held the character now coming under the print hammers. When the print wheel sync pulse arrives, it causes the readout of the entire Output Section of the memory to the print hammer solenoids. This sync pulse also increments the NCR register to indicate the next character to be searched for in the memory.

3.2.1.4 Output Static Register (OCR)

The static register which holds the "1" to be written into the bottom 2 rows of the memory after a successful comparison described in the previous paragraph is the 2-bit section of the In-Out Register, labelled OCR. It will be noted that the setting of OCR_1 to 1 is always accompanied by the clearing of ICR_{1-7} (the upper half of ICR) in the mechanization. Similarly, the setting of OCR_2 is always accompanied by the clearing of ICR_{8-14}.

3.2.1.5 First Character Register (FCR)

In order to obtain maximum printing speed, the actual print out is started as soon as possible after the buffer memory has received its last character. As a result, the print-out phase can start anywhere on the periphery of the drum; therefore, note must be taken of which character was the first one printed during any print-out cycle, so that when this same character is again reached, we know the print-out cycle for that line is complete. This is done at the start of each print-out cycle by storing the current contents of the NCR register in the FCR register. After each character is printed and the NCR incremented to its new value, we compare the contents of these two registers. Coincidence ends the print-out cycle.

3.2.1.6 Tab Counter (TCTR)

The tab counter keeps track of which tab position is the next one to be used. Fifteen plugs on the control panel each indicate one of the 120 columns of memory. Initially, TCTR is set to ZERO and therefore selects the first of these plugs. Whatever code is wired into that plug is selected as the first "tab position"; and as long as it is active, a Fieldata "Tab" code causes the insertion of this quantity into the memory column number (PA). The tab counter is then incremented by 1, causing the selection of the next tab plug.

3.2.1.7 Line Counter (LCTR)

The line counter keeps track of how many lines have been fed from the top of the page. At the top of a page, LCTR contains a ZERO. A set of six toggle switches indicates the number of lines within a page of printing. When a comparison is made between the LCTR and this set of switches, it indicates that we have just passed the bottom line of the page. We use the setting of a set of 2 toggle switches to cause the printer to line feed from one to four times more. At the end of this operation, we are again at the top of a page; and the Line Counter indicates ZERO.

3.2.1.8 Main Sequence Counter (PCT)

The PCT counter is a collection of 7 flip-flops which together control the operation of the buffer. It is not a counter in the usual sense of the word. The various flip-flops are operated, one at a time where possible, to give a succession of states which controls the actions of the various other sections of the buffer.

3.2.2 General Flow Chart of the Line Printer Buffer

Out of the 2^7, or 128 possible states, 86 are used. These 86 are divided into 5 phases, as shown in the General Flow Diagram of Figure 3-2. The 17 states which are used to accomplish the reading-in phase are labelled states 500 to 516, inclusive, and form the 500 series. Similarly, the 600 series of states are used to accomplish the printing out; the 700 series, the automatic single or double line feeding; the 800 series, the line feed to the top of the next page and halt; and the 900 series, the programmed line feeding.

In addition to the operational designation of the various states, each state is also designated by the numerical equivalent of its configuration, expressed in octal. These 7 flip-flops can give rise to states 000 to 177 in this system. Each configuration is listed in both ways in the mechanization. (500)(000) indicates that the state is No. 500, and corresponds to the 7 bits being 0 000 000. (503)(025) denotes state No. 503, where the configuration is 0 010 101.

Figure 3-2 also shows that, when we are operating in the Programmed Line Feed mode (denoted by PLFM), we line feed BEFORE we print out the line, while in the Automatic Line-Feed mode (denoted by PLFM'), we only line feed AFTER the print out. Regardless of which of these two modes we are in, if we have received a STOP code during the last reading-in operation, (denoted by STOP), we must feed paper until we reach the top of the next page and then wait for the MANUAL RESET to be pressed. This feeding is done after the print out.

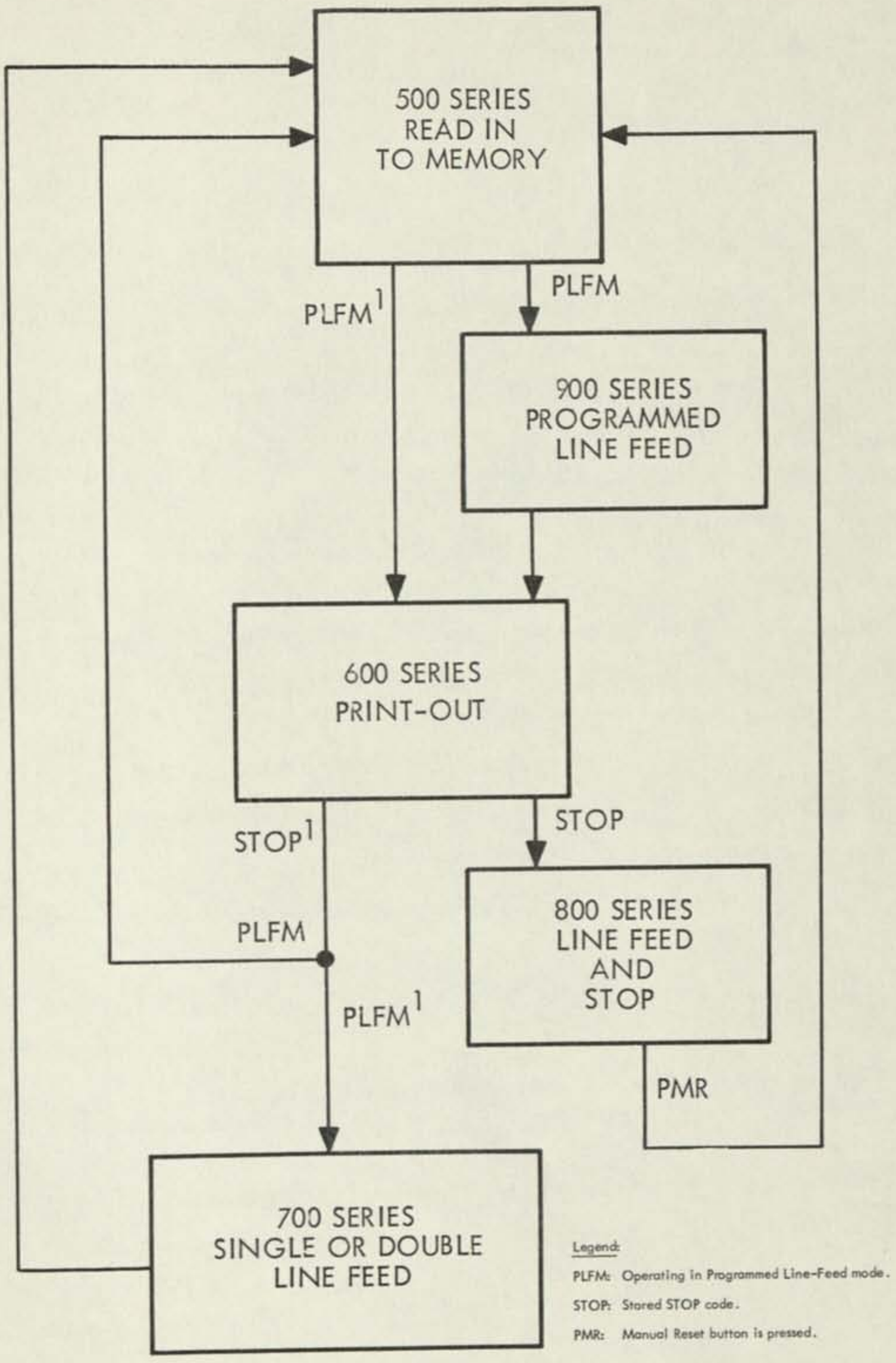


Figure 3-2. General Flow Chart of the Line Printer Buffer

3.2.3 Detailed Flow Charts of the Line-Printer Buffer

Figures 3-3 to 3-9 are the detailed operational flow charts for the Line-Printer Buffer. Note that there is continual reference made to the state involved in each part of the operation. The operation of the Line-Printer Buffer is broken down into 7 convenient sections, each presented in one of the flow charts:

3-3.	Standby, and Reading in line-feed character
3-4.	Reading in characters
3-5.	Responding to read-in control characters
3-6A, B.	Printing out (including 4-16B. Cycling through memory)
3-7.	Automatic line feeding
3-8.	Page feed and stop
3-9.	Programmed line feed

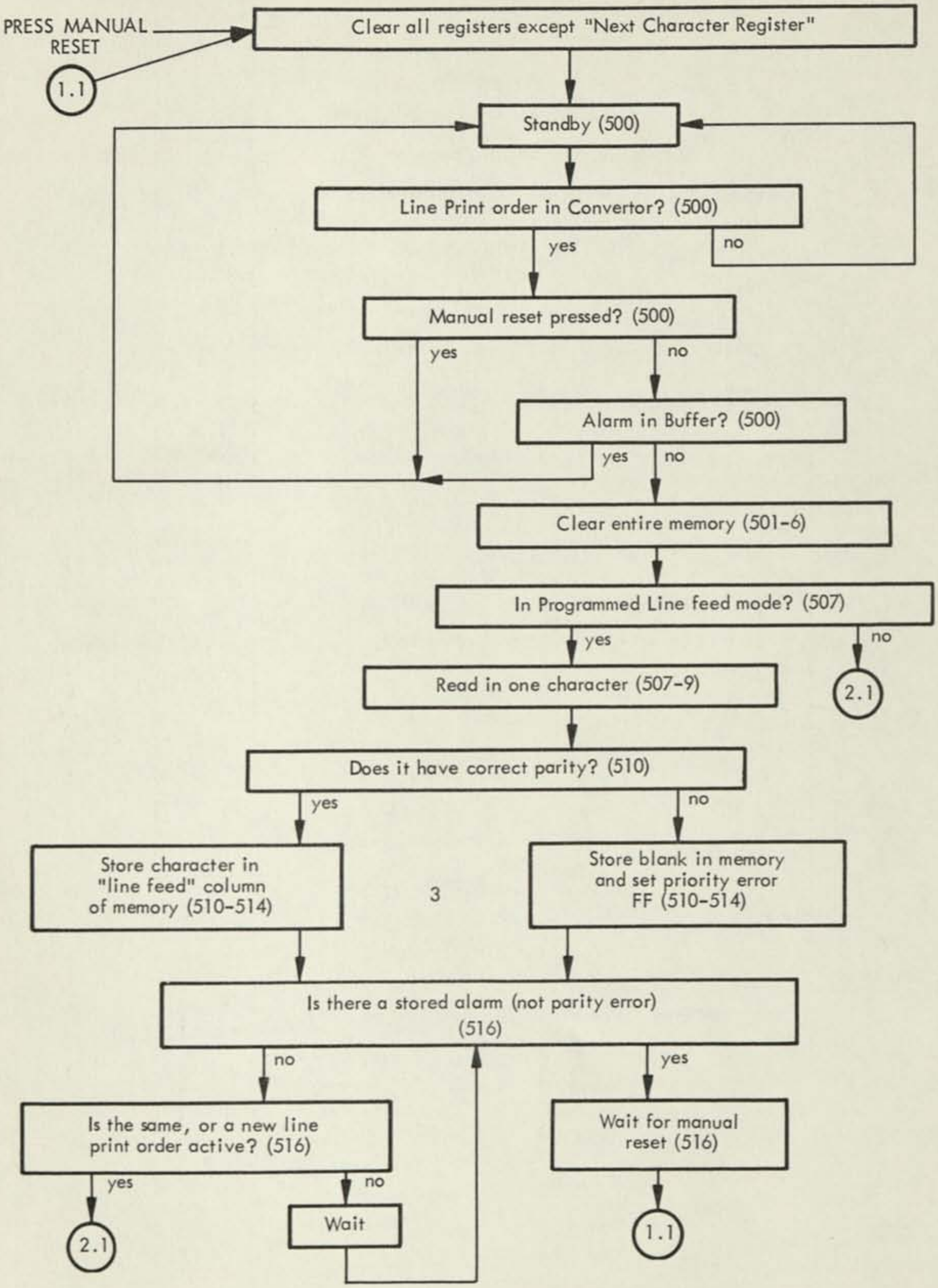

PRESS MANUAL RESET
1.1
Clear all registers except "Next Character Register"
Standby (500)
Line Print order in Convertor? (500)
yes
no
Manual reset pressed? (500)
yes
no
Alarm in Buffer? (500)
yes
no
Clear entire memory (501-6)
In Programmed Line feed mode? (507)
yes
no
2.1
Read in one character (507-9)
Does it have correct parity? (510)
yes
no
Store character in "line feed" column of memory (510-514)
3
Store blank in memory and set priority error FF (510-514)
Is there a stored alarm (not parity error) (516)
no
yes
Is the same, or a new line print order active? (516)
yes
no
2.1
Wait
Wait for manual reset (516)
1.1

Figure 3-3. Flow Chart, Standby and Reading In Line Feed Character

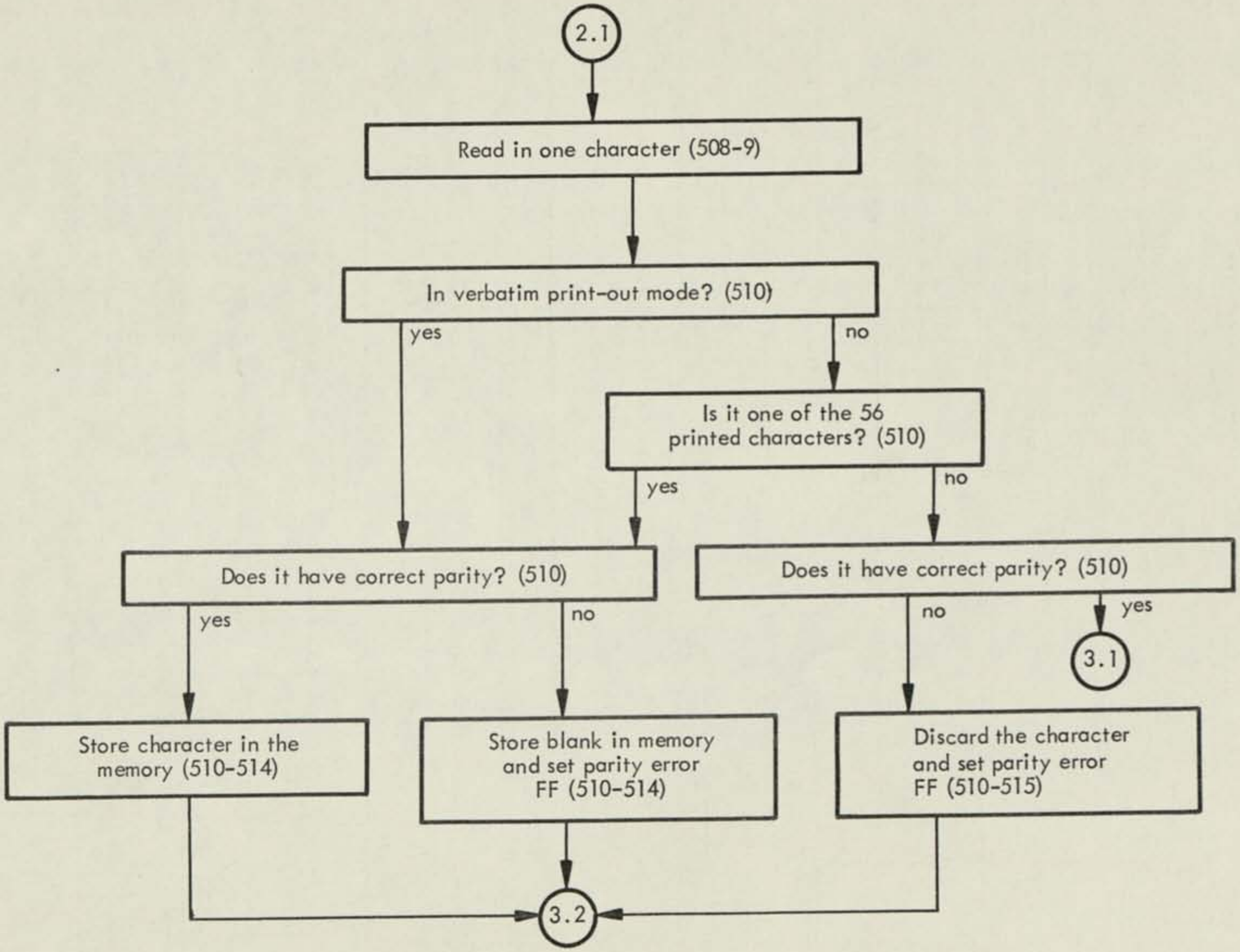

2.1
Read in one character (508-9)
In verbatim print-out mode? (510)
yes
no
Is it one of the 56 printed characters? (510)
yes
no
Does it have correct parity? (510)
Does it have correct parity? (510)
yes
no
no
yes
3.1
Store character in the memory (510-514)
Store blank in memory and set parity error FF (510-514)
Discard the character and set parity error FF (510-515)
3.2

Figure 3-4. Flow Chart, Reading In Data

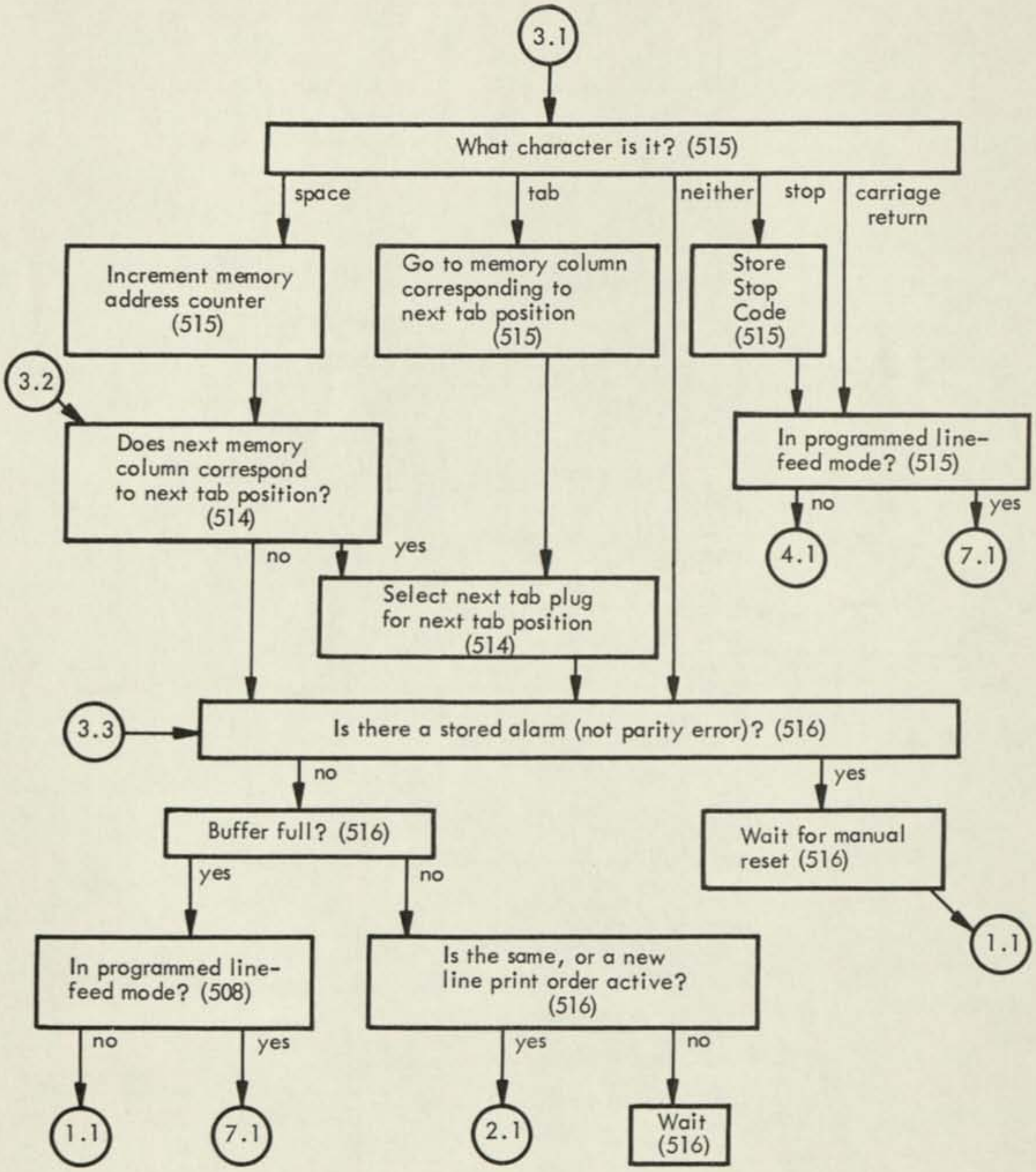

3.1
What character is it? (515)
space
tab
neither
stop
carriage return
Increment memory address counter (515)
Go to memory column corresponding to next tab position (515)
Store Stop Code (515)
3.2
Does next memory column correspond to next tab position? (514)
In programmed line-feed mode? (515)
no
yes
4.1
7.1
no
yes
Select next tab plug for next tab position (514)
3.3
Is there a stored alarm (not parity error)? (516)
no
yes
Buffer full? (516)
Wait for manual reset (516)
1.1
yes
no
In programmed line-feed mode? (508)
Is the same, or a new line print order active? (516)
no
yes
yes
no
1.1
7.1
2.1
Wait (516)

Figure 3-5. Flow Chart, Responding to Control Character

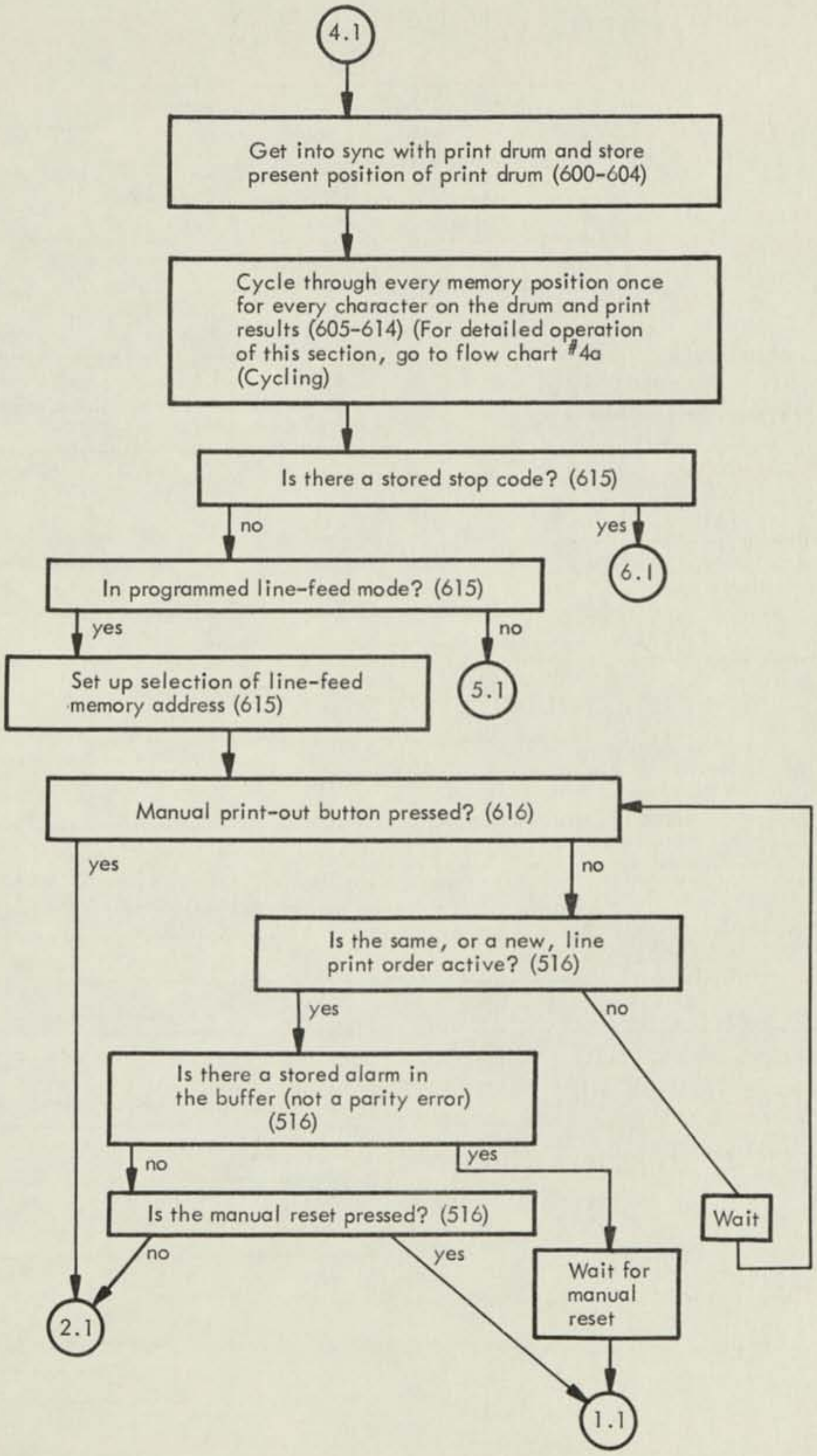

4.1
Get into sync with print drum and store present position of print drum (600-604)
Cycle through every memory position once for every character on the drum and print results (605-614) (For detailed operation of this section, go to flow chart #4a (Cycling)
Is there a stored stop code? (615)
no
yes
6.1
In programmed line-feed mode? (615)
yes
no
5.1
Set up selection of line-feed memory address (615)
Manual print-out button pressed? (616)
yes
no
Is the same, or a new, line print order active? (516)
yes
no
Is there a stored alarm in the buffer (not a parity error) (516)
no
yes
Is the manual reset pressed? (516)
no
yes
Wait
Wait for manual reset
2.1
1.1

Figure 3-6A. Flow Chart, Print Out

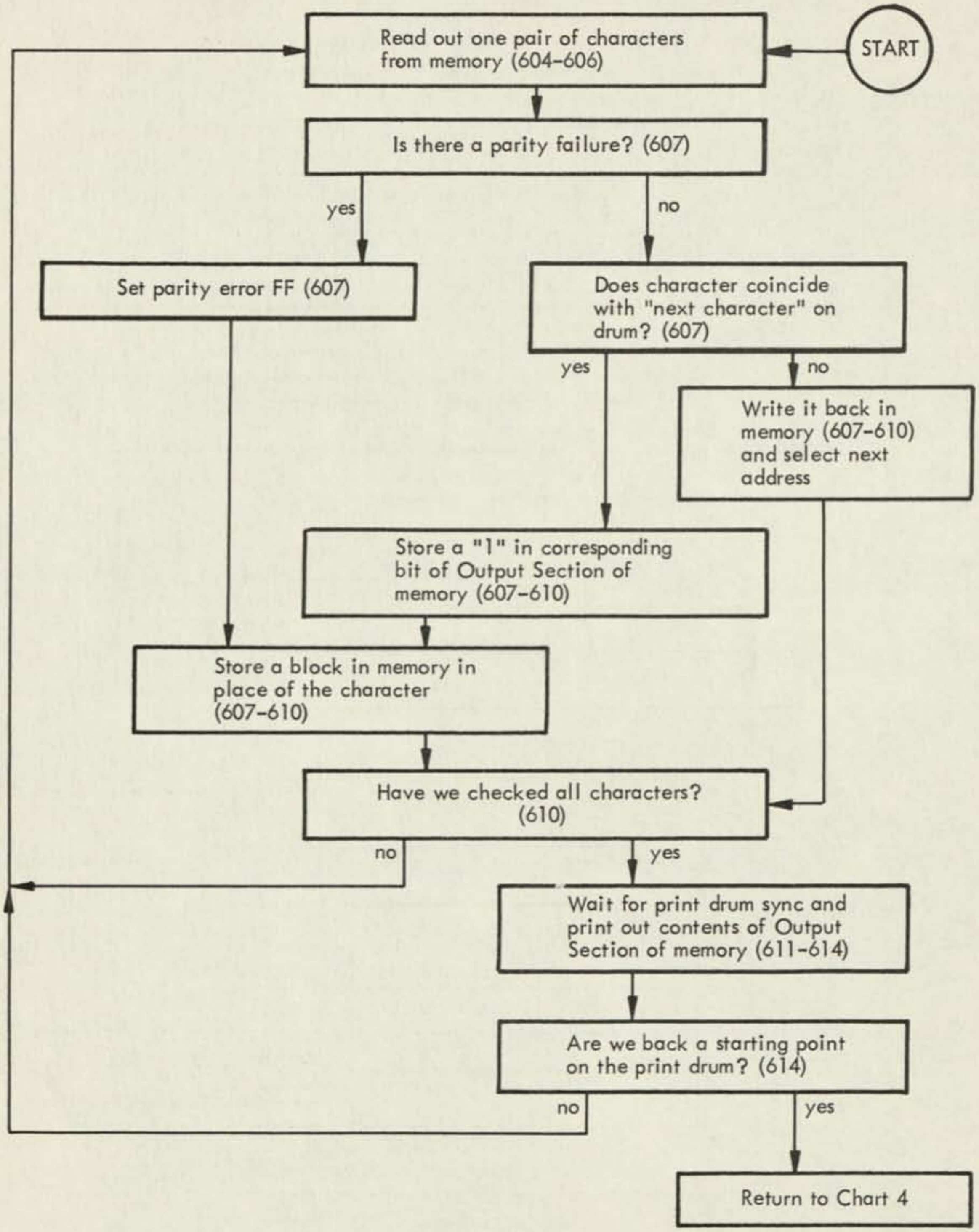

START
Read out one pair of characters from memory (604-606)
Is there a parity failure? (607)
yes
no
Set parity error FF (607)
Does character coincide with "next character" on drum? (607)
yes
no
Write it back in memory (607-610) and select next address
Store a "1" in corresponding bit of Output Section of memory (607-610)
Store a block in memory in place of the character (607-610)
Have we checked all characters? (610)
no
yes
Wait for print drum sync and print out contents of Output Section of memory (611-614)
Are we back a starting point on the print drum? (614)
no
yes
Return to Chart 4

Figure 3-6B. Flow Chart, Cycling

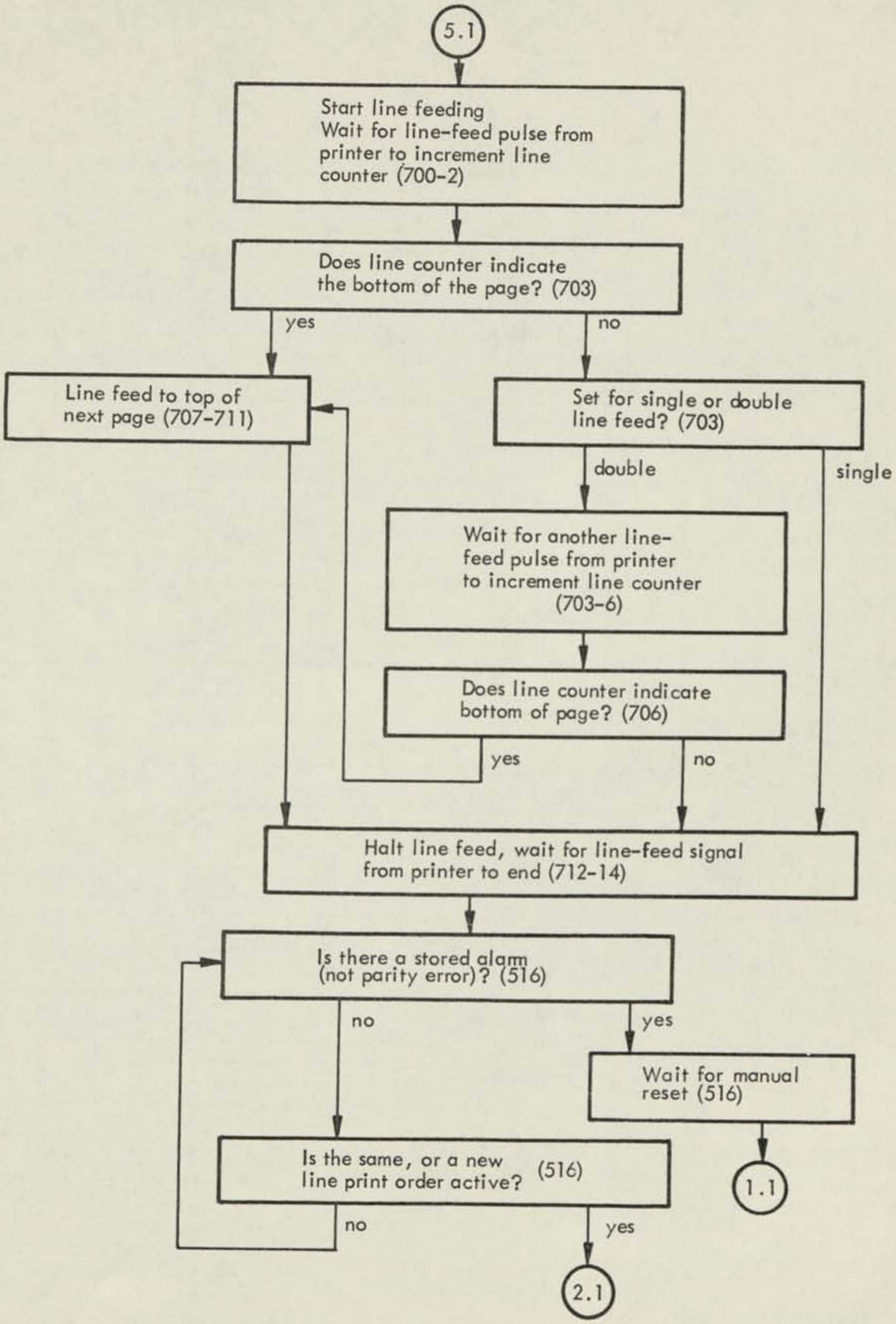

5.1
Start line feeding
Wait for line-feed pulse from printer to increment line counter (700-2)
Does line counter indicate the bottom of the page? (703)
yes
no
Line feed to top of next page (707-711)
Set for single or double line feed? (703)
double
single
Wait for another line-feed pulse from printer to increment line counter (703-6)
Does line counter indicate bottom of page? (706)
yes
no
Halt line feed, wait for line-feed signal from printer to end (712-14)
Is there a stored alarm (not parity error)? (516)
no
yes
Wait for manual reset (516)
Is the same, or a new line print order active? (516)
no
yes
1.1
2.1

Figure 3-7. Flow Chart, Automatic Line Feed

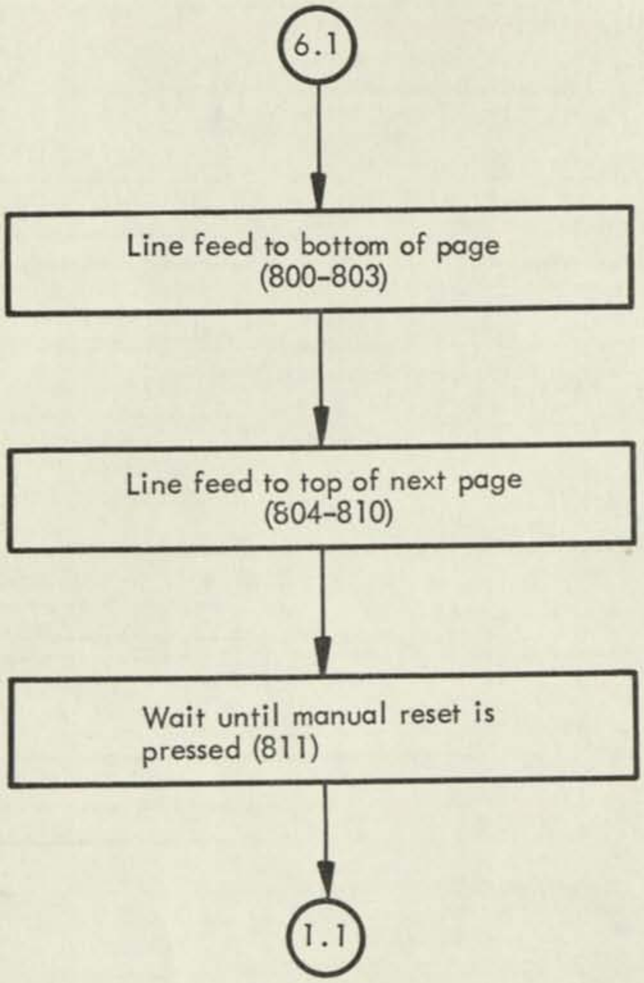

6.1
Line feed to bottom of page (800-803)
Line feed to top of next page (804-810)
Wait until manual reset is pressed (811)
1.1

Figure 3-8. Flow Chart, Page Feed and Stop

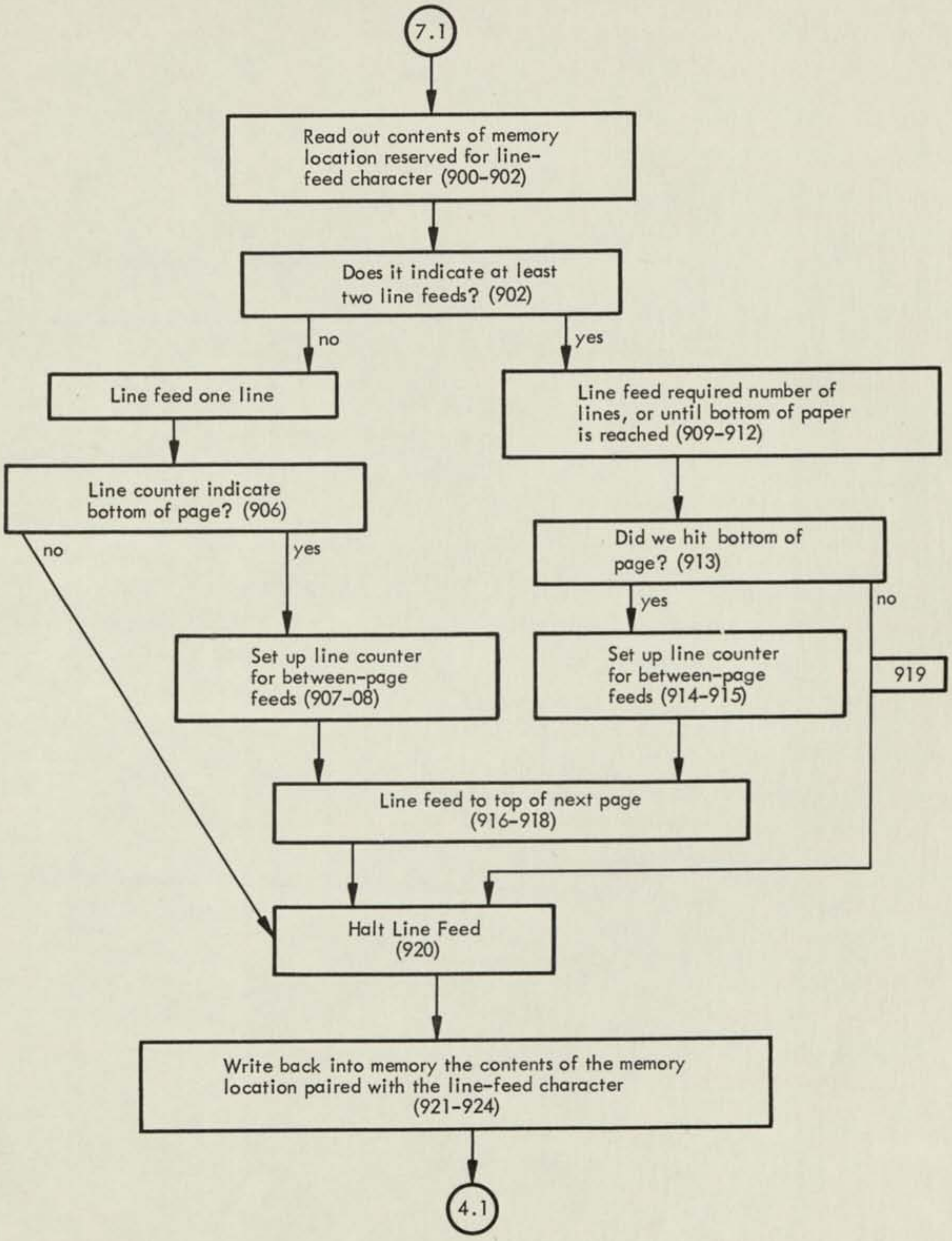

7.1
Read out contents of memory location reserved for line-feed character (900-902)
Does it indicate at least two line feeds? (902)
no
yes
Line feed one line
Line feed required number of lines, or until bottom of paper is reached (909-912)
Line counter indicate bottom of page? (906)
no
yes
Did we hit bottom of page? (913)
yes
no
Set up line counter for between-page feeds (907-08)
Set up line counter for between-page feeds (914-915)
919
Line feed to top of next page (916-918)
Halt Line Feed (920)
Write back into memory the contents of the memory location paired with the line-feed character (921-924)
4.1

Figure 3-9. Flow Chart, Programmed Line Feed

SECTION IV

CARD READER-PUNCH AND CARD READER-PUNCH BUFFER

4.1 GENERAL

The IBM 533 Card Reader-Punch is the card device used with MOBIDIC 7A. The reader and punch sections are essentially two independent devices, each processing cards on a row-by-row basis; the reader operates at 200 cards per minute, and the punch at 100 cards per minute. In order to match this low-speed operation with the electronic-speed operation of the In-Out Converter, a buffer had to be provided.

The card buffer had to permit two different types of operations:

1) It must be possible to punch data in such a way that, when read back in the proper way, the original data is retrieved. In this mode, it is not necessary to be able to punch or read any given hole on the card, and the exact arrangement of the data on the cards is not important.

2) It must also be possible to punch or read any given hole on the card and have that information available to the user. In this mode, it is not important if certain bits get lost in the punching operation or if some bits are inserted into the data being read from the cards. This mode would be used to produce cards punched in the Hollerith or other code foreign to the MOBIDIC computer.

The design of the Card Buffer for MOBIDIC 7A system meets both requirements. In punching, 14 characters are used for each row of the card. In the Interpret Sign Mode, these 14 characters represent 2 complete words. Since the converter has taken the sign bits of each of the words and expanded them to full 6-bit characters, more bits are punched than originated in the Central Processor; but, in any event, no bit from the processor failed to reach the card. Since only 80 bits can fit in a row, and the 14 characters make 84, it is necessary to lose 4 bits from each 14 characters. Naturally, we chose four of the bits which were inserted by the Converter-namely, the high-order four bits of the first character of each row. In other words the data is arranged on the cards as follows, assuming a 1-to-1 plugboard:

Character 1, bits 2 and 1: column 1 and 2, respectively

Character 2, bits 6 to 1: column 3 to 8, respectively

Character 3, bits 6 to 1: column 9 to 14, respectively
(etc., until...)
Character 14, bits 6 to 1: column 75 to 80, respectively

Conversely, when we read cards alphanumerically in the Interpret Sign Mode, we take each row of the card, break it up into 14 characters and insert bits 1100 to the left of the first two btis to complete the first character. If we had punched the data in the same Interpret Sign Mode, these four bits would be the ones stripped off; therefore, we have reconstructed the data exactly as it was sent out by the Converter.

The punching and reading of cards alphanumerically in the Interpret Sign Mode satisfied the first of the two required types of operation. Data may be sent to the cards and retrieved unaltered. However, to obtain this ability, we have punched bits on cards that never existed in the processor; but this is of little consequence.

The second requirement is satisfied by punching and reading cards in the non-interpret Sign Mode. When punching in this mode, the Converter takes each word and sends out six 6-bit characters, having ignored the sign bit. The buffer, as before, discards the high-order four bits of the first character of each row; thus, data is lost, but no extraneous bits are inserted. Since the fourteen characters of a row now constitute 2 1/3 full words (or 7 full words for every three rows of a card), the programmer must take cognizance of the fact that he will lose the high-order four bits of each of the following : word 1, character 1, word 3, character 3; and word 5, character 5, out of every seven words he sends to the card punch in this mode. In return for this, he now has the ability to punch any hole on any card, and hence can punch the card in Hollerith or any other code system.

Conversely, every row read in this mode will produce 2 1/3 words in the processor, with four bits of that group being meaningless. The programmer will be able, however, in this Non-interpret Sign Mode, to read any 12 x 80 hole card, no matter what device punched it, and regardless of what code was used. Every hole on the card will go into the memory as a known bit, available to any required decoding routine.

Punching of cards must be done in the alphanumeric mode.

During either the reading or punching operation, if the corresponding hopper becomes empty, the operation of the card system will halt; but no alarm will be energized. This gives the operator the opportunity to do one of the following procedures:

1) He may insert more cards in the hopper and then press the RESET button. This permits the Read or Punch operation to proceed.

2) If, in the case of the Read operation, there are no more cards to be processed, the operator merely presses the RESET button. This enables the reading of the last two cards (which have left the hopper but have not yet reached the reading station). When these two cards have been read, the buffer will forcibly terminate the order in the Converter. The buffer sends blockmarks continuously until the order is counted down in the Converter. This again resets the buffer to the standby state.

4.2 BLOCK DIAGRAM

4.2.1 Row Register (R1-R80)

Figure 4-1, is the block diagram of the Card Reader and Punch Buffer. The heart of the buffer is the 84-bit Row Register which is the temporary storage for the 80 bits of data or each row read or to be punched.

When reading, the 80 bits of a row are read in at once and sent to the Converter one character (6 bits) at a time. When punching, the Row Register is filled one character at a time from the Converter and read out, all 80 bits at once to the Punch solenoids.

4.2.2 Character Selection Counter and Decoder (CSC, CS)

The Character Selection Counter is a four-bit binary counter which is used to select the character from the Row Register to be read out to, or read in from, the In-Out Converter. Its associated decoder has the 16 outputs, CS_0 to CS_{15}, which do the actual selection. Since there is at most 14 characters read in or out of any row of a card, CS_{15} is used only to indicate that the entire row has been received or sent out, whichever is applicable.

card is left blank. A new punch order would not take effect until the next card. It will soon be apparent that no alphanumeric read operation can terminate in the middle of a card.

The punch- and read-octal orders for cards specify the number of words to be punched on or read from cards. The read alphanumeric order, on the other hand, specifies the number of cards to be read. In order to accomplish this, the converter is set up as if the card buffer were a magnetic-tape device, for the RAN order ONLY. The buffer then sends dummy start and end-block marks before and after every card read, thereby allowing the converter to count the number of cards read, rather than the number of words received. The counting of cards, for this order, rather than words, is the reason that RAN orders terminate only at the end of a card, never in the middle.

During either the reading or punching operation, if the corresponding hopper becomes empty, the operation of the card system will halt; but no alarm will be energized. This gives the operator the opportunity to do one of the following procedures:

1) He may insert more cards in the hopper and then press the RESET button. This permits the Read or Punch operation to proceed.

2) If, in the case of the Read operation, there are no more cards to be processed, the operator merely presses the RESET button. This enables the reading of the last two cards (which have left the hopper but have not yet reached the reading station). When these two cards have been read, the buffer will forcibly terminate the order in the Converter. In the Read-Octal order, this is accomplished by the buffer by sending the STOP code to the Converter, which terminates the order in the latter unit, that in turn puts the buffer back into the standby state. For the Read Alphanumeric order, however, the buffer sends blockmarks continuously until the order is counted down in the Converter. This again resets the buffer back to the standby state.

4.2 BLOCK DIAGRAM

4.2.1 Row Register (R1 - R80)

Figure 4-1 is the block diagram of the Card Reader and Punch Buffer. The heart of the buffer is the 80-bit Row Register which is the temporary storage for the 80 bits of data of each row read or to be punched.

4.2.3 Parity Circuits

The parity circuit has two sections: one forms the correct parity before sending the character to the Converter, the other monitors the data input lines to make sure that the parity of input data is correct.

For punching, the parity circuit checks that each character input has an even parity. For reading, however, the parity circuit sends out on the 8th bus the bit which would give the character the proper parity.

4.2.4 Control Sequence Counter (CT)

The Card Buffer has been designed as a sequential switching circuit, using the technique devised by D. A. Huffman, and described by others, especially the switching-circuit books by Humphrey and by Caldwell. This technique analyzes the operation of a device by means of flow charts and state diagram, and then devises a "counter" which will step through the states in the required manner. The CT Counter is that counter in the Card Buffer. The sequence of operations is controlled by CT.

At any one instant, the Buffer is in some "state". This state is labelled in two ways: 1) A sequence number (100, 101, 102, 103, . . .) essentially in the order in which the states occur, and 2) By the octal equivalent of the number in the counter, if the various flip-flops are arranged in a given order. For example, if state 109 corresponds to the counter's being in 110 001, we label that state as 31, as well as 109. The sequential labels are of three decimal digits, 000, 100 series, and 200 series. 000 is the standby state; 100 series is for reading; and the 200 series is for punching. The configuration label is of two octal digits, from 00 to 77. In the mechanization, both labels are generally used for each state; and the states are listed in order of their sequential label.

4.3 FLOW CHARTS

4.3.1 Standby Flow Chart

The first of the operational flow diagrams, Figure 4-2, shows the standby state, in which the buffer awaits the punch or card read alphanumeric order, to be present in the converter.

When reading, the 80 bits of a row are read in at once and sent to the Converter one character (6 bits) at a time. When punching, the Row Register is filled one character at a time from the Converter and read out, all 80 bits at once to the Punch solenoids.

4.2.2 Character Selection Counter and Decoder (CSC, CS)

The Character Selection Counter is a four-bit binary counter which is used to select the character from the Row Register to be read out to, or read in from, the In-Out Converter. Its associated decoder has the 16 outputs, CS_0 to CS_{15}, which do the actual selection. Since there is at most 14 characters read in or out of any row of a card, CS_{15} is used only to indicate that the entire row has been received or sent out, whichever is applicable.

4.2.3 Parity Circuits

The parity circuit has two sections: one forms the correct parity before sending the character to the Converter, the other monitors the data input lines to make sure that the parity of input data is correct.

For punching, the parity circuit checks that each character input has an even parity. For reading, however, the parity circuit sends out on the 8th bus the bit which would give the character the proper parity; even for the Read Octal, and odd for the Read Alphanumeric operations.

4.2.4 Control Sequence Counter (CT)

The Card Buffer has been designed as a sequential switching circuit, using the technique devised by D. A. Huffman, and described by others, especially the switching-circuit books by Humphrey and by Caldwell. This technique analyzes the operation of a device by means of flow charts and state diagram, and then devises a "counter" which will step through the states in the required manner. The CT Counter is that counter in the Card Buffer.

The sequence of operations is controlled by CT, and the action of this counter is shown in detail in the mechanization, as is the action of all other registers and signals.

4.3.2 Card Read Alphanumeric Flow Chart

Figure 4-3 is the operational flow chart for the card read alphanumeric operation. Note that the read alphanumeric operation includes the sending of Start Blockmarks at the beginning of each card and End Blockmarks at the end. When reading alphanumeric, then, the buffer will look like magnetic tape and each card will be treated as a block.

4.3.3 Card Punching Flow Chart

Figure 4-4 shows the detailed operation of the buffer during the punching mode of operation.

During the punching operation, if the order goes down in the middle of the card, the row that has been received by the buffer will be punched; then, the card will be fed out without further punching; and the buffer will return to the standby state.

4.4 AUXILIARY CARD BUFFER

The auxiliary card buffer provides error checking capability for either off-line or on-line card systems. It is not an independent device, but rather supplements and is indirectly controlled by the present card buffer. The actual checking operation will be done in either the off-line control unit or the central processor, depending upon the type of installation.

4.4.1 Read Operation (Off-Line)

Cards are read by the IBM 533 card reader-punch by passing them through two read stations. The auxiliary card buffer is connected to the first read station, and the main card buffer to the second. When one card is being read at the second station, the following card is simultaneously read at the first. Multiplexing the two reads is accomplished by utilizing the synchronizing signals from the card machine.

Characters from the auxiliary buffer and the main buffer will be transmitted to the receiver in identical formats over the same set of data and strobe

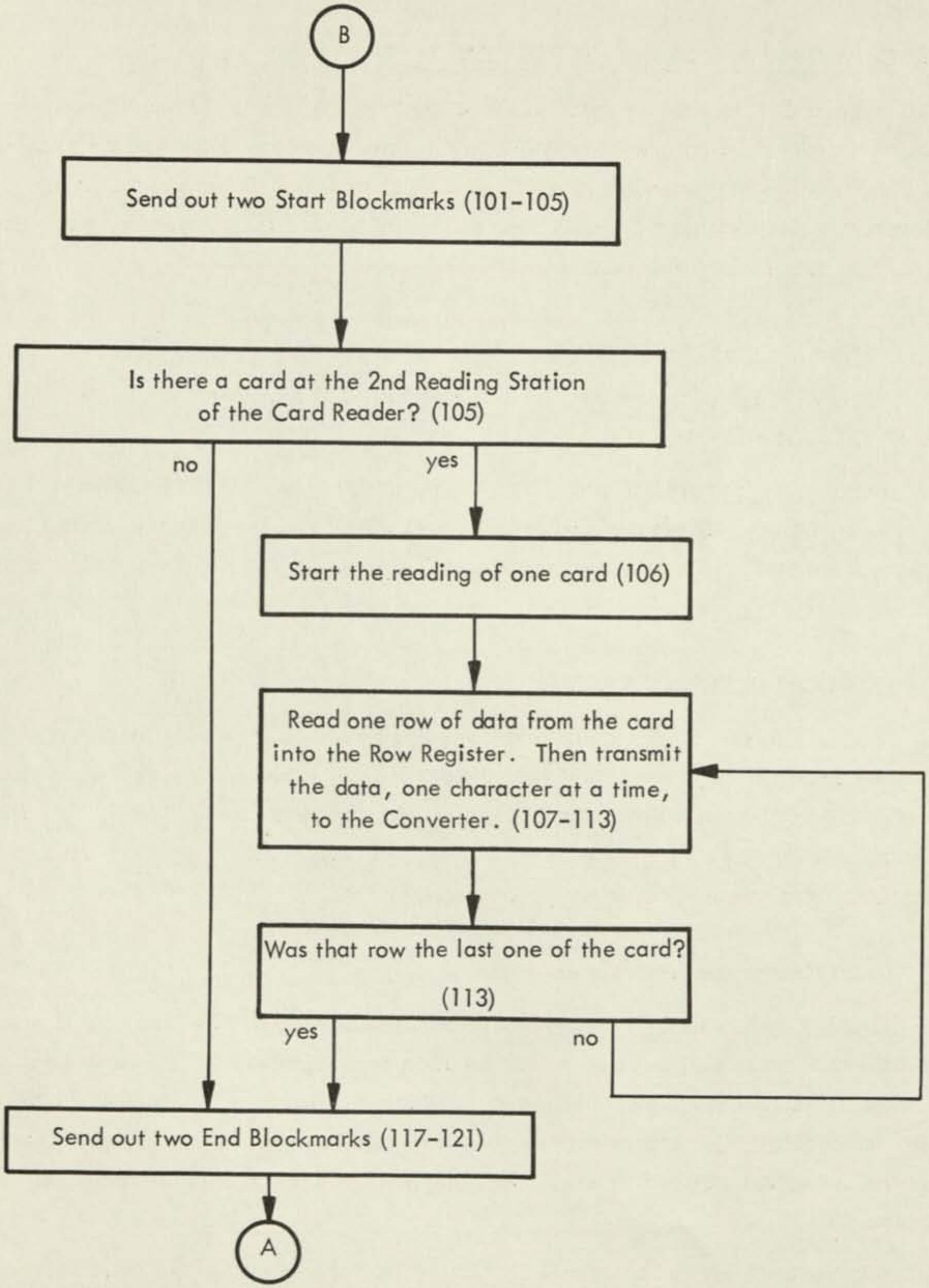

B
Send out two Start Blockmarks (101-105)
Is there a card at the 2nd Reading Station of the Card Reader? (105)
no
yes
Start the reading of one card (106)
Read one row of data from the card into the Row Register. Then transmit the data, one character at a time, to the Converter. (107-113)
Was that row the last one of the card? (113)
yes
no
Send out two End Blockmarks (117-121)
A

Figure 4-3. Operational Flow Diagram, Read Alphanumeric Operation, Card Reader and Punch Buffer

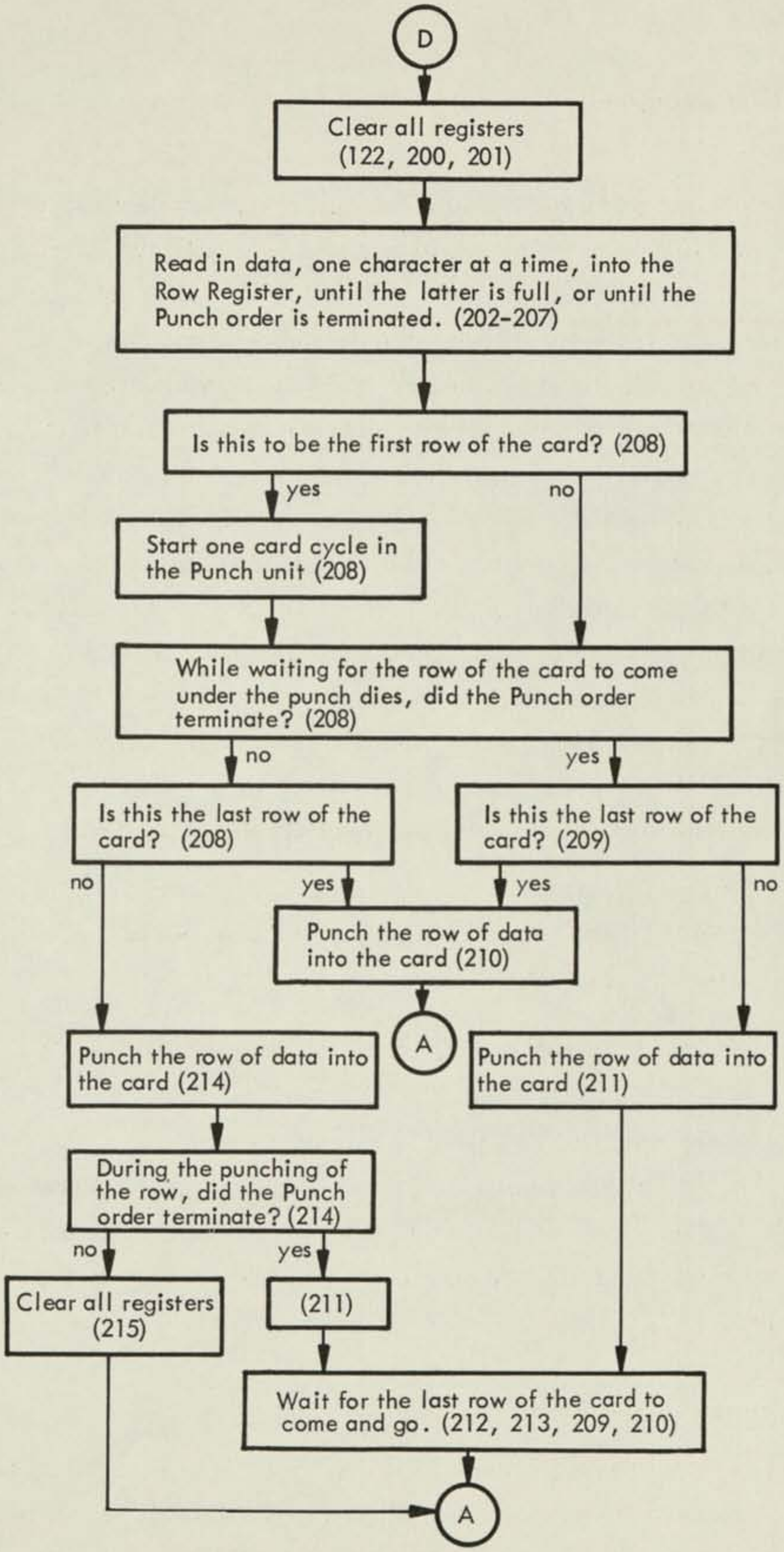

D
Clear all registers (122, 200, 201)
Read in data, one character at a time, into the Row Register, until the latter is full, or until the Punch order is terminated. (202-207)
Is this to be the first row of the card? (208)
yes
no
Start one card cycle in the Punch unit (208)
While waiting for the row of the card to come under the punch dies, did the Punch order terminate? (208)
no
yes
Is this the last row of the card? (208)
Is this the last row of the card? (209)
no
yes
yes
no
Punch the row of data into the card (210)
A
Punch the row of data into the card (214)
Punch the row of data into the card (211)
During the punching of the row, did the Punch order terminate? (214)
no
yes
Clear all registers (215)
(211)
Wait for the last row of the card to come and go. (212, 213, 209, 210)
A

Figure 4-4. Operational Flow Diagram, Punching Operation, Card Reader and Punch Buffer

lines. The card reading sequence is such that the information will arrive in the order indicated in Table 4-1.

Initially the auxiliary buffer and the main buffer wait for a read or punch command from the receiver. The main buffer then initiates one card feed cycle. During reading the 533 machine sends out read synchronizing signals (RROW) which occur during the time the read brushes make through the holes in the card. The read brush outputs to the main buffer are gated into an 80-bit register and stored until the RROW signal goes down. The row is then transmitted (six bits at a time) to the receiver. In the auxiliary buffer a countdown of 3.84 milliseconds is initiated as soon as the RROW signal is detected. At the end of this time, the read brush outputs from the first read station are gated, six at a time, through a selective switch to the receiver. Two microsecond strobe signals are sent during each gating period. Since the read synch signal is approximately 9 milliseconds, the 14 characters associated with the row will be sent before the RROW signal goes down provided the receiver operates fast enough. (No conflict should ensue since the OLCU can accept one character every ten microseconds). The auxiliary buffer then waits for the end of the RROW signal before resetting.

When operating with the OLCU, the auxiliary buffer will send a signal which indicates when the last row of the last card has been read. This will be used by the OLCU to terminate the operation.

If the synchronizing signal goes down before the complete row has been transmitted, the auxiliary buffer will reset and wait for the next synch signal.

The initial setup of the card reader must be such that the first data card will be read twice. It is thus necessary to insert a single blank card at the beginning of the stack. Then when the start button on the 533 is depressed, the blank card and the first data card will be positioned at the second and first reading stations, respectively.

4.4.2 Punch Operation (Off-Line)

The auxiliary buffer will read from the punch read station and transmit it to the receiver. Data is first transmitted from the OLCU to the main card buffer where it is stored until the 533 is ready to punch. When the punch synchronizing

TABLE 4-1. AUXILIARY CARD BUFFER

Row No.	Card No.	From Auxiliary Buffer	From Main Buffer
(1) 2 Start-of-block marks			X
(2) 12	n	X	
(3) 12	n-1		X
(4) 11	n	X	
(5) 11	n-1		X
(6) 10	n	X	
(7) 10	n-1		X
(8) 1	n	X	
(9) 1	n-1		X
(10) 2	n	X	
(11) 2	n-1		X
.			
.			
.			
(24) 9	n	X	
(25) 9	n-1		X
(26) 2 End-of-block marks			X

signal (PROW) is received, the 80 bits are gated out to the punch magnets. At the same time, the corresponding row of the previous card will be read at the reading station. The PROW signal is used in the same manner as the RROW signals in the read operation. There are no multiplexing problems in this case since the two buffers operate over different sets of input and output lines.

When the auxiliary buffer detects the last row, it will transmit an end-of-block mark (00 110 100) after the 14th character is sent. The OLCU will use this to keep track of the cards.

4.4.3 On-Line Operation

The auxiliary buffer may be used for on-line reading only. It is not presently possible to obtain a read-after-punch check using this device. Operation is identical to the off-line read cycle. Information will be received in the processor in the same interlaced pattern indicated in Table 4-1.

The operator will have the additional facility of electing whether or not to use the checking feature. This is provided by a switch which effectively prevents the auxiliary buffer from operating when it is turned on.

SECTION V

LOGIC AND MECHANIZATION DESIGNATIONS

The following alphabetical list includes all of the logic and mechanization designations used in the MOBIDIC 7A central processor and converter. All converter symbols have the superscript i to designate the specific converter number. Symbols which are too difficult to define in words have been defined by their Boolean expressions. The asterisk is used to specify pulse control lines. The subscript n is used to designate the stage number within a register or counter.

A_{sn}	Signdigit of the A - Register.
A_n	A - Register digit (n = 1 to 38).
$A_{sn} + B_{sn}$	Boolean Expression $(A_{sn}B'_{sn} + A'_{sn}B_{sn})$.
a	Address bit 13 of In-Out Memory Selection Decoder.
ADB	Order type decoder output "add Beta".
ADC^i_n	Stage n of the 15-stage address counter in converter.
ADD	Order type decoder output "add".
ADM	Order type decoder output "Add Magnitude".

AF^i A three stage counter used in reading magnetic tape; when a synchronizing signal appears in TAR, the AF counter measures an interval of time after which TAR is cleared.

AFF^β Addressable flip-flop specified by β of a SEN, SNS, OTSNR instruction.

$A1D_x^i*$ "Add One to Dx-Counter" pulse control line.

A1N Add One to the "N" counter

AOS Order type decoder output "Add or Subtract".

AOS* Add or Subtract pulse line of the Arithmetic Unit.

AOSAR* Add or Subtract pulse line of the Address Register.

AR_n Address Register bit (n = 1 to 15)

NOTE: When bits 13 through 15 in α all contain ones, an addressable register is specified whose address is contained in bits 1 through 5. When bits 13 through 15 are not all ones, the specific memory is specified by these bits; and bits 1 through 12 specify the location in the memory.

Address Register Decoder Outputs

Symbol	Bits 15 to 13	Bits 12 to 6	Bits 5 to 1	Address register decoder output specifying:
$[AR_{1-5}(A)']$	111	xxxxxx	01000	A - Register
$[AR_{1-5}(B)']$	111	"	01010	B - Register
$[AR_{1-5}(cis^1)']$	111	"	11000	CNV 1 Instruction
$[AR_{1-5}(cis^2)]'$	111	"	11001	CNV 2 Instruction
$[AR_{1-5}(cis^3)]'$	111	"	11010	CNV 3 Instruction
$[AR_{1-5}(cis^4)]'$	111	"	11011	CNV 4 Instruction
$[AR_{1-5}(I^1)]'$	111	"	00001	Index Register No. 1

Symbol	Bits 15 to 13	Bits 12 to 6	Bits 5 to 1	Address register decoder output specifying:
$\left[A_{1-5}(I^{2})\right]'$	111	xxxxxx	00010	Index Register # 2
$\left[AR_{1-5}(I^{3})\right]'$	111	"	00011	Index Register # 3
$\left[AR_{1-5}(I^{4})\right]'$	111	"	00100	Index Register # 4
$\left[AR_{13-15}(MO^{1})\right]'$	000	location in memory		Memory In-Out Reg. #1
$\left[AR_{13-15}(MO^{2})\right]'$	001	location in memory		Memory In-Out Reg. #2
$\left[AR_{13-15}(MO^{3})\right]'$	010	location in memory		Memory In-Out Reg. #3
$\left[AR_{13-15}(MO^{4})\right]'$	011	location in memory		Memory In-Out Reg. #4
$\left[AR_{13-15}(MO^{5})\right]'$	100	location in memory		Memory In-Out Reg. #5
$\left[AR_{13-15}(MO^{6})\right]'$	101	location in memory		Memory In-Out Reg. #6
$\left[AR_{13-15}(MO^{7})\right]'$	110	location in memory		Memory In-Out Reg. #7
$\left[AR_{1-5}(PC)\right]'$	111		01011	Program Counter
$\left[AR_{1-5}(Q)\right]'$	111		01001	Q - Register
$\left[AR_{1-5}(RAR)\right]'$	111		10001	Real-time Address Register 1
$\left[AR_{1-5}(ROR)\right]'$	111		10010	Real-time Output Register
$\left[AR_{1-5}(WSR)\right]'$	111		10000	Word Switch Register
$\left[AR_{1-5}(010)\right]'$	see $\left[AR_{1-5}(A)\right]'$		(interconnection which has same logical significance)	
$\left[AR_{1-5}(012)\right]'$	see $\left[AR_{1-5}(B)\right]'$		(interconnection which has same logical significance)	
$\left[AR_{13-15}(0)\right]'$	000			(see $\left[AR_{13-15}(MO^{n})\right]$ for logical significance)

Symbol	Bits 15 to 13	Bits 12 to 6	Bits 5 to 1	Address register decoder output specifying:
$\left[AR_{13-15}(1)\right]'$	001			
$\left[AR_{13-15}(2)\right]'$	010			
$\left[AR_{13-15}(3)\right]'$	011			
$\left[AR_{13-15}(4)\right]'$	100			
$\left[AR_{13-15}(5)\right]'$	101			
$\left[AR_{13-15}(6)\right]'$	110			
$\left[AR_{13-15}(7)\right]'$	111			

Symbol	Bits 3 to 1
$AR_{1-3}(0)$	000
$AR_{1-3}(1)$	001
$AR_{1-3}(2)$	010
$AR_{1-3}(3)$	011
$AR_{1-3}(4)$	100

ARC_n

ARC_n Address Register carry-chain output of stage n.

AR_nD Address Register emitter follower output.

$AR\rho_n$ Address Register carry-chain interconnection.

AS Add-Subtract control flip-flop.

ASB Order type decoder output "Add or Subtract Beta".

ASR_n	Address Switch Register on console. (n = 1 to 15)
$AS \oplus OF$	Boolean Expression (AS)(OF)' + (AS)'(OF)
AU	Arithmetic Unit
A1ADC	Add One to the Address Counter
A1AF	Add One to the AF Counter
$A1D_X$*	Pulse line that adds One to the D_x Counter
A1PC	"Add One to Program Counter" pulse control line.
A1RAR	"Add One to Real-time Address Register" pulse control line.
A1SHC	Add One to the Shift Counter for Real-Time Output Register
AIT*	"Add One to T-Counter" pulse control line.
B_{sn}	Sign digit of B-Register
B_n	B - Register digit (N = 1 to 38)
b	Address digit 14 of In-Out Memory Selection Decoder
BB^i	The busy bit flip-flop of the ith converter except that when i = 9 the real time unit is referred to.
$\sum BB^i$	The Boolean expression $(BB^1 + BB^2 + \cdots\cdots + BB^n)$
BCB	Eight bit fieldata code for the Beginning of Control blockmark
BCD	Decoder which is high when the Beginning of a Control blockmark is decoded from magnetic tape.
BD^i	Converter flip-flop: $BD^i = 1$ indicates that the device addressed by DAR^i may be used for both input and output - magnetic tape units excluded.

BFR_n^i	37-bit Buffer Register in CNV^i, stage n
BI	Bus Indicator control level which enables the reading of the contents of various registers onto the main transfer bus.
BLE^i	Level which is high whenever CIR^i contains the code for an end of block mark (10001001).
BLEF	Block end former
BLS^i	Level which is high whenever CIR^i contains the code for start of block mark (10100100).
BLSF	Block start former
BMT^i	$BMT^i = 1$ when BB^i and MT^i are high simultaneously.
$\sum BMT^i$	Boolean expression $(BMT^1 + BMT^2 + \cdots\cdots + BMT^n)$.
BOT	Beginning of tape sensing switch in magnetic tape unit.
BSP^i	ISR^i decoder output "Backspace".
BSR_n^i (n = 1 to 8)	An eight-stage character buffer register.
BUS_n	An emitter follower which is high when bus n is low.
BUS_{28}	A signal from tape unit to converter which warns of approach of end of tape.
BWA	Bus Indicator Switch - "Display A Register".
BWB	Bus Indicator Switch - "Display B Register".
$BWCIS^n$	Bus Indicator Switch - "Display CIS Register."
BWI^n	Bus Indicator Switch - "Display I^n Register".

$BWMO^n$	Bus Indicator Switch - "Display MO^n Register".
BWPCS	Bus Indicator Switch - "Display PCS Register".
BWQ	Bus Indicator Switch - "Display Q-Register".
BWRAR	Bus Indicator Switch - "Display RAR Register".
BXR^i_n(n = 1 to 8)	An eight-stage character buffer register
C_n	Inverter matrix output of stage n in A - Register carry chain
C1W	See C_nW
CAD^i	A timing level generated by AF
CAM	Order type decoder output "Clear Add Magnitude"
CCA^i	Push button in converter which clears the converter alarms
$CC1^i$	$\overbrace{(MT)(GS)(CH')}^{CC12}\ \overbrace{(PBM1')(PBM2')}^{CC14}$
$CC2^i$	$(CC12)\ \overbrace{(PBM1)(PBM2')}^{CC16}$
$CC3^i$	$(CC12)\ \overbrace{(PBM1)(PBM2)}^{CC15}$
$CC4^i$	$(CC12)\ \overbrace{(PBM1')(PBM2)}^{CC13}$
$CC5^i$	(MT)(RCF) (CC15)(CAD)
$CC6^i$	(MT)(RCF) (CC13)
$CC7^i$	$\overbrace{(MT)(CH)(GS)}^{CC17}$ (ORF1)(ORF2)
$CC8^i$	(CC17) (ORF1)(ORF2)
$CC9^i$	CC17 (ORF1')(ORF2)

$CC10^i$	(RH')(Dx = C)(WBC = 0)'
$CC11^i$	(RH')(Dx = C)(WBC = 0)
$CC12^i$	(MT)(GS)(CH)'
$CC13^i$	(PBM1')(PBM2)
$CC14^i$	(PBM1')(PBM2)'
$CC15^i$	(PBM1)(PBM2')
$CC16^i$	(PBM1)(PBM2)'
$CC17^i$	(MT)(GS)(CH)
$CC18^i$	(RR + SS + WWA)
$CC19^i$	(ID)(RCF)
$CCAW^i$	Clear all converter alarm push buttons
CEF	A control bus line between Central Processor and Converter which sets the End of File flip-flop in the Central Processor.
CETA	A control bus line between Central Processor and Converter which sets ETA flip-flop.
CF^i	A control flip-flop in converter i; used in conjunction with block marks for in-out orders involving magnetic tape.
CFC-C	A control bus line between Central Processor and Converter which sets the FCC^i flip-flop
CFF	Commutating flip-flop in the timer
CFK	A control flip-flop used to sync control characters in the Converter
CGC^i	Control signal which inactivates the GS flip-flop
CH^i	A control flip-flop in converter i.
CIA^i	Level which clears In-out Alarm flip-flop in CNV^i
CIPW	Console switch used to reset TPE flip-flop

CIR$_n^i$	8-bit character register in CNV^i.
CIS	Converter instruction word register
CISW	"Clear ISN" push button
CLA	Order type decoder output "Clear and Add"
CLAW	Clear All Error Flip-Flops switch
CLCW^i	Push button which generates CLRCIS^i (see below)
CLOCK*	Central timing clock pulse
CLPER	Clear Parity Error Flip-Flop
CLR(CLR*)	Level (pulse) which clears the timing flip-flop when computer is first energized or reset
CLRA*	Clear A - Register pulse control line.
CLRAR*	Clear Address Register pulse control line.
CLRB*	Clear B - Register pulse control line.
CLRBB^i	Clear the BB^{ith} flip-flop
CLRBFR^i*	Clear BFR register pulse control line.
CLRBSR^i*	Clear BSR register pulse control line.
CLRBXR^i*	Clear BXR register pulse control line.
CLRCIR*	Clear Converter Instruction Register pulse control line.
CLRCIS^i	Control level which clears all flip-flops in converter i.
CLRCIS^i*	Clear pulse line for converter i.
CLRD$_x^i$*	Clear Delay-Counter pulse control line.
CLRDX^i	Clear delay-counter control flip-flop (DX).
CLRGA*	Clear all GA flip-flops pulse control line.
CLRI^n*	Clear Index Register No. n pulse control line. (n = 1 to 4)
CLRIR*	Clear Instruction register pulse control line.

CLRMA^{n}* Clear Memory Address No. n pulse control line. (n = 1 to 7).

CLRMO^{n}* Clear In-out register of Memory No. n pulse control line. (n = 1 to 7).

CLRPC* Clear Program Counter pulse control line.

CLRPCS* Clear Program Counter Store pulse control line.

CLRQ* Clear Q - Register pulse control line.

CLRRAR* Clear Real-time Address Register pulse control.

CLRRIR Clear Real-time Input Register

CLRRH^{i} Clear the run-halt flip-flop.

CLRROR Clear Real-time Output Register

CLRT* Clear T-counter pulse control line.

CLRTAR^{i}* Clear TAR register pulse control line.

CLRXG* Clear X and G registers pulse control line.

CLS Order type decoder output "Clear and Subtract".

CLW Push button on Console that resets all Flip-Flops and Registers to their initial state

CLWAR^{i} Control line that clears the Write amplifier Flip-Flops of the Ampex FR-300 Tape Transport

CLXCIS A level which is part of the CLRCIS logic.

CM Clear Memory control flip-flop.

CMBD_{n} Clear Memory Bus Driver which puts memory designation on bus 13 to 15. (n = 13 to 15)

CMPX(L707) Clears all MPX Flip-Flops

CMW^{n} Clear Memory n push button (one for each memory)

$\sum \text{CMW}^{n}$ Boolean expression ($\text{CMW}^{1} + \text{CMW}^{2} + \cdots + \text{CMW}^{n}$) a level which is high whenever a CMW button is pushed.

CNPW	"Clear NHP" - push button
CNV^i	Control flip-flop which is in the <u>one</u> state when converter i is busy.
$(\sum CNV^i)'$	Boolean expression $(CNV^1 + CNV^2 + \cdots + CNV^n)'$
$(\pi CNV^i)'$	Boolean expression $(CNV^1)(CNV^2)(\cdots)(CNV^n)'$
C_nW	Console switch that clears the Sense Flip-Flop n
COMPA*	Complement A - Register pulse control line of AU.
COMPB*	Complement B - Register pulse control line of AU.
COW	Continue -- push button.
$CRMOD^n$	Sub-function which is used in MO logic.
CSL^i	Converter selector level; "i" determines which converter will be used to execute the next in-out order."
CSM	Order-type decoder output "Clear Subtract Magnitude".
CTMW	"Clear TRA" push button switch
CTP^i	A control level originating in CNV^i; when $CTP^i = 1$, the TP flip-flop will normally be set to one.
CVB^i	A level that is high when CNV^i is in use.
$(CVB^o)'$	Register driver output generated under conditions: (PR3')(TP) + 10'
CYL	Order type decoder output "Cycle Long"
CYS	Order type decoder output "Cycle Short"
CNW	Clear sense flip-flop n. (n = 1 - 16)
D_n	Decoder register bit. (n = 31 to 36)
d	Address Bit 1 of In-Out Memory Selector Decoder
DAR_n^i(N = 16 to 21)	6 bit Device Address Register in CNV^i.

D_{33} D_{32} D_{31}	Decoder register decoded outputs.
DEE	Decoder output line of the P_x counter
DK-n	Order type decoder outputs
DLT	Decoder which detects Delete code on paper tape
DPIW	Disable program interrupt switch
DSU	Device Switching Unit
DVA^i	Device Alarm flip-flop in CNV^i.
DVB	Control level which is high when the x register holds the address of an In-Out device which is in use.
DVD	Order type decoder output "Divide"
DVH	Device halt flip-flop
DVL	Order type decoder output "Divide long"
Dx^{i}n	Delay counter for timing of magnetic tape.
(Dx^i = n)	Decoded time intervals measured by the Dx timer. (n = A, AAB, ABD, B, C, CD, DEE, 24µs, 34µs.)
DxE^i	Level which is high when Dx overflows.
DX^i	Control flip-flop for Dx counter. When Dx = i counter is in operation.
e	Address Bit 2 of In-Out Memory Selection Decoder
ECB	Eight bit fieldata code for the End of Control blockmark
ECD	Decoder which detects the End Blockmark of a control block on magnetic tape
EFD	Decoder which detects receipt of End of File mark from magnetic tape
EFI	End File Interrupt flip-flop
EFK	End of File flip-flop in the Converter
EH	Control line for Emergency Halt

EHW	Emergency Halt Switch.
END^i	Control level originating in CNV^i which indicates end of in-out order.
EOF	A level representing an eight-bit Fieldata configuration meaning End of File.
EOF^i	End of File flip-flops (i = 1, 2, 3, or 4)
ETA^i	End Tape Area flip-flops (i = 1, 2, 3, or 4)
ETI	End Tape Interrupt flip-flop
ETK	Flip-flop set by Bus 28.
EOT	End of tape sensing switch in magnetic tape unit.
ERR	Error signal in the converter.
F	A control level which is high whenever IR holds the code for one of the following instructions MOV, LDX, TRL, TRX.
f	Address Bit 3 of In-Out Memory Selection Decoder.
FCI	Fieldata Control Character Interrupt flip-flop.
FCC^i	Read Fieldata Control Character flip-flop (from CNV^i)
FX	Intermediate condition (TF8 + H). FX has no special logical significance in itself.
G$_n$	G - Register bit n of the instruction word (γ) (n = 28 to 30).
g(0)	In-Out Memory Selection - Intermediate Decoder A - Register.
g(1)	In-Out Memory Selection - Intermediate Decoder Q - Register.
g(2)	In-Out Memory Selection - Intermediate Decoder B - Register.
g(3)	In-Out Memory Selection - Intermediate Decoder PC - Register.
g(4)	In-Out Memory Selection - Intermediate Decoder PCS - Register.
GA^i	A flip-flop which is high whenever Converter i has access to In-Out Memory Cycle.

$\sum GA^i$ — The Boolean expression $(GA^1 + GA^2 + GA^3 + \cdots + GA^n)$

$\left[GA_{28-30}(I^n)\right]$ — Emitter follower outputs of the function next defined. $(n = 1 \text{ to } 4)$

$\left[GA_{28-30}(n)\right]$ — G-Address decoder output generated when bits 28 to 30 contain binary (n). (where $n = 1$ to 4)

GS^i — Control flip-flop which indicates whether or not the converter may be released to the converter selector circuit.

H — Halt flip-flop

HAW — Halt push-button

HLT — Order-type decoder output "Halt".

HLTD — Pulse line which sets Halt instruction into D - Register.

HT — A control level which is high whenever IR holds the code for the HLT instruction.

HTW — Console Halt Switch

$(I^3 = 0)$ — A level which is high when Index Register No. 3 contains all zeroes.

$(I^\gamma + 1) = 0$ — A level which is high whenever the next higher order Index Register than the one specified by γ contains all zeroes.

[IAD(A)] — In-Out Address Decoder output specifying A - Register.

[IAD(B)] — In-Out Address Decoder output specifying B - Register.

[IAD(CIS)] — In-Out Address Decoder output specifying CIS Register.

$\left[\text{IAD}(\text{MO}^n)\right]$ — In-Out Address Decoder output specifying Memory In-Out Register n.

[IAD(PC)] — In-Out Address Decoder output specifying Program Counter.

[IAD(PCS)] — In-Out Address Decoder output specifying Program Counter store.

[IAD(Q)] — In-Out Address Decoder output specifying Q - Register.

[IAD(7)] In-Out Address Decoder intermediate output generated when Address bits 13 to 15 contain binary (7).

IC_8 Emitter follower output stage 8 in carry chain of Index Register.

ID^i Flip-flop that is set when an input type device is selected by converter and is not magnetic tape unit.

IGT Flip-flop that is used to gather information into the TAR Register.

IMO^i Improper order alarm flip-flop in CNV^1.

INT_i Sub groups which are combined to form $\sum$INT (i = 1 to 8)

$\sum$INT BUS bus line from main machine to converter which is high whenever the X - Register contains address of an input-type device.

IO A control level which is high whenever IR holds the code for an in-out instruction.

IOA^i In-out Alarm flip-flop in CNV^i.

IORDY Device ready line.

IOT_i Sub groups which are combined to form $\sum$IOT (i = 1 to 8).

$\sum$IOT BUS Bus line from main machine to converter which is high whenever the X - Register contains address of an input-output type device.

IPE^i In-out Parity Error flip-flop in CNV^i.

IR Instruction Register

IR_n Instruction register bits (n = 31 to 36).

IS^i Interpret Sign flip-flop in CNV^i.

ISBUS Interpret Sign Bus

ISE^i Sign Error flip-flop in CNV^i.

ISG A signal to the converter indicating the Central Processor is in TF3 or TF4 timing function in the basic cycle.

ISN Interpret sign flip-flop.

$ISR^i_n (n = 31 \text{ to } 36)$ 6-bit Instruction Storage Register in CNV^i.

$\left[ISR^i_{31-34}(n)\right]$ Instruction Storage Register decoder output which is high whenever ISR contains an octal n.

KA^i (CC17)(CC14)(RCF)(DX)'

KH^i through KA^i are "or" gated to clear TAR.

KB^i - (WCF)(BXR = 0)'(TAR = 0) + (OD)(CC14)(BXR = 0)'(TAR = 0) The logical conditions which transfer BSR to TAR and clear BXR.

KC^i - (WCF)(BSR = 0)'(BXR) + (OD)(BSR = 0)'(BXR = 0) The logical conditions which transfer BSR to BXR and clear BSR.

KD^i - (CC3)(RCF)(CAD)(TBLS + TBLE)' + (CC4)(RCF)(CAD)(TBLS + TBLE) The logical conditions which transfer TAR to BXR.

KE^i - (CC19)(CC16)(BXR = 0)(TAR_7)(TAR_6)(STC)(IS') The logical conditions which transfer TAR to BXR and TBLE to TAR.

KH^i (CC19)(CC16)(TAR_6')(TAR_7')(TAR = 0)' + (TAR_6)(TAR_7) (STC')

KJ^i (WCF)(Dx = 24μs)(CC4) + (TBLE')(CC3)

KK^i (WAN)(CC2)(Dx = 24μs) + (CC3)(Dx = B)

KL^i (OD)(CC13)(TAR = 0)'

KM^i (RCF)(CC12)(Dx = 34μs)(PBM2')(CC18)

KEW Inhibit Error Halt Switch

KOR Ready level from receiving element

KS Strobe signal from sending module

KSS Strobe quantizing flip-flop

KSX	Transfer-in and shift control flip-flop
LC^{i}	A control level which is high whenever CIR^{i} holds a legitimate character for the ROK order.
LDX	Order-type decoder output "Load Index".
LGA	Order-type decoder output "Logical Add".
LGM	Order-type decoder output "Logical Multiply".
LGM*	Logical Multiple pulse control line of AU.
LGN	Order-type decoder output "Logical Negation".
LL20^{i}	The Boolean Expression (ISR$_{31}$')(ISR$_{32}$)'
LL21^{i}	(ISR$_{31}$)(ISR$_{32}$')
LL22^{i}	(ISR$_{31}$')(IRS$_{32}$)
LL23^{i}	(ISR$_{31}$)(ISR$_{32}$)
LL24^{i}	(RAN)(RRV)
LL25^{i}	(WAN)(WWA)
LL26^{i}	(BSP)(SKP)
LMAR	A control level which is high when the Address Register contains the address of the highest numbered memory contained in the computer system.
LMIAD	A control level which is high during in-out memory access when the selected RAR or ADC contains the address of the highest numbered memory contained in the computer system.
LMPC	A control level which is high when the Program Counter contains the address of the highest numbered memory contained in the computer system.
LOD	Order-type decoder output "Load".
MA^{n}	nth Memory Address Register
MA$^{i}_{n}$	Memory Address Register bit n of memory No. i. (n = 1 to 8)

MI	Manual Instruction control level from console.
MIW	Manual Instruction push-button.
MLR	Order-type decoder output "Multiply and Round".
MLY	Order-type decoder output "Multiply".
MM	Mass memory control line.
MO_n^i	In-Out Register bit n of memory No. i (n = 1 to 36).
MO_{38}^i	Parity bit of In-out register of memory 1.
MO_{sn}	Sign bit of In-out register of memory 1.
MOV	Order-type decoder output "Move".
MOR^n	Output generated for condition ($MO^n + REG^n$).
MPE^n	Memory Parity Error flip-flop of Memory n.
MPX^n	Memory Parity control flip-flop.
MR	Manual Read In-Out control flip-flop.
MRA	A control level which is high whenever D holds the code for one of the following instructions: CLA, CAM, CLS, LGN.
MRB	A control level which is high whenever D holds the code for one of the following instructions: ADD, ADM, SUB, SBM, MLY, MLR, DVD, DVL, LOD, MOV, TRC, LGM, LGA.
MSK	Order-type decoder output. "Replace through mask".
MT^i	Control level which is high whenever the address stored in DAR^i specifies a Magnetic Tape Unit.
$\sum MTT_n$	Sub-groups which are combined to form $\sum$ MTT (n = 1 to 8)
MTU	Magnetic Tape Unit

$\sum$MTT BUS Bus line from main machine to converter which is high whenever the X - Register contains address of a magnetic tape device.

N_n^i A four-stage counter which counts the number of characters shifted into or out of BFR^i.

NHC Non-Halt on Converter Error.

NM A control level which is high whenever IR holds the code for NRM instruction.

NRM Order-type decoder output "Normalize".

NH^i Ignore halt on In-out alarm flip-flop in CNV^i.

NXI Non-Existent Instruction flip-flop.

NXIL Level which is high whenever the Decoder holds the code for a non-existent instruction.

NXM Non-existent memory flip-flop.

NXML Level which is high whenever the Program Counter holds the code for a non-existent memory.

OA Overflow Alarm flip-flop.

OD^i Flip-flop set when an output-type device is selected.

OF Overflow flip-flop.

OK_n^i (n = 1 to 3) The Boolean expression $(GA^1)(TCn)$ (ϕ). [n = 1 to 3]

ONC One-cycle operation flip-flop

ONE Fieldata Code for One

$ORF1^i$ Control flip-flop which was just completed was a read order (including SKP, AND, BSP).

$ORF2^i$ Control flip-flop which is in the one state if the tape was moving forward for the in-out order just completed.

OTT_i Sub groups which are combined to form OTT (i = 1 to 8)

$\sum$OTT BUS	Bus line from main machine to converter which is high whenever the X-Register contains address of an output-type device.
P^1_{1-37}	Level which is high whenever bits MO_1 to MO_{37} contain an even number of ones.
p-level	A level generated at the CFF flip-flop at 500 k. c. rate in the timer.
PBM^i_n	Control flip-flops used to control the detection and writing of block marks.
PC_n	Program Counter bit (n = 13 to 15)
$\left[PC_{13-15}(MO^n)\right]$	Emitter follower outputs for the next defined functions.
PCS	Program counter store register.
PE	Parity error level.
$\left[PC_{13-15}(n)\right]$	Program Counter address decoder output generated when bits 13 to 15 contain binary (n). (n = 1 to 7)
PFM^i	Parity Former output associated with CIR^i; $PFM^i = 1$ whenever $CIR^i_{1 \text{ to } 7}$ contains an even number of ones.
PHI	A group of conditions which set the ϕ flip-flop.
PI	Program Interrupt Flip-Flop.
PR_n	Timing Function for Program Read-In (n = 0 + 3)
PRF_n	Control for Program Read-In flip-flops (n = 1, 2)
PRW	Program Read-In push-button.
PTI	Control level for Photo-Electric Tape Reader
$\sum$PX	A signal which is high when Program Interrupt is desired.
Q_n	Multiplier Quotient Register bit (n = 1 to 38).

Q_{sn}	Q - Register sign bit.
QA^i	Boolean expression (RAN + SKP + WWA)(TBLS) + (RRV + BSP)(TBLE) expression is used for writing and reading conscience and has no logical significance.
QB^i	Boolean expression (MT)(GS)(CH)'(PBM1)'(PBM2)' (WAN)(TAR) (See note for QA^i).
QVn	Intermediate logic functions.
QW_n	Intermediate logic functions, commonly used, but have no logical significance in themselves.
RA	"Replace Address" level control line of AU.
RAN^i	ISR^i decoder output "Read Alphanumeric".
RAR	Real time address register.
RAR_n^i	Real time Address bit (n = 1 to 15) of real time address register No. i. (i = 1 to 4)
$RARC_{12}$	Real-time Address Register carry-chain emitter follower in stage 12.
RCF^i	Read control flip-flop.
RCI	Real Time Control Character Interrupt flip-flop.
RE^n	Level input to Real Pulser of Memory No. n. (n = 1 to 7)
REG^n	Any addressable register.
RDY	Converter ready signal control line.
RDYC	Converter ready line (Ready line from converter).
RDYD	Ready line from device.
RH^i	Control flip-flop which turns tape on or off.
RI	Manual Read in control flip-flop.

RINC	Real Time Control Character (Input) flip-flop.
RISN	Real Time Input Interpret Sign flip-flop.
RIR	Real Time Input Register.
RIW	Manual Read-in push button.
RM	"Replace through Mask" level control line of AU.
ROBB	Real Time Output Register busy bit flip-flop.
ROBP	Real Time Output Register Parity Signal to Kineplex.
ROK^i	ISR^i decoder output "Read Octal".
ROPI	Real Time output program interrupt flip-flop.
ROR	Real Time output register.
RORC	Real Time Output seventh bit control level.
ROS	Strobe flip-flop.
ROSN	Real Time Output Interpret Sign flip-flop.
ROTC	Real Time Control Character (Output) flip-flop.
ROW	Manual Read-Out push button.
ROX	Ready signal quantizing flip-flop.
ROZ	Strobe shift control flip-flop.
RP	"Repeat" control flip-flop.
RPA	Order-type decoder output "Replace Address".
RPE	Real Time parity error flip-flop.
RPL	Order-type decoder output "Replace".
RPT	Order-type decoder output "Repeat".
RR^i	A control level which is high whenever ISR^i contains the code for RAN or RRV.
RRV^i	ISR^i decoder output "Read Reverse".
RSD$_x$	Reset D$_x$ - counter control line in the converter.
RV^i	Flip-flop which is high when the tape unit runs in the reverse direction not rewinding.
RW^i	Read-write control; RW^i = 0 for read; RW^i is a level in CNV^i and a flip-flop in the Real-time unit i.

$\sum RW$	The Boolean expression $(RW^1 + RW^2 + \cdots + RW^n)$.
RWB^i	Flip-flop that is in the one state when magnetic tape unit is not rewinding.
RWD^i	ISR^i decoder output "Rewind".
SIT*	Subtract one from T counter.
SIWBC	Subtract one from WBC counter.
$S1I^\gamma$	Subtract one from Index register.
S1W-S16W	"Set SFFβ" push button (β = 1 to 16)
SA^1_{1-38}	Sense Amplifiers bits 1 to 38 for Memory No. 1.
SBB	Order-type decoder output "Subtract Beta".
SBCB	Set BCB in the TAR register.
SBF	Strobe flip-flop.
SBLE	Set BLE in the TAR register.
SBLS	Set BLS in the TAR register.
SBM	Order-type decoder output "Subtract Magnitude".
SBX	Strobe control flip-flop.
SC	A control level which is high whenever IR holds the code for one of the following instructions: SHL, SLL, SHR, SRL, CYS, CYL.
SCI^i	A control level which is high whenever an out-of-sync block mark is detected.
SCL	Control line which indicates that the device conforms to SCL 1986.
SECB	Set ECB in the TAR register.
SEN	Order-type decoder output "Sense".
SEOF	Set EOF in the TAR register.
SFF^n	Sense flip-flop n. (n = 1 to 16).

SKP^{i}	ISR^{i} decoder output "Skip".
SG	Timing flip-flop for halt circuit.
SGA*	Set selected GA flip-flop pulse control line.
[(SG)(PRW')]	A control level which is high when an operation is initiated and it is not Program Read-in.
SH	A control level which is high whenever D holds the code for one of the following instructions: SHL, SLL, SHR, SRL, CYS, CYL.
SHC	Real Time output shift counter.
SHL	Order-type decoder output "Shift left".
SHR	Order-type decoder output "Shift right".
SIM^{i}	Control signal which is part of CGC^{i}.
SIP	Single-pulse rotary switch.
SISW	Set ISN Flip-flop push button.
SKP	Skip instruction.
SLA*	"Shift left A - Register" pulse control line of AU.
SLB^{i}*	Pulse line which shifts the contents of BFR^{i} left three places.
SLC	A control level which is high whenever D holds the code for one of the following instructions: SHL, SLL, CYS, CYL.
SLL	Decoder output "Shift left long".
SLQ*	"Shift Left Q - Register" pulse control line of AU.
SLROR	"Shift left ROR" control line.
SNFCIR	A level which indicates that a sign character is to be formed in CIR.
SNL^{β}	A sense level whose address is β.
SNPW	Set NHP flip-flop push button.

SNR	Order-type decoder output "Sense and Reset".
SNS	Order-type decoder output "Sense and Set".
SP	Single pulse flip-flop.
SPE	Parity error sampling flip-flop.
SPI	Stop Program Interrupt flip-flop.
SPL*	A level which gates the clock source to the timer.
SPL	*Single pulse control level.
SPR*	Pulse line which is generated by the program read-in switch. (Converter 1 only).
SRA	"Shift Right A - Register" pulse control line of AU.
SRL*	Order-type decoder output "Shift Right Long".
SRQ	"Shift Right Q - Register" pulse control line of AU.
SS^i	A control level which is high whenever ISR^i contains the code for BSP or SKP.
STB^i	Strobe flip-flop in the converter.
STBC	Strobe output level from converter.
STP	A level that is high when photoelectric reader decodes the stop code character on paper tape.
STMW	Set TRA flip-flop push button.
STP	Stop signal from paper tape.
STR	Order-type decoder output "Store".
STW	"Start" push button.
SUB	Order-type decoder output "Subtract".
SWC	A level used to clear the seventh bit in the TAR register.
T_n	T - Counter bit (n = 1 to 6) .
te	A pulse which occurs before the normal t-pulse.

t-level	A level generated at the CFF flip-flop at a 500 k. c. rate in the timer.
$(T = n)$	T - Counter contains binary contents (n).
$(T \leq n)$	T - Counter contains binary contents equal to or less than (n).
TAR_n^i	8 stage buffer character register which connects to in-out busses, and the BXR register.
$TBLE^i$	End block mark detected in TAR.
TBLEF	Decoder which detects the End of Blockmark of a normal data block on magnetic tape.
$TBLS^i$	Start block mark detected in TAR.
TBLSD	Decoder which detects the Start Blockmark of a normal data block on magnetic tape.
TCF_n	Flip-flops which generate the TC timing functions for the In-out cycle. (n = 1, 2)
TCX	Timing flip-flop for in-out cycle.
TC - n	In-out Memory Cycle timing function. (n = 1 to 4)
TF - n	Timing functions for basic computer cycle. (n = 1 to 8)
TFF_n	Timing flip-flop n. (n = 1 to 3).
$TH\text{-}_n^i$	Timing functions in the converter.
TOT	Transfer Order Trapped flip-flop.
TP	Stop flip-flop.
TPE	Tape erase flip-flop.
TR	Transfer control line.
TRA	Trapping mode flip-flop.
TRC	Order-type decoder output "compare".
TRE^i	Flip-flop set to one when there is Timing read error.

TRIA*	"Transfer into A - Register" pulse control line.
TRIAR*	"Transfer into the Address Register"
TRIB*	"Transfer into B - Register" pulse control line.
TRIBFR$^{i}_{1-6}$*	Transfer into BFR stages 1 to 6 pulse line.
TRIBFR^{i}*	Transfer into BFR
TRIBSRA^{i}*	Transfer into BSR register from CIR pulse line
TRIBSRB^{i}*	Transfer into BXR register from BXR pulse line.
TRIBXRA^{i}*	Transfer into BXR register from BSR pulse line
TRIBXRB^{i}*	Transfer into BXR register from TAR pulse line
TRICIR^{i}*	Transfer into CIR pulse line.
TRICIS^{i}*	Transfer into converter instruction register pulse line.
TRIFCB^{i}*	Transfer in from converter bus.
TRID*	"Transfer into D - Register" pulse control line.
TRIDAR^{i}*	Transfer into device Address Register pulse line.
TRII^{n}*	"Transfer into Index Register No. n" pulse control line. (n = 1 to 4)
TRIIR*	"Transfer into Instruction Register " pulse control line.
TRIMA^{n}*	"Transfer into Memory Address Register No. n" pulse control line. (n = 1 to 7)
TRIMO^{n}*	"Transfer into In-out Register of Memory No. n" pulse control line. (n = 1 to 7).
TRIMTH*	Transfer in buses 9-16 to magnetic tape control signal.
TRIMTL*	Transfer in buses 1-8 to magnetic tape control signal.
TRIPC*	"Transfer into Program Counter Store" pulse control line.
TRIPCS*	"Transfer into Program Counter Store" pulse control line.

TRIQ* "Transfer into Q - Register" pulse control line.

TRIRAR* "Transfer into Real-time Address Register" pulse line.

TRIRIR* "Transfer into Real-time Register" pulse control line.

TRIROR* "Transfer into Read-time Output Register" pulse control line.

TRIT* "Transfer into T - Counter" pulse control line.

TRITAR^{i}* Transfer into TAR pulse line.

TROTAR^{i} Transfer out of TAR level.

TRIXGH* "Transfer into X-G-Register High" pulse control line (high-meaning bits 16 to 27 into G and bits 28 to 30 into X).

TRIXGL* "Transfer into X-G-Register Low" pulse control line (low-meaning bits 13 to 15 into G and bits 1 to 12 into X).

TRL Order-type decoder output "transfer and load PCS".

TRN Order-type decoder output "transfer on negative".

TROA "Transfer out of A - Register" control level.

TROAR "Transfer out of Address Register" control level.

TROASR "Transfer out of Address Switch Register" control level.

TROB "Transfer out of B - Register" control level.

TROBFR^{i} Transfer out of Buffer Register.

TROCIS A control level which is high when the Converter Instruction Word (ISR, WBC, DAR, and ADC) is transferred onto the main bus.

TROGL "Transfer out of G Register Low" control level (low bits of G-Register are bits 13 to 15).

TROI^{n} "Transfer out of Index Register No. n" control level ($n = 1$ to 4).

TROMO^{i} "Transfer out of Memory In-Out Register No. i control level ($i = 1$ to 7).

TROMT	Transfer out of magnetic tape.
TROMTS	Transfer out of magnetic tape strobe signal.
TROPC	"Transfer out of Program Counter" control level.
TROPCS	"Transfer out of Program Counter Store" control level.
TROQ	"Transfer out of Q - Register" control level.
TRORAR	"Transfer out of Real-time Address Register" control level.
TROREG^n	"Transfer out of Addressable Register".
TRORIR	"Transfer out of Real-time Register" control level.
TROT	"Transfer out of T-Counter" control level.
TROTAR	"Transfer out of TAR Register" control level.
TROWSR	"Transfer out of Word Switch Register" control level.
TROXGH	"Transfer out of X-G-Register High" control level (high bits of X are 16 to 27, high bits of G are 28 to 30).
TROXL	"Transfer out of X - Register Low" control level (low bits of X are 1 to 12).
TRP	Order-type decoder output "Transfer on Positive".
TRS	Order-type decoder output "Transfer to PCS".
TRU	Order-type decoder output "Transfer unconditionally".
TRX	Order-type decoder output "Transfer on Index".
TRZ	Order-type decoder output "Transfer on Zero".
TR1	Transfer function generated for conditions (PR1)(TC2).
TR2	Transfer $(TF5)(MLY)(T = 35) + (TF4)(MLY')(\lambda')$
TR3	Transfer (TF3)(LGA)
TR4	Transfer function generated for conditions: $(\lambda)'$ $[TF8 + (TF1)(NRM)]$

TR5	Transfer function generated for conditions: $(TP')(TF6)(\lambda')(RP')(MI')$
TR6	Transfer $(TF3)(T = 0)(MOV)$
TR7	Transfer $(TF3)(T = 3)(MOV)$
TR8	Transfer $(TF3)(T = 5)(MOV)$
TR9	Transfer $\left[(X_{16-22})AFF^{\beta}\right]\left[(SEN)(AFF^{\beta}) + (SNS)(AFF^{\beta})'\right] + (TF3)(ZT)(TRZ) + (TF3)(TRU) + (X_{16})$
TR10	Transfer $(TC1)(MR) + (T = 2)(TF3)(MOV)$
TR11	Transfer $(TF3)(\lambda)(LOD) + (T = 1)(MOV) + (T = 0)(TRX)(I^{\alpha+1}) = 0)'$
TR12	Transfer $(TF2)(LDX) + (TF3)\left[(RPT) + (T = 2)(TRX)\right]$
TR13	Transfer $(TF3)\left[(TRL) + (A_{sn})(TRN) + (A'_{sn})(TRP)\right]$
TR14	Transfer $(TF2)(RPT) + (TF5)(TRL) + (TF3)(LDX)$
TR15	Transfer $\left[AR_{1-5}(A)\right]'\left[(TF1)(T = 2)(ASB) + (TF2)\right]\left\{(MRA) + (T = 0)(RPL)\right\}$
TR16	Transfer $\left[AR_{1-5}(B)\right]'(MRB)(TF2)$
TR17	Transfer $(TF1)(STR) + (TFZ)(T = 2)(RPL)$
TR18	Transfer $(TF5)(MOV)$
TR19	Transfer $(TF2)(X'_{16})(TRA)(TRU)$
TR20	Transfer $(TF3)(TRS)$
TR21	Transfer function generated for conditions: $(TF2)(TRL)$
TR22	Transfer $(TF2)(TRX) + (TF3)(\lambda')(TRC)$
TR23	Transfer $(TF3)(\lambda')(I^{\alpha+1} = 0)'(TRX)$
TR24	Transfer $(TF1)(T = 0)(ASB) + (TF3)\left[(MLY + (T = 0)(TRC)\right]$
TR25	Transfer $(TF5)(TRC)$

TR26	Transfer $(TF1)(RPL)$
TR27	Transfer $(RPL)[(TF5) + (TF2)(T = 1)]$
TR28	Transfer $(TF3)(T = 1)(TRX) + (TF7)(F')(RP')(\lambda')(IO')(TP')$
TR29	Transfer $(TF8)(\lambda')(TP)[PRO + (PR3)(ONC)]$
TR30	Transfer $(TF3)(T = 4)(MOV)$
TR31	Transfer $(XP)'(TF7)(TP')(\lambda')(IO')(RP)$
TR32	Transfer $(TF3)(\lambda')(IO)(PR3 + TP')$
TR33	Transfer $(TF8)(\lambda')(PR2 + TP')$
TR34	Transfer $(TR15 + TR16)$
TR35	Transfer $[(T = 2)' + TF1' + ASB']\{TF2' + (MRA)'(MRB)'$ $\{[(T = 0)' RPL']\}$
TR36	Transfer $\{[AR_{1-5}(A)]'(TF1)(ASB) + (T = 5)'\}$ $\{(T = 2)' + [AR_{1-5}(A)]'(TF2)(RPL)\}$ $\{[AR_{1-5}(A)]'(STR)(TF1) + (MR)(RI)(TC2)\}$ $\{(TF2)(NRM) + \lambda'\}\{(T = 3)' + [AR_{1-5}(B)]'$ $(MOV)(TF3)\}$
TR37	Transfer $(TRA)(\lambda')(TS)(TF7)$
TR38	Transfer $(TC2)(RI)(MR)$
TR39	Transfer $TC2' + \phi + RW$ or $[(TC2)(\phi)(RW')]'$
TR40	Transfer $(MR)(TCX)$
TR41	Transfer $TF3' + \lambda + LOD'$
TR42	Transfer $(\lambda')(MI)(TF6)$
TR43	Transfer function generated for conditions: $(\phi)(TC3)(RW)'$
TR44	Transfer $\{(T = 5)' + [(TF1)(ASB)]'\}\{(TF3)'$ $+ (LDX')[(T = 2)' + TRX']\}\{TF5' + TRL'\}$
TR45	Transfer $(TF2' + LDX')$

TR46 Transfer (TC1)' + MR'MOV' + (T = 2)'-+ TF3' λ + (NRM' + TF1')(TF8' + NM)

TR47 Transfer (TC1' + ϕ)

TR48 Transfer (λ + MLY + TF4')(T = 35' + MLY' + TF5')

TR49 Transfer (T = 1)(ASB)(TF1)

TR50 Transfer (T = 5)(ASB)(TF1)

TR51 Transfer (λ')(PR3)(TF8)

TR52 Transfer ($\sum$CMW^n)(SG)

TS A control level which is high whenever IR holds the code for one of the following instructions: TRL, TRN, TRP, TRS, TRU, TRX, TRZ, SEN, SNR, SNS.

TU A level which is high when the Instruction Register contains TRU order.

$\sum$W A level which is high whenever any switch is depressed.

W An output identical to CMW with a delay.

WAN^i ISR^i decoder output "Write Alphanumeric".

WAR Write amplifier flip-flops in FR-300.

WB^i Word Block control flip-flop in CNV^i: WB^i = 0 when words are specified.

WBC^i_n Word Block Counter in CNV^i.

(WBC^i = 0) Word Block Counter contains zeroes.

WBI^i A level which is high whenever a transfer into WBC^i is about to take place.

WCC Write Control Characters flip-flop in the Central Processor.

WCF^i Write control flip-flop.

WCK Write control character flip-flop in the converter.

WEF Write End of File flip-flop in the Central Processor.

WOK^i ISR^i decoder output "Write Octal".

WR^n Level input to Write Pulser of Memory n.

WSR_n Word switch register bit n on console.

WW^i A control level which is high whenever ISR^i holds the code for WAN or WWA.

WWA^i ISR^i decoder output "Rewrite Alphanumeric.

X X-register.

X_n X-register bit n (n = 16 to 22) of instruction word.

X_nD Emitter follower output of X_n bit.

X_nDR Second parallel emitter follower of X_n bit.

X_nDRR Third parallel emitter follower of X_n bit.

XG A control flip-flop used in the single pulse circuit.

$\left[X_{16-22}(IOA^i)\right]$ X - Address decoder output "In-Out Alarm Converter^i" (i = 1 to 4).

$\left[X_{16-22}(IOD)\right]$ X - Address decoder output "In-Out Device".

$\left[X_{16-22}(ISN)\right]$ X - Address decoder output "Interpret Sign".

$\left[X_{16-22}(NHP)\right]$ X - Address decoder output "Ignore Halt on Parity Error".

$\left[X_{16-22}(OA)\right]$ X - Address decoder output "Overflow Alarm".

$\left[X_{16-22}(ROBB)\right]$ X - Address decoder output "Real time output register busy bit flip-flop.

$\left[X_{16-22}(ROP1)\right]$ X - Address decoder output "Real Time output Program Interrupt".

$\left[X_{16-22}(RPE)\right]$ X - Address decoder output "Real Time parity error flip-flop.

$\left[X_{16-22}(SFF^{\beta})\right]$ X - Address decoder output "Sense Flip-Flop$^{\beta}$".

$\left[X_{16-22}(TRA)\right]$	X - Address Decoder output "Tape Erase".
XP	A control flip-flop which is set to one whenever an operation is initiated and is not Program Read-in.
XT	Control flip-flop used in the single gate circuit.
$X_{18}X_{17}X_{16}$	X - Register bits $_{16}$ to $_{18}$ decoded outputs (emitter-followers).
$X_{18}X_{17}X_{16}DR$	X - Register bits $_{16}$ to $_{18}$ decoded outputs (second parallel emitter-followers).
Z_n	A - Register emitter-follower outputs bit. (n = 1 to 36)
ZRO	Fieldata code for zero.
Zt_1^n to Zt_{12}^n	Index register Digit Outputs (emitter follower) bits 1 - 12.
ZT	Zero test control line of AU.
ZZ	A control level which complements the parity flip-flop.
ϕ	Flip-flop which enables in-out memory cycle.
λ	A control FF (normal sequencing of the T. P. D. is interrupted whenever λ is in the one state.)
η	Output generated for conditions $BB^9 + BB^{10}$.
ρ_n	Emitter follower "or" gate output of bit n in A - register carry chain.
θ_n^i	Flip-flop controlling timing of converter cycle.
θ_1	Flip-flops that determine TH-n functions.
θ_2	Flip-flops that determine TH-n functions.

Logical and Mechanization Designations: Card Reader and Punch Buffer

A1CSC	Add 1 to CSC
ALF	Order in converter is an Alphanumeric one
BIN	Inputs from Card-reading brushes
BOUT	Outputs to Card-Punching solenoids
CBA	Card Buffer Alarm Storage
CBAL	Card Buffer Alarm or Off-Line Output
CLSA	Clear CSC (to 0 for RAN or Write, to 1 for ROK)
CMR	Card Buffer Manual Reset
CPE	Card Buffer Parity Error Storage
CPI	Data inputs from DSU (for punching)
CRAN	Card Buffer is doing a RAN order
CREAD	Order in converter is a Read
CRO	Card read data outputs to DSU
CSC	Character Section Counter
CROK	Card Buffer is doing a ROK order
CS_n	Contents of CSC = n
CT	Principal Card Buffer control sequence counter
CTF	Frequency Divider FF which controls p and t levels
DH_n	Highest 3 bits of CT = n
DL_n	Lowest 3 bits of CT = n
LOFF	Off-line/on-line control on OFF LINE
LON	Off-line/on-line control on ON LINE
LR	Have read in last row, or have punched at least one row of the present card

PAL	Card Punch alarm
PARA	Parity error detected at CPI lines from DSU
PBLE	Put EOB code on CRO lines to DSU
PBLS	Put SOB code on CRO lines to DSU
PIN	Input from Card Punch, preventing any PST output to it
PROW	Punch row sync, rows 12 through 8
PROW_9	Punch row sync, row 9
PST	Start one card-punch cycle, to Card Punch
PSTOP	Put STOP code on CRO lines to DSU
PUNCH	Order in Converter is a Card-Punch order
R	80-bit Row register
RAL	Card Reader alarm
RCPE	Reset CPE, input from converter
RDST	Start one card-reading cycle, to Card Reader
RDYI	Ready signal input from converter
RDYO	Ready signal output to converter
RIN	Input from Card Reader, preventing any RDST
RROW	Reader row sync, all rows
RROW9	Reader row sync, row 9 only
RSC	Card-reading station is clear
SCBA	Set CBA
STBI	Strobe pulse input from converter
STBO	Strobe pulse output to converter
STB_n	Strobe pulse input to character section n of R register

TRCB^i	Set up transfer lines between converter i and Card Buffer (for all card operations)
TRCR^i	Set up transfer lines between converter i and Card Buffer (for Card-Reading operation)
TRIR	Read in 80 bits from Card-Reader brushes to R register
TRPCH^i	Set up transfer lines between converter i and Card Buffer (for Card-Punching operation)
TROR	Read out 80 bits from R Register to Punch solenoids

Logical and Mechanization Designation - Real Time System

BB^9	Busy Bit control flip-flop
K-n	Data input lines
KC	Input control bit line
KCF	Input control bit recording flip-flop
KOR	Ready signal from receiving element
KS	Strobe signal input
KSS	Strobe quantizing flip-flop
KSX	Transfer in and shift control flip-flop
PE	Parity error signal
RAR	Real time address register
RINC	Input control bit flip-flop (in central processor)
RIR	Real time input register
RISN	Real time input interpret sign flip-flop
ROB-n	Data output lines
ROBB	Real time output busy flip-flop
ROBP	Output parity signal
ROR	Real time output register

RORC	Output control bit line
ROS	Output strobe flip-flop
ROSN	Real time output interpret sign flip-flop
ROTC	Real time output control character flip-flop
ROX	Ready signal output quantizing flip-flop
ROZ	Output strobe control flip-flop
RTU	Real time unit
SHC	Shift counter

MECHANIZATION LISTS

LOGICAL MECHANIZATION IN-OUT CONVERTER

The mechanization presented in this paragraph consists of converter selection logic, converter stop or destruction logic, converter alarms, the in-out memory cycle, and the in-out instructions.

CONVERTER SELECTION

$(IO)\left[(TP)' + (PR3)\right]\left[\prod_{m=1}^{i-1}(CNV^{m})\right](CNV^{i})' \quad : \left[CSL^{i} = 1\right]$

$(CSL^{i})(t2) \quad : \left[1 => CLRCIS^{i*}\right]$ Note 1

$(CSL^{i})(t3) \quad : \left[IR_{31-36} => ISR^{i}\right]$

$\left[G_{30} => WB^{i}\right]\left[G_{29-28} => WBC^{i}_{29-28}\right]$

$\left[X_{27-22} => WBC^{i}_{27-22}\right]$

$\left[X_{21-16} => DAR^{i}\right]\left[AR => ADC^{i}\right]$

$\left[1 => CF^{i}\right]\left[ISN => IS^{i}\right]\left[0 => ISN\right]$

$[NHC => NH^{i}][TPE => IWR^{i}]$

$[WEF => EFK^{i}][0 => WEF]$

$[0 => TPE][0 => NHC][INT => ID^{i}]$

$[WCC => WCK^{i}][0 => WCC]$

$\left[OTT => OD^{i}\right]\left[IOT => BD^{i}\right]$

$\left[MTT => MT^{i}\right]$ Note 2

$\left[SKP + RAN\right](BD^{i})p4 \quad : \left[1 => MT^{i}\right]$

$(ROK)(BD^{i})p4 \quad : \left[1 => ID^{i}\right]$

$\left[WAN + WOK\right](BD^{i})p4 \quad : \left[1 => OD^{i}\right]$

Note 1: $CLRCIS^{i*}$ is a pulse line used to clear all registers and flip-flops in converter "i". One exception is IGT^{i}. $CLRCIS^{i*}$ sets IGT^{i} to ONE.

Note 2: ISR^{i} WB^{i}, WBC^{i}, DAR^{i}, and ADC^{i} together constitute an addressable register. However, a transfer into DAR^{i} (BITS 16-21) occurs only when an in-out order is programmed.

$(PR1)(t4)$: $\left[1 => ID^i\right]$

$(CSL^i)(ID^i)'(MT^i)'(OD^i)'(BD^i)'(t4)$: $\left[1 => IMO^i\right]$

$(CSL^i)(IMO^i)'(t5)$: $\left[1 => CNV^i\right]\left[1 => GS^i\right]$

$(CSL^i)(IMO^i)(t5)$: $\left[1 => IOA^i\right]$

$(\lambda)'(IO)\left[(\pi CNV^i) + (DVB)\right](t7)$: $\left[1 => \lambda\right]$

$(RAN)'(WAN)'\left[(MM^i)(SKP)' + (MM^i)'(BD^i)(ROK)'(WOK)'\right]$
$+ (EFK^i)\left[WW' + (MT^i)'(MM^i)'\right] + (WCK^i)\left[RW' + WOK\right]$: $\left[ERR^i = 1\right]$

$(GS^i)(CH^i)'(RWB^i)'(D^i_{x1-4} \geq 12\mu s)(ERR^i)t$: $\left[1 => IMO^i\right]\left[0 => GS^i\right]\left[1 => CH^i\right]$

$(BD^i)(MM^i)'(RAN)(RWB^i)(WB^i)'\,t$: $\left[1 => WB^i\right]$

CONVERTER STOP LOGIC

$(RWB^i)\left[(CNV^i)(CH^i)'\right]$ Enables $\theta^i_2\ \theta^i_1$ Timing Functions

$\left[(END^i) + (DVH^i)\right](t)$: $\left[1 => CH^i\right]$

(DVH^i) : $\left[0 => GS^i\right]$

$(CH^i)(BB^i)'(GS^i)'(t8)$: $\left[0 => CNV^i\right]$

CONVERTER ALARMS

$(DVH^i)(t)$: $\left[1 => DVA^i\right]$

$(DVA^i + IPE^i + ISE^i + TRE^i + IMO^i)(IOA)'(t)$: $\left[1 => IOA^i\right]$

$\left[(DVA^i) + (NH^i)'(IPE^i + ISE^i + TRE^i + IMO^i)\right]$: $\left[CTP^i = 1\right]$

$(\sum CTP^i)(t5)$: $\left[1 => TP\right]$

$(BUS26^i)(PBM1)(PRM2)(t)$: $\left[1 => IPE\right]$

IN-OUT MEMORY CYCLE

$\sum \mathrm{BB}^i$ sampled in basic cycle and various instructions and in-out memory cycle.

$$\left(\sum_{m=1}^{n} \mathrm{GA}^n\right)' (\mathrm{Logic})(\Sigma\, \mathrm{BB}^i)(t) \qquad : \left[1 \Rightarrow \mathrm{SGA}^*\right] \left[1 \Rightarrow \phi\right]$$

The BB^i are acknowledged according to:

$$\left\{\left(\sum_{m=9}^{i-1} \mathrm{BB}^m\right)' + \left(\sum_{m=9}^{8+r} \mathrm{BB}^m\right)' \left[\left(\sum_{m=1}^{i-1} \mathrm{BMT}^m\right)' (\mathrm{MT}^i) + \left(\sum_{m=1}^{8} \mathrm{BMT}^m\right)' \left(\sum_{m=1}^{i-1} \mathrm{BB}^m\right)\right]\right\} (\mathrm{BB}^i)(\mathrm{SGA}^*)$$

$$: \left[1 \Rightarrow \mathrm{GA}^i\right]$$

where $(\mathrm{BMT}^m) = (\mathrm{BB}^m)(\mathrm{MT}^m)$

$i = 1, 2, 3, \ldots .\, 8$ for Converters
$i = 9, 10, 11, \ldots (8+r)$ for Real Time Input Registers
r = total number of Real Time Input Registers

The particular GA^i. determines "i" in the equations below:

$(\mathrm{TC1})(\phi)(\mathrm{PR0} + \mathrm{PR3}) \qquad : \left[\mathrm{RE}^n = 1\right]$

$(\mathrm{TC1})(\phi)(t) \qquad : \left[\mathrm{ADC}^i_{1\text{-}12} \Rightarrow \mathrm{MA}^n\right]$

$(\mathrm{TC2})(\phi)(t)(\mathrm{RW}^i)' \qquad : \left[\mathrm{BFR}^i \Rightarrow \mathrm{MOR}^n\right]$

$(\mathrm{TC2})(\phi)(t) \qquad : \left[0 \Rightarrow \mathrm{BFR}^i\right]$

$(\mathrm{TC2})(\phi)(t)(\mathrm{PR1}) \qquad : \left[\mathrm{BFR}^i \Rightarrow \mathrm{IR, G, X, AR}\right]\left[1 \Rightarrow \mathrm{PR2}\right]$

$(\mathrm{TC2})(\phi)(t)(\mathrm{PR2}) \qquad : \left[\mathrm{BFR}^i_{1\text{-}15} \Rightarrow \mathrm{PC}\right]$

$(\mathrm{TC3})(\phi)(\mathrm{PR0} + \mathrm{PR3}) \qquad : \left[\mathrm{WR}^n = 1\right]$

$(\mathrm{TC3})(\phi)(t)(\mathrm{RW}^i) \qquad : \left[\mathrm{MOR}^n \Rightarrow \mathrm{BFR}^i\right]$

$(\mathrm{TC3})(\phi)(t)(\mathrm{ADC}^i_{13\text{-}15} = 7)' \qquad : \left[(\mathrm{ADC}^i + 1) \Rightarrow \mathrm{ADC}^i\right]$

$$(TC3)(\phi)(t) \quad : \left[0 => BB^{i}\right]\left[1 => CLRGA*\right]$$

$$(TC4)(\phi)(t)\left[(\sum BB^{i})' + (T \leq 30)'(MLY + DVD)\right] \quad : \left[0 => \phi\right]$$

$$(TC4)(\phi)(t)(\sum BB^{i})\left[(T \leq 30) + (MLY)'(DVD)'\right] \quad : \left[1 => SGA*\right]$$

$$(TF8)(t)(\sum BB^{i})' \left\{(TP + HT)'\left[(IO)' + (\pi CNV^{i})'(DVB)'\right] + (\sum CNV^{i})'\right\}(\lambda) : [0 => \lambda]$$

IN-OUT INSTRUCTIONS

Note: The superscript "i" has been omitted from the remainder of this mechanization. It applies to all functions except "IR" and "IO."

READ ALPHANUMERIC

$$(IR = 70) \quad : \left[IO => 1\right]$$

$$(ISR = 70) \quad : \left[RAN = 1\right]\left[RR = 1\right]$$

$$(RAN)\left\{(WBC = 0)\left[MT + (IS)(WB)'\right] + OD\right\}(CSL)(t4) \quad : \left[1 => IMO\right]$$

1) $(\theta_2)'(\theta_1)'(p) \quad : \left[0 => CIR\right]$

$(\theta_2)'(\theta_1)'(p)(RAN)\left[(MT)' + BLS\right] \quad : \left[0 => CF\right]$

$(\theta_2)'(\theta_1)'(t)(RR)(BSR = 0)' \quad : \left[BSR => CIR\right]\left[0 => BSR\right]$

$(\theta_2)'(\theta_1)'(t)(RR)(BSR = 0)'(CF)' \quad : \left[1 => \theta_1\right]$

$(\theta_2)'(\theta_1)'(TRE)(BB)' \quad : \left[END = 1\right]$

2) $(\theta_2)'(\theta_1)(t)(RAN)(N = 0)(BLE)'\left[IS' + WCK\right] \quad : \left[1 => N_1\right]$

$(\theta_2)'(\theta_1)(t)(RAN)(N = 0)(IS)(ZRO)'(ONE)'(BLE)'(WCK)' \quad : \left[1 => ISE\right]$

$(\theta_2)'(\theta_1)(t)(BB)' \quad : \left[1 => \theta_2\right]$

3) $(\theta_2)(\theta_1)(p)(RAN) \quad : \left[1 => SLB*\right]$

$(\theta_2)(\theta_1)(p)(RAN)\left[(BLE)' + (N = 0)'\right] \quad : \left[(N + 1) => N\right]$

$(\theta_2)(\theta_1)(t)(RAN)\left[(MT' + WB')(N = 7) + (MT)(WB)(BLE)\right]$

$(WBI)'\left[(WBC = 0)' + (RR' + MT'\ IS)(WB)\right] \quad : \left[(WBC - 1) => WBC\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{RAN}) \quad : \left[1 => \mathrm{SLB}^*\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{RR}) \left[(\mathrm{MT}) + (\mathrm{BUS}_{15})'\right] (\mathrm{PFM} \oplus \mathrm{CIR}_8) \quad : \left[1 => \mathrm{IPE}\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{RAN}) \left[(\mathrm{MT})'(\mathrm{IS})'(\mathrm{WB})' + (N = 7)\right] \quad : \left[1 => \mathrm{BB}\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{RAN}) \left[(N = 0)'(N = 7)'(\mathrm{BLE})(\mathrm{MT})' \left\{\mathrm{WCK} + (\mathrm{IS})'(\mathrm{WB})\right\}\right]' \quad : \left[0 => \theta_1\right]$

4) $(\theta_2)(\theta_1)'(\mathrm{RAN}) \left\{(\mathrm{MT}) \left[(\mathrm{WBC} = 0) + (\mathrm{TRE})\right]\right.$

$\left. + (\mathrm{MT})' \left[(\mathrm{BLE}) + (\mathrm{WB})' \left\{(\mathrm{WBC} = 0) + (\mathrm{IS})'\right\}\right]\right\} \quad : \left[\mathrm{END} = 1\right]$

$(\theta_2)(\theta_1)'(\mathrm{RAN})(\mathrm{BLE})(\mathrm{MT}) \left[\mathrm{EFK} + \mathrm{ETK}\right] \quad : \left[\mathrm{END} = 1\right]$

$(\theta_2)(\theta_1)'(p)(\mathrm{RAN})(\mathrm{BLE})' \quad : \left[\mathrm{CIR}_{1-6} => \mathrm{BFR}_{1-6}\right]$

$(\theta_2)(\theta_1)'(t)(\mathrm{RR})(N = 7) \quad : \left[0 => N\right]$

$(\theta_2)(\theta_1)'(t)(\mathrm{RR}) \quad : \left[0 => \theta_2\right]$

$(\theta_2)(\theta_1)'(t)(\mathrm{RAN})(\mathrm{BLE})(N = 0) \quad : \left[1 => \mathrm{CF}\right]$

$(\theta_2)(\theta_1)'(t)(\mathrm{RAN}) \left\{(\mathrm{BLS}) + (\mathrm{BLE})(N = 0)' \left[(\mathrm{MT}) + (\mathrm{IS})\right]\right\} \quad : \left[1 => \mathrm{TRE}\right]$

READ REVERSE

$(\mathrm{IR} = 71) \quad : \left[\mathrm{IO} = 1\right]$

$(\mathrm{ISR} = 71) \quad : \left[\mathrm{RRV} = 1\right] \left[\mathrm{RR} = 1\right]$

$(\mathrm{RRV}) \left[(\mathrm{WBC} = 0) + (\mathrm{MT})'\right] (\mathrm{CSL})(t4) \quad : \left[1 => \mathrm{IMO}\right]$

1) $(\theta_2)'(\theta_1)'(p) \quad : \left[0 => \mathrm{CIR}\right]$

$(\theta_2)'(\theta_1)'(p)(\mathrm{RRV})(\mathrm{BLE}) \quad : \left[0 => \mathrm{CF}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{RR})(\mathrm{BSR} = 0)' \quad : \left[\mathrm{BSR} => \mathrm{CIR}\right] \left[0 => \mathrm{BSR}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{RR})(\mathrm{BSR} = 0)'(\mathrm{CF})' \quad : \left[1 => \theta_1\right]$

$(\theta_2)'(\theta_1)'(\mathrm{TRE})(\mathrm{BB})' \quad : \left[\mathrm{END} = 1\right]$

2) $(\theta_2)'(\theta_1)(t)(\mathrm{RRV})(N = 0)(\mathrm{BLS})' \left[\mathrm{IS}' + \mathrm{WCK}\right] \quad : \left[1 => N_1\right]$

$(\theta_2)'(\theta_1)(t)(\mathrm{RRV})(N = 6)(\mathrm{IS})(\mathrm{ZRO})'(\mathrm{ONE})'(\mathrm{WCK})' \quad : \left[1 => \mathrm{ISE}\right]$

$(\theta_2)'(\theta_1)(t)(\mathrm{BB})' \quad : \left[1 => \theta_2\right]$

3) $(\theta_2)(\theta_1)(p)(RRV)\left[(N = 6)' + (IS)'\right]$: $\left[1 \Rightarrow SLB^*\right]$

$(\theta_2)(\theta_1)(p)(RRV)(BLS)'$: $\left[(N + 1) \Rightarrow N\right]$

$(\theta_2)(\theta_1)(t)(RRV)\left[(WB)'(N = 7) + (WB)(BLS)\right]$

  $(WBI)'\left[(WBC = 0)' + (RR' + MT'\ IS)(WB)\right]$: $\left[(WBC - 1) \Rightarrow WBC\right]$

$(\theta_2)(\theta_1)(t)(RRV)\left[(N = 7)' + (IS)'\right]$: $\left[1 \Rightarrow SLB^*\right]$

$(\theta_2)(\theta_1)(t)(RR)\left[(MT) + (BUS_{15})'\right](PFM \oplus CIR_8)$: $\left[1 \Rightarrow IPE\right]$

$(\theta_2)(\theta_1)(t)(RRV)(N = 7)$: $\left[1 \Rightarrow BB\right]$

$(\theta_2)(\theta_1)(t)(RRV)$: $\left[0 \Rightarrow \theta_1\right]$

4) $(\theta_2)(\theta_1)'(RRV)\left[(WBC = 0) + (TRE)\right]$: $\left[END = 1\right]$

$(\theta_2)(\theta_1)'(p)(RRV)\left[(N = 7)' + (IS)'\right]\left[BLS\right]'$: $\left[CIR_{1-6} \Rightarrow BFR_{1-6}\right]$

$(\theta_2)(\theta_1)'(p)(RRV)(N = 7)(IS)(CIR_1)'$: $\left[0 \Rightarrow BFR_{sn}\right]$

$(\theta_2)(\theta_1)'(p)(RRV)(N = 7)(IS)(CIR_1)$: $\left[1 \Rightarrow BFR_{sn}\right]$

$(\theta_2)(\theta_1)'(t)(RR)(N = 7)$: $\left[0 \Rightarrow N\right]$

$(\theta_2)(\theta_1)'(t)(RR)$: $\left[0 \Rightarrow \theta_2\right]$

$(\theta_2)(\theta_1)'(t)(RRV)(BLS)(N = 0)$: $\left[1 \Rightarrow CF\right]$

$(\theta_2)(\theta_1)'(t)(RRV)\left[(BLS)(N = 0)' + BLE\right]$: $\left[1 \Rightarrow TRE\right]$

$(\theta_2)(\theta_1)'(RRV)(BLS)\left[EFK + ETK\right](MT)$: $\left[END = 1\right]$

READ OCTAL

$(IR = 72)$: $\left[IO = 1\right]$

$(ISR = 72)$: $\left[ROK = 1\right]$

$(ROK)\left[(WB)'(WBC = 0) + (ID)'\right](CSL)(t4)$: $\left[1 \Rightarrow IMO\right]$

1) $(\theta_2)'(\theta_1)'(p)$: $\left[0 \Rightarrow CIR\right]$

$(\theta_2)'(\theta_1)'(p)(ROK)$: $\left[0 \Rightarrow CF\right]$

$(\theta_2)'(\theta_1)'(t)(ROK)(BSR = 0)'$: $\left[BSR \Rightarrow CIR\right]\left[0 \Rightarrow BSR\right]\left[1 \Rightarrow \theta_1\right]$

$(\theta_2)'(\theta_1)'(t)(ROK)$: $\left[1 \Rightarrow IS\right]$

$(\theta_2)'(\theta_1)'(TRE)(BB)'$: $\left[END = 1\right]$

2) $(\theta_2)'(\theta_1)(t)(\mathrm{ROK})(N = 0)(\mathrm{ZRO})'(\mathrm{ONE})'(\mathrm{LC})$: $\left[1 \Rightarrow \mathrm{ISE}\right]$

$(\theta_2)'(\theta_1)(t)(\mathrm{BB})'$: $\left[1 \Rightarrow \theta_2\right]$

$(\theta_2)'(\theta_1)(t)(\mathrm{ROK})(N = 0)'(\mathrm{LC})'$: $\left[1 \Rightarrow \mathrm{TRE}\right]$

3) $(\theta_2)(\theta_1)(p)(\mathrm{ROK})(\mathrm{LC})$: $\left[1 \Rightarrow \mathrm{SLB}^*\right]\left[(N + 1) \Rightarrow N\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{ROK})(N = 13)$: $\left[1 \Rightarrow \mathrm{BB}\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{ROK})(N = 13)(\mathrm{WBI})'\big[(\mathrm{WBC} = 0)'$
$+ (\mathrm{RR}' + \mathrm{MT}'\mathrm{IS})(\mathrm{WB})\big]$: $\left[(\mathrm{WBC} = 1) \Rightarrow \mathrm{WBC}\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{ROK})$: $\left[0 \Rightarrow \theta_1\right]$

$(\theta_2)(\theta_1)(t)(\mathrm{ROK})(\mathrm{BUS}_{15})'(\mathrm{PFM} \oplus \mathrm{CIR}_8)$: $\left[1 = \mathrm{IPE}\right]$

4) $(\theta_2)(\theta_1)'(\mathrm{ROK})\left[(\mathrm{WB})'(\mathrm{WBC} = 0) + \mathrm{BLE}\right]$: $\left[\mathrm{END} = 1\right]$

$(\theta_2)(\theta_1)'(p)(\mathrm{ROK})(\mathrm{LC})$: $\left[\mathrm{CIR}_{1-3} \Rightarrow \mathrm{BFR}_{1-3}\right]$

$(\theta_2)(\theta_1)'(t)(\mathrm{ROK})(N = 13)$: $\left[0 \Rightarrow N\right]$

$(\theta_2)(\theta_1)'(t)(\mathrm{ROK})$: $\left[0 \Rightarrow \theta_2\right]$

WRITE ALPHANUMERIC

$(\mathrm{IR} = 74)$: $\left[\mathrm{IO} = 1\right]$

$(\mathrm{ISR} = 74)$: $\left[\mathrm{WAN} = 1\right]\left[\mathrm{WW} = 1\right]\left[\mathrm{RW} = 1\right]$

$(\mathrm{WAN})\left[(\mathrm{WBC} = 0)(\mathrm{WB})' + (\mathrm{ID})\right](\mathrm{CSL})(t4)$: $\left[1 \Rightarrow \mathrm{IMO}\right]$

1) $(\theta_2)'(\theta_1)'(p)$: $\left[0 \Rightarrow \mathrm{CIR}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{WAN})(\mathrm{MT})'(\mathrm{CF})$: $\left[1 \Rightarrow \mathrm{BB}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{WAN})(\mathrm{MT})(\mathrm{CF})\left[(\mathrm{WB}) + (\mathrm{WBC} = 0)'\right]$: $\left[\mathrm{BLS} \Rightarrow \mathrm{CIR}_{1-8}\right]\left[1 \Rightarrow \mathrm{BB}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{WAN})(\mathrm{MT})(\mathrm{CF})$: $\left[1 \Rightarrow \theta_2\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{RW})$: $\left[0 \Rightarrow \mathrm{CF}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{WW})(\mathrm{WB})'(\mathrm{WBC} = 0)$: $\left[\mathrm{BLE} \Rightarrow \mathrm{CIR}_{1-8}\right]$

$(\theta_2)'(\theta_1)'(t)(\mathrm{RW})(\mathrm{CF})'(\mathrm{BB})'$: $\left[1 \Rightarrow \theta_1\right]$

2) $(\theta_2)'(\theta_1)(t)(RW)(N = 0) \quad : \left[1 \Rightarrow N_1\right]$

$(\theta_2)'(\theta_1)(t)(RW)(N = 0)(IS)(WCK)' \quad : \left[0 \Rightarrow \theta_1\right]$

$(\theta_2)'(\theta_1)(t)(RW)(N = 0)(IS)(BFR_{sn})'(WCK)' \quad : \left[ZRO \Rightarrow CIR_{1-6}\right]$

$(\theta_2)'(\theta_1)(t)(RW)(N = 0)(IS)(BFR_{sn})(WCK)' \quad : \left[ONE \Rightarrow CIR_{1-6}\right]\left[1 \Rightarrow CIR_8\right]$

$(\theta_2)'(\theta_1)(t)(WW)\left[(N = 0)' + (IS)' + WCK\right] \quad : \left[BFR_{31-36} \Rightarrow CIR_{1-6}\right]$

$(\theta_2)'(\theta_1)(t) \quad : \left[1 \Rightarrow \theta_2\right]$

$(\theta_2)'(\theta_1)(t)(RW) \quad : \left[1 \Rightarrow CIR_7\right]$

3) $(\theta_2)(\theta_1)(p)(RW) \quad : \left[1 \Rightarrow SLB^*\right]\left[(N + 1) \Rightarrow N\right]$

$(\theta_2)(\theta_1)(t)(WW)(N - 7)(WBI)'\left[(WBC = 0)'\right.$

$\left.+ (RR' + MT'IS)(WB)\right] \quad : \left[(WBC - 1) \Rightarrow WBC\right]$

$(\theta_2)(\theta_1)(t)(RW)(PFM) \quad : \left[1 \Rightarrow CIR_8\right]$

$(\theta_2)(\theta_1)(t)(WW) \quad : \left[1 \Rightarrow SLB^*\right]\left[0 \Rightarrow \theta_1\right]$

4) $(\theta_2)(\theta_1)'(RW)(WB)'(WBC = 0)\left[(MT)' + BLE\right]$

$\left[BSR = 0\right] \quad : \left[END = 1\right]$

$(\partial_2)(\theta_1)'(t)(WW)(N = 7) \quad : \left[0 \Rightarrow N\right]$

$(\theta_2)(\theta_1)'(t)(WW)\left[(WB) + (WBC = 0)'\right](N = 7) \quad : \left[1 \Rightarrow BB\right]$

$(\theta_2)(\theta_1)'(t)(WW)(WB)'(WBC = 0) \quad : \left[1 \Rightarrow CF\right]$

$(\theta_2)(\theta_1)'(t)(RW)(BSR = 0) \quad : \left[CIR \Rightarrow BSR\right]$

$(\theta_2)(\theta_1)'(t)(RW)(BSR = 0) \quad : \left[0 \Rightarrow \theta_2\right]$

REWRITE ALPHANUMERIC

$(IR = 75) \quad : \left[IO = 1\right]$

$(ISR = 75) \quad : \left[WWA = 1\right]\left[WW = 1\right]\left[RW = 1\right]$

$(WWA)\left[(WB)'(WBC = 0) + (MT)'\right](CSL)(t4) \quad : \left[1 \Rightarrow IMO\right]$

1) $(\theta_2)'(\theta_1)'(p) \quad : \left[0 \Rightarrow CIR\right]$

$(\theta_2)'(\theta_1)'(t)(WWA)(CF)\left[(WB) + (WBC = 0)'\right] : \left[1 \Rightarrow BB\right]$

$(\theta_2)'(\theta_1)'(t)(WWA)(WB)'(WBC = 0) \quad : \left[1 \Rightarrow \theta_2\right]$

$(\theta_2)'(\theta_1)'(t)(WW)(WB)'(WBC = 0) \quad : \left[BLE \Rightarrow CIR_{1-8}\right]$

$(\theta_2)'(\theta_1)'(t)(RW) \quad : \left[0 \Rightarrow CF\right]$

$(\theta_2)'(\theta_1)'(t)(RW)(CF)'(BB)' \quad : \left[1 \Rightarrow \theta_1\right]$

2-4. Same as WAN

WRITE OCTAL

$(IR = 76) \quad : \left[IO = 1\right]$

$(ISR = 76) \quad : \left[WOK = 1\right]\left[RW = 1\right]$

$(WOK)\left[(WB)'(WBC = 0) + (OD)'\right](CSL)(t4) \quad : \left[1 \Rightarrow IMO\right]$

1) $(\theta_2)'(\theta_1)'(p) \quad : \left[0 \Rightarrow CIR\right]$

$(\theta_2)'(\theta_1)'(t)(WOK)(CF) \quad : \left[1 \Rightarrow BB\right]$

$(\theta_2)'(\theta_1)'(t)'(RW) \quad : \left[0 \Rightarrow CF\right]$

$(\theta_2)'(\theta_1)'(t)(RW)(CF)'(BB)' \quad : \left[1 \Rightarrow \theta_1\right]$

$(\theta_2)'(\theta_1)'(t)(WOK) \quad : \left[1 \Rightarrow IS\right]$

2) $(\theta_2)'(\theta_1)(t)(RW)(N = 0)$: $\left[1 => N_1\right]$

$(\theta_2)'(\theta_1)(t)(RW)(N = 0)(IS)$: $\left[0 => \theta_1\right]$

$(\theta_2)'(\theta_1)(t)(RW)(N = 0)(BFR_{sn})'(IS)$: $\left[ZRO => CIR_{1-6}\right]$

$(\theta_2)'(\theta_1)(t)(RW)(N = 0)(BFR_{sn})(IS)$: $\left[ONE => CIR_{1-6}\right]\left[1 => CIR_8\right]$

$(\theta_2)'(\theta_1)(t)(WOK)(N = 0)'$: $\left[BFR_{34-36} => CIR_{1-3}\right]$

$(\theta_2)'(\theta_1)(t)$: $\left[1 => \theta_2\right]$

$(\theta_2)'(\theta_1)(t)(WOK)$: $\left[1 => CIR_5\right]\left[1 => CIR_6\right]$

$(\theta_2)'(\theta_1)(t)(RW)$: $\left[1 => CIR_7\right]$

3) $(\theta_2)(\theta_1)(p)(RW)$: $\left[1 => SLB^*\right]\left[(N + 1) => N\right]$

$(\theta_2)(\theta_1)(t)(WOK)(N = 13)(WBI)'\left[(WBC = 0)' + (RR' + MT'IS)(WB)\right]$: $\left[(WBC - 1) => WBC\right]$

$(\theta_2)(\theta_1)(t)(WOK)$: $\left[0 => \theta_1\right]$

$(\theta_2)(\theta_1)(t)(RW)(PFM)$: $\left[1 => CIR_8\right]$

4) $(\theta_2)(\theta_1)'(RW)(WB)'(WBC = 0)\left[(MT)' + BLE\right]\left[BSR = 0\right]$: $\left[END = 1\right]$

$(\theta_2)(\theta_1)'(t)(WOK)(N = 13)$: $\left[0 => N\right]$

$(\theta_2)(\theta_1)'(t)(WOK)\left[(WB) + (WBC = 0)'\right](N = 13)$: $\left[1 => BB\right]$

$(\theta_2)(\theta_1)'(t)(RW)(BSR = 0)$: $\left[CIR => BSR\right]$

$(\theta_2)(\theta_1)'(t)(RW)(BSR = 0)$: $\left[0 => \theta_2\right]$

SKIP

$(IR = 66)$: $[IO = 1]$

$(ISR = 66)$: $[SKP = 1][SS = 1]$

$(SS)\left[(WBC = 0) + (MT)'\right](CSL)(t4)$: $[1 \Rightarrow IMO]$

1) $(\theta_2)'(\theta_1)'(p)$: $[0 \Rightarrow CIR]$

$(\theta_2)'(\theta_1)'(p)(SKP)(BLE)$: $[0 \Rightarrow CF]$

$(\theta_2)'(\theta_1)'(t)(SS)(BSR = 0)'$: $[BSR \Rightarrow CIR][0 \Rightarrow BSR]$

$(\theta_2)'(\theta_1)'(t)(SS)(CF)'$: $[1 \Rightarrow \theta_1]$

2) $(\theta_2)'(\theta_1)(t)(BB)'$: $[1 \Rightarrow \theta_2]$

3) $(\theta_2)(\theta_1)(t)(SS)(WBI)'\left[(WBC = 0)' + \{(RR)'(SS)' + (MT)'(IS)\}(WB)\right]\left[EFK + WB'\right]$: $[(WBC - 1) \Rightarrow WBC]$

$(\theta_2)(\theta_1)(t)(SS)$: $[0 \Rightarrow \theta_1]$

4) $(\theta_2)(\theta_1)'(SS)(WBC = 0)$: $[END = 1]$

$(\theta_2)(\theta_1)'(t)(SS)$: $[1 = CF]$

$(\theta_2)(\theta_1)'(t)(SS)$: $[0 \Rightarrow \theta_2]$

$(\theta_2)(\theta_1)'(SS)\left[WB' + EFK\right](ETK)$: $[END = 1]$

$(\theta_2)(\theta_1)'(SS)(WBC = 0)'(EFK)(t)$: $[0 \Rightarrow EFK]$

BACKSPACE

(IR = 67) : [IO = 1]

(ISR = 67) : [BSP = 1] [SS = 1]

(SS) [(WBC = 0) + (MT)'] (CSL)(t4) : [1 => IMO]

1) $(\theta_2)'(\theta_1)'(p)$: [0 => CIR]

$(\theta_2)'(\theta_1)'(p)$(BSP)(BLS) : [0 => CF]

$(\theta_2)'(\theta_1)'(t)$(SS)(BSR = 0)' : [BSR => CIR] [0 => BSR]

$(\theta_2)'(\theta_1)'(t)$(SS)(CF)' : [1 => θ_1]

2 - 4. SAME AS SKP.

REWIND

(IR = 77) : [IO = 1]

(ISR = 77) : [RWD = 1]

1) (RWD)(MT)'(CSL)(t4) : [1 => IMO]

MAGNETIC TAPE CONTROL - READ

1) (GS)(CH)'(RWB)'(DX)'(t) : [1 => DX]

(RWB)'($D_{x1-4} \geq 12\mu$sec)(t) : [0 => DX][0 => D_x]

(GS)(CH)'(RWB)'($D_{x1-4} \geq 12\mu$sec)[(MT)(BD)'(BUS13)' + [(MT)(BD)']'(ERR)'](t) : [1 => RWB]

(MT)(GS)(CH)'(RH)'(RWD)'(RWB)(t) : [1 => RH]

(MT)(GS)(CH)'(DX)'[(RWD) + (RH)'](RWB)(BD)'(t) : [1 => DX]

(RWD)(GS)(RWB)(RH)'(CH)' : [BUS13 = 1]

2) (GS)(CH)'(RH)'(RRV + BSP)(RV)'(t) : [1 => RV]

(GS)(RH)(RV) : [BUS9 = 1]

(GS)(RH)(RV)' : [BUS10 = 1]

3) (MT)(GS)(CH)'(RCF)'[(D_x = A) + BD](RR + SS + WWA)(t) : [1 => RCF][0 => DX][0 => D_x]

(MT)(GS)(CH)'(PBM2)'(TAR = 0)'(DX)'(RR + SS + WWA)[(BD)' + (RWB)'](t) : [1 => DX]

(MT)(GS)(CH)'(PBM2)'($D_x = 34_{\mu s}$)(RCF)(RR + SS + WWA)(t) : [0 => DX][0 => D_x][0 => TAR][0 => AF_{1-3}][0 => PBM1]

(MT)(GS)(CH)'(PBM2)'($D_x = 34\mu$sec)(RCF)(RR + SS)(t) : [0 => EFK][0 => CFK]

(BCD + TBLSD) : [TBLS = 1]

(ECD + TBLED + EFD) : [TBLE = 1]

(RAN + SKP + WWA)(TBLS) + (RRV + BSP)(TBLE) : [QA = 1]

(MT)(GS)(CH)'(RCF)(CAD)(EFD)(EFK)'(PBM1)'(PBM2)'(RRV + BSP)(t) : [1 => EFK]

(MT)(GS)(CH)'(PBM1)'(PBM2)'(CAD)[(ECD)(RRV) + (BCD)(RAN)](t) : [1 => CFK]

(MT)(GS)(CH)'(PBM1)'(PBM2)'(QA)(CAD)(t) : [1 => PBM1][0 => D_x][0 => AF_{1-3}]

$(MT)(GS)(CH)'(PBM1)(PBM2)'(QA)(CAD)(t) \quad : [0 \Rightarrow AF_{1-3'}]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(EFK)(EFD)(CAD)(RRV)(t) \quad : [1 \Rightarrow WCK]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(QA)(CAD)(WWA)'(t) \quad : [1 \Rightarrow PBM2]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(WCF)(WWA)(D_x4)(D_x6)(t) \quad : [1 \Rightarrow PBM2] [0 \Rightarrow D_x] [0 \Rightarrow TAR]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(QA)(CAD)(RR + SS)(t) \quad : [TAR \Rightarrow BXR] [0 \Rightarrow DX] [0 \Rightarrow D_x]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(RR + SS)(CAD)(BCD)(CFK)(p) \quad : [TBLS \Rightarrow TAR]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(RR + SS)(CAD)[(EFD)(EFK) + (ECD)(CFK)](p) \quad : [TBLE \Rightarrow TAR]$

$(MT)(GS)(CH)'(PBM1)(PBM2)'(QA)(CAD)(WWA)(t) \quad : [0 \Rightarrow D_x] [0 \Rightarrow RCF] [0 = WAR] [1 \Rightarrow WCF]$

$(D_x = 34\mu s)(t) \quad : [0 \Rightarrow AF_{1-3}]$

Note: Bus 9 contains Go Reverse Level.
Bus 10 contains Go Forward Level.
Bus 11 contains Read Level.
Bus 13 contains Rewind Level.
Bus 23 contains Ready Level.
Bus 24 contains Strobe Level.
Bus 12 contains Write Level.

$(RCF)(IGT)'(SCL)'$: $\left[BUS_n \Rightarrow TAR_n\right]$

$(MT)(RCF)\Big[(\sum_{n=1}^{n=8} BUS)'(IGT)(SCL)' + (SCL)$

$(STB)'\left[CAD + (TAR = 0)\right]\Big]\ (t)$: $\left[0 \Rightarrow IGT\right]$

$(MT)(IGT)'(CAD)'(TAR_7 + TAR_6)(RCF)$

$(D_x = 34\mu s)'(SCL)'(t)$: $\left[(AF + 1) \Rightarrow AF\right]$

$(MT)\Big[(SCL)'(AF_1)(AF_2)'(AF_3) + (SCL)$

$(IGT)(TAR = 0)'(BXR = 0)\Big]$: $\left[CAD = 1\right]$

$(MT)(CAD)(SCL)'(t)$: $\left[0 \Rightarrow AF_{1-3}\right]$

$(RCF)(CAD)(GCL)'(t)$: $[1 \Rightarrow IGT]$

$(SCL)(RCF)(IGT)'(GS)(TAR = 0)(RWB)$

$\left[(MT)(BD)' + (MM)\right]$: $\left[BUS23 = 1\right]$

$(SCL)(RCF)(STB)'(BUS24)$: $\left[1 \Rightarrow STB\right]$

$(SCL)(RCF)(IGT)'(BUS24)$: $\left[BUS_n \Rightarrow TAR_n\right]$

$(RCF)(STB)(t)$: $\left[1 \Rightarrow IGT\right]$

$(RCF)(STB)(IGT)(BUS24)'(p)$: $\left[0 \Rightarrow STB\right]$

$(MT)(GS)(RCF)(RWB)$: $\left[BUS11 = 1\right]$

$(MT)(RCF)\left[(IGT)(SCL)' + (SCL)(CAD)\right]$

$(TAR = 0)'(t)$: $\left[0 \Rightarrow TAR\right]$

4) NO ACTION.

5) $(MT)(RCF)(PBM1)(PBM2)(CAD)(CH)'$

$(TBLS + TBLE)'(t)$: $\left[TAR \Rightarrow BXR\right]\ \left[0 \Rightarrow AF_{1-3}\right]$

$(MT)(RCF)(CAD)(BXR = 0)'(t)$: $\left[1 \Rightarrow TRE\right]\ \left[0 \Rightarrow AF_{1-3}\right]$

$(RCF)(BSR = 0)(BXR = 0)'(t)$: $\left[BXR \Rightarrow BSR\right]\ \left[0 \Rightarrow BXR\right]$

(MT)(RCF)(PBM1)(PBM2)(TBLS + TBLE)(CAD)(t) : [0 => PBM1][0 => AF_{1-3}]

(MT)(GS)(CH)'(RCF)(CAD)(EFD)(EFK)'(PBM1)(PBM2)(t) : [1 => EFK]

(MT)(GS)(CH)'(PBM1)(PBM2)(CFK)(WCK)'[EFK'+ (TAR_7)'(CAD)](RR)(t) : [1 => WCK]

(MT)(GS)(CH)'(PBM1)(PBM2)(CFK)(EFK)(CAD)(RR)(t) : [0 => CFK]

(MT)(RCF)(PBM1)(PBM2)(TBLS + TBLE)(CAD)[BD' + RWB'](t) : [1 => DX]

(MT)(RCF)(PBM1)'(PBM2)(TBLS + TBLE)(CAD)(t) : [0 => PBM2][0 => AF_{1-3}][0 => DX][0 => D_x]

(MT)(RCF)(PBM1)'(PBM2)(TBLS + TBLE)(CAD)(CH)'(t) : [TAR => BXR]

(MT)(GS)(CH)'(PBM1)'(PBM2)(RR + SS)(CAD)(BCD)(CFK)(p) : [TBLS => TAR]

(MT)(GS)(CH)'(PBM1)'(PBM2)(RR + SS)(CAD)[(EFD)(EFK) + (ECD)(CFK)](p) : [TBLE => TAR]

(MT)(RCF)(PBM1)'(PBM2)($D_x = 34_{\mu s}$)(t) : [1 => TRE][0 => DX][0 => D_x][0 => AF_{1-3}]

6) (MT)(GS)(CH)(PBM1)'(PBM2)'(RCF)(DX)'[BD' + RWB'](t) : [1 => DX]

(MT)(GS)(CH)(PBM1)'(PBM2)'(RCF)(DX)'(t) : [0 => RCF][0 => BXR][0 => BSR]

(D_xE)(t) : [0 => GS][0 => RH][0 => RCK][0 => WCF]
[0 => WAR][1 => TRE][0 => DX][0 => D_x]

7) (MT)(CH)'(END)(RR + SS)(t) : [1 => ORF1]

(MT)(CH)'(END)(RAN + SKP + WW)(t) : [1 => ORF2]

(RWD)(D_x = $24_{\mu s}$)(CH)'(RH)'(t) : [0 => GS][0 => DX][0 => D_x][1 => CH]

(MT)(GS)(CH)'(ETK)'(BUS_{28})(t) : [1 => ETK]

(MT)(CH)(ISG)'(ETK) : [CETA = 1]

(CETA)(t) : [0 => ETK]

(MT)(ISG)'(EFK){(CH)[RR + (SS)(WB)] + (WB)'(SS)(PBM1 ⊕ PBM2)'} : [CEF = 1] [CEF - C = 1]

{(MT)(GS)(CH)'(PBM2)'(D_x = 34μsec)(RCF)(RR + SS + WWA) + CEF}(t) : [0 => EFK]

(MT)(GS)(CH)'(ISG)'(CFK)(WCK)(RR)(PBM1 ⊕ PBM2)'(EFK' + CEF) : [CFC - C = 1][CFC = 1]

(MT)(GS)(CH)'(ISG)'(CFK)(WCK)(RR)(PBM1)'(PBM2)'(EFK' + CEF)(t) : [0 => WCK]

(CFC)(t) : [0 => CFK]

(EFK + ETK + CTP) : [CC21 = 1]

(MT)(GS)(CH)(ORF1)(ORF2)(D_x = A)
[(WBC = 0) + (RRV + BSP + RWD)] (RH)(t) : [0 => D_x]

(MT)(GS)(CH)(ORF1)(ORF2)(D_x = A)[(WBC = 0) +
CC21 + RRV + BSP + RWD] (RH)(t) : [0 => RH]

(MT)(GS)(CH)(ORF1)(ORF2)(D_x = A
[(WBC = 0)'(RAN + SKP + WWA)] (t) : [0 => D_x] [1 => CF]

(MT)(GS)(CH)(ORF1)(ORF2)(D_x = A)
[(WBC = 0)'(CC21)'(RAN + SKP + WWA)] (t) : [0 => CH] [0 => ORF1]
[=> ORF2]

(MT)(GS)(CH)(ORF1)(ORF2)(D_x = A)
[(WBC = 0)'(WAN)] (t) : [1 => WCF] [0 => WAR]
[0 => CH] [0 => D_x] [1 => CF]

(MT)(GS)(CH)(ORF1)(ORF2)(RH)'(D_x = C)
(WBC = 0)'(t) : [0 => DX] [0 => D_x] [1 => CF]
[0 => ORF1] [0 => ORF2]

(MT)(GS)(CH)(ORF1)(ORF2)(RH)'(D_x = C)
(WBC = 0)'(CC21)'(t) : [0 => CH]

(MT)(GS)(CH)(ORF1)(ORF2) [(RH)'(D_x = C)
(WBC = 0) + BD] (t) : [0 => DX] [0 => D_x]

(MT)(GS)(CH)(ORF1)(ORF2)[(RH)'(D_x = C)
{(WBC = 0) + CC21} + BD] (t) : [0 => GS]

(MT)(GS)(CH)(ORF1)(ORF2)'(D_x = ABD)
[(WBC = 0) + (RAN + SKP + WW + RWD)] (t) : [0 => D_x]

(MT)(GS)(CH)(ORF1)(ORF2)'(D_x = ABD)
[(WBC = 0) + CC21 + RAN + SKP + WW +
RWD] (t) : [0 => RH]

(MT)(GS)(CH)(ORF1)(ORF2)'(D_x = ABD)
[(WBC = 0)'(RRV + BSP)] (t) : [1 => CF] [0 => D_x]

(MT)(GS)(CH)(ORF1)(ORF2)'(D_x = ABD)
[(WBC = 0)'(CC21)'(RRV + BSP)] (t) : [0 => CH] [0 => ORF1]

(MT)(GS)(CH)(ORF1)(ORF2)'(RH)'(D_x = C)
(WBC = 0)'(t) : [0 => DX] [0 => D_x] [0 => ORF1]
[1 => CF]

(MT)(GS)(CH)(ORF1)(ORF2)'(RH)'(D_x = C)
(WBC = 0)'(CC21)'(t) : [0 => CH]

(MT)(GS)(CH)(ORF1)(ORF2)'(RH)'(D_x = C)
(WBC = 0)(t) : [0 => DX] [0 => D_x]

(MT)(GS)(CH)(ORF1)(ORF2)'(RH)'(D_x = C)
[(WBC = 0) + CC21] (t) : [0 => GS]

$(MT)(GS)(CH)(ORF1)'(ORF2)(D_x = A)(t) \quad : \quad [0 \Rightarrow WAR]$

$(MT)(GS)(CH)(ORF1)'(ORF_2)(D_x = AAB)$

$\quad [(WBC = 0) + (RRV + BSP + RWD)]\,(t) \quad : \quad [0 \Rightarrow RH][0 \Rightarrow D_x]$

$(MT)(GS)(CH)(ORF1)'(ORF2)(D_x = AAB)$

$\quad [(WBC = 0)'(WAN)]\,(t) \quad : \quad [0 \Rightarrow CH]\ [0 \Rightarrow D_x]\ [1 \Rightarrow CF]$

$(MT)(GS)(CH)(ORF1)'(ORF2)(D_x = AAB)$

$\quad [(WBC = 0)'(RAN + SKP + WWA)]\,(t) \quad : \quad [0 \Rightarrow WCF]\ [0 \Rightarrow ORF2]$

$\qquad [0 \Rightarrow CH\ \ [0 \Rightarrow D_x]\ [1 \Rightarrow CF]$

$(MT)(GS)(CH)(ORF1)'(ORF2)(RH)'$

$\quad (D_x = C)(WBC = 0)'(t) \quad : \quad [0 \Rightarrow CH]\ \ 0 \Rightarrow DX]\ \ 1 \Rightarrow CF]$

$\qquad [0 \Rightarrow D_x]\ [0 \Rightarrow ORF2]\ [0 \Rightarrow WCF]$

$(MT)(GS)(CH)(ORF1)'(ORF2)(RH)'$

$\quad (D_x = C)(WBC = 0)(t) \quad : \quad [0 \Rightarrow GS\ \ [0 \Rightarrow DX]\ [0 \Rightarrow D_x]$

$\qquad [0 \Rightarrow WCF]$

MAGNETIC TAPE CONTROL-WRITE

1)	SAME AS READ	
2)	(MT)(GS)(CH)'(RH)'(WAN)(RWB)(t)	: [1 => WCF] [0 => WAR]
	(WCF)(GS)(SCL)(RWB)	: [TROTAR = 1]
	(GS)(RH)(RV)'	: [BUS10 = 1]
	(GS)(WCF)	: [BUS12 = 1]
3)	SAME AS READ	
4)	(WCF)(BSR = 0)'(BXR = 0)(t)	: [BSR => BXR] [0 => BSR]
	(MT)(WCF)(BXR = 0)'(TAR = 0)(t)	: [BXR => TAR] [0 => BXR]
	(MT)(GS)(CH)'(PBM1)'(PBM2)'(WAN)	: [QB = 1]
	(QB)(TBLSD)(WCK)(p)	: [BCB => TAR]
	(QB)(Dx = CD)(ORF1)'(ORF2)'(t)	: [0 => Dx] [1 => PBM1]
	(SCL)'(QB)(Dx = CD)(ORF1)'(ORF2)'(TARn)(IWR)'(t)	: [WARn' => WAR]
	(SCL)(QB)(Dx = CD)(ORF1)'(ORF2)'(IWR)'	: [BUS24 = 1]
	(QB)(Dx = DEE)(ORF1)'(ORF2)(t)	: [0 => ORF2] [0 => Dx] [1 => PBM1]
	(SCL)'(QB)(Dx = DEE)(ORF1)'(ORF2)(TARn)(IWR)'(t)	: [WARn' => WAR]
	(SCL)(QB)(Dx = DEE)(ORF1)'(ORF2)(IWR)'	: [BUS 24 = 1]
	(QB)(Dx = DEE)(ORF1)(ORF2)(t)	: [0 => ORF1] [0 => ORF2] [0 => Dx] [1 => PBM1]
	(SCL)'(QB)(Dx = DEE)(ORF1)(ORF2)(TARn)(t)(IWR)'	: [WARn' => WAR]
	(SCL)(QB)(Dx = DEE)(ORF1)(ORF2)(IWR)'	: [BUS24 = 1]

(MT)(GS)(CH)'(PBM1)(PBM2)'(WAN)(Dx = 24μs)(t) : [1 => PBM2] [0 => TAR]

(SCL)'(MT)(GS)(CH)'(PBM1)(PBM2)'(WAN)(Dx = 24μs)(TARn)(IWR)'(t) : [WARn' => WAR]

(SCL)(MT)(GS)(CH)'(PBM1)(PBM2)'(WAN)(Dx = 24μs)(IWR)' : [BUS24 = 1]

(MT)(GS)(CH)'(PBM1)(PBM2)(WAN)(Dx = B)(t) : [0 => Dx] [0 => TAR]

(SCL)'(MT)(GS)(CH)'(PBM1)(PBM2)(WAN)(Dx = B)(TARn)(IWR)'(t) : [WARn' => WAR]

(SCL)(MT)(GS)(CH)'(PBM1)(PBM2)(WAN)(Dx = B)(IWR)' : [BUS24 = 1]

5) (MT)(WCF)(PBM1)(PBM2)(Dx = 24μs)(TBLE)'(t) : [0 => TAR] [0 => Dx]

(MT)(WCF)(PBM1)(PBM2)(D$_x$ = 22μsec)(WCK)(TBLED)(EFK)'(t) : [ECB => TAR]

(MT)(WCF)(PBM1)(PBM2)(D$_x$ = 22μsec)(TBLED)(EFK)'(t) : [EOF => TAR]

(MT)(WCF)(PBM1)(PBM2)(Dx = 24µs)(TBLE)(t) : [0 => PBM1] [0 => Dx]

(SCL)'(MT)(WCF)(PBM1)(PBM2)(Dx = 24µs)(TARn)(t)(IWR)' : [WARn' => WAR]

(SCL)(MT)(WCF)(PBM1)(PBM2)(Dx = 24µs)(IWR)' : [BUS24 = 1]

(MT)(WCF)(PBM1)'(PBM2)(Dx = 24µs)(t) : [0 => PBM2] [0 => Dx]
[0 => TAR]

(SCL)'(MT)(WCF)(PBM1)'(PBM2)(Dx = 24µs)(TARn)(t)(IWR)' : [WARn' => WAR]

(SCL)(MT)(WCF)(PBM1)'(PBM2)(Dx = 24µs)(IWR)' : [BUS24 = 1]

(BUS24) [(RDY)' + (TAR = 0)] (WCF) : [1 => TRE]

6) NO ACTION

7) SAME AS READ

8) SAME AS READ

CONTROL; PHOTO READER, PUNCH, FLEXOWRITER-READ

1) (ID)(GS)(RWB) : [BUS_{10} = 1]

(GS)(CH)'(RWB)'(DX)'(t) : [1 => DX]

(RWB)'($D_{x1-4} \geq 12\mu s$)(t) : [0 => DX] [0 => D_x]

(ID)(RWB)(t) : [1 => RCF]

2) (RCF)(BUS24)'(STB)(IGT)(P) : [0 => STB]

(RCF)(ID)(IGT) [(SCL)'($\sum_{1}^{8}$ BUS_n)' + (SCL)(STB)'](t) : [0 => IGT]

(RWB)(IGT)'(RCF)(SCL)(GS)(TAR = 0)(BXR = 0)
(BSR = 0)(TH1) [(ID) + (BD)(MM)'] : [BUS23 = 1]

(RCF)(SCL)(STB)'(BUS24) + (SCL)'(RCF)(ID)
(BUS_{13})(IGT)'(t) : [1 => STB]

(RCF)(STB)(t) : [1 => IGT]

(RCF)(IGT)' [(BUS24) + (SCL)'] : [BUS_n => TAR_n]

(RCF)(ID)(PBM1)'(PBM2)' [(SCL)'(STB) +
(SCL)(IGT)'(STB)] (t) : [1 = PBM1]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0)(p) : [1 => IGT]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0)(t) : [1 => PBM2]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0)(TAR_7)'
(TAR_6)(t) : [1 => TAR_7]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0)(TAR$_7$)(TAR$_6$)(STP)(t) : [TBLE => TAR] [1 => EFK]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0)(TAR$_7$)(TAR$_6$)(STP)(IS)'(t) : [TAR => BXR]

[(TAR$_6$)(TAR$_7$)(STP)'(DLT)' + (TAR$_6$)'(TAR$_7$)'(TAR = 0)'](RAN) : [QW9 = 1]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0)(QW9)(t) : [TAR => BXR][TBLE => TAR]

(RCF)(ID)(PBM1)(PBM2)'[(TAR = 0)'(TAR$_6$)'(TAR$_7$)' + (TAR$_6$)(TAR$_7$)(STP)'](t)(QW9)' : [0 => TAR]

(RCF)(ID)(PBM1)(PBM2)'(BXR = 0){(TAR$_7$) [(TAR$_6$)' + (TAR$_6$)(IS)'(STP)] + (QW9)}(p) : [TAR$_8$' => TAR$_8$]

3) (RCF)(ID)(PBM1)(PBM2)(TAR = 0)'(BXR = 0)(t) : [TAR => BXR][0 => TAR]

(RCF)(ID)(PBM1)(PBM2) [(BUS$_{13}$)'(SCL)' + (SCL)](BXR = 0)(t) : [0 => PBM1]

(RCF)(ID)(PBM1)'(PBM2)(t) : [0 => PBM2]

(RCF)(BXR = 0)'(BSR = 0)(t) : [BXR => BSR][0 => BXR]

(RCF)(ID)(CH)(ISG)'(WCK) : [CFC - C = 1]

(RCF)(ID)(CH)(ISG)'(EFK) : [CEF = 1][CEF - C = 1]

4) (ID)(CH)(GS)(t) : [0 => GS]

Note: Bus 10 contains Go level.
Bus 24 contains Strobe level.
Bus 23 contains Ready level.
Bus 13 contains Sync. level.

CONTROL; PUNCH, FLEXOWRITER-WRITE

1) $(OD)(BSR = 0)'(BXR = 0)(t) \quad : [BSR \Rightarrow BXR]\,[0 \Rightarrow BSR]$

2) $(OD)(PBM1)'(PBM2)'(BXR = 0)'(TAR = 0)(t) \quad : [BSR \Rightarrow TAR]\,[0 \Rightarrow BXR]$

$(OD)(PBM1)'(PBM2)'(TAR = 0)'\left[(BUS_{12})(BUS_{13})'(SCL)' + (SCL)\right](t) \quad : [1 \Rightarrow PBM1]$

3) $(MM)'(OD)(PBM1)(PBM2)'(IS)(TAR_6)(t)(WCK)' \quad : [0 \Rightarrow TAR_7]$

$(WCK)'(MM)'(OD)(PBM1)(PBM2)'(IS)'\left[(TAR_{1-6} = 0) + (TAR_6)(STP)'\right](t) \quad : [0 \Rightarrow TAR_7]$

$(OD)(PBM1)(PBM2)'(WCK)(TAR_6)'(t) \quad : [0 \Rightarrow TAR_7]$

$(OD)(PBM1)(PBM2)'(t) \quad : [1 \Rightarrow PBM2]$

$(MM)'(OD)(PBM1)(PBM2)'\left[(IS)(TAR_6)' + (IS)'\left\{(TAR_6)(STP) + (TAR_6)'(TAR_{1-6} = 0)'\right\}\right](p)(WCK)' \quad : [TAR_8' \Rightarrow TAR_8]$

$(OD)(PBM1)(PBM2)'(TAR_{1-6} = 0)(IS)' \quad : [0 = TAR_8]$

4) $(OD)\left[(PBM2)(SCL)' + (SCL)(GS)(RWB)\right] \quad : [TROTAR = 1]\,[BUS_9 = 1]$

$(OD)(PBM1)(PBM2)(STB)'(BUS23)(t) \quad : [1 \Rightarrow STB]$

$(OD)(STB)(PBM2) \quad : [BUS24 = 1]$

5) $(OD)(PBM1)(PBM2)\left[(BUS_{12})'(BUS_{13})(SCL)' + (SCL)(STB)\right](t) \quad : [0 \Rightarrow PBM1]$

$(OD)(PBM2)(STB)(SCL)(t) \quad : [0 \Rightarrow STB]$

$(OD)(PBM1)'(PBM2)(TAR = 0)'(t) \quad : [0 \Rightarrow TAR]$

$(OD)(PBM1)'(PBM2)\left[(BUS_{12})'(BUS_{13}) + SCL\right](t) \quad : [0 \Rightarrow PBM2]$

6) $(OD)(GS)(CH)(BSR = 0)(TAR = 0)(PBM1)'(PBM2)'(t) \quad : [0 \Rightarrow GS]$

$(MM)(GS)(WCK) \quad : [BUS_{13} = 1]$

$(MM)(GS)(EFK)(WW) \quad : [BUS_{15} = 1]$

Note: Bus 9 contains Write Out level.
Bus 23 contains Ready level.
Bus 24 contains Strobe level.
Bus 12 and Bus 13 contain Sync. levels.

LOGICAL MECHANIZATION: CENTRAL PROCESSOR

The mechanization presented in this paragraph consists of the MOBIDIC basic cycle plus the mechanization of all instructions other than the input-output instructions.

BASIC CYCLE

1. No Action

2. $(\lambda)'(TF2)$: $\left[WR^{n} = 1\right]$

   $(\lambda)'(I^{3} = 0)(p2)$: $\left[0 \Rightarrow RP\right]$

3. No Action

4. $(\lambda)'(MLY)'(t4)$: $\left[PC \Rightarrow MA^{n}\right]$

   $(\lambda)'(MLY)'(TF4)$: $\left[RE^{n} \Rightarrow 1\right]$

5. No Action

6. $(\lambda)'(MI)'(RP)'(TP)'(t6)$: $\left[MO^{n}{}_{31-37} \Rightarrow IR\right]$

   $\left[MO^{n}{}_{28-30} \Rightarrow G\right]$

   $\left[MO^{n}{}_{16-27} \Rightarrow X\right]$

   $\left[MO^{n}{}_{1-15} \Rightarrow AR\right]$

   $(\lambda)'(RP)(TP)'(t6)$: $\left[(I^{3} - 1) \Rightarrow I^{3}\right]$

   $(\lambda)'(TF6)$: $\left[WR^{n} \Rightarrow 1\right]$

7. $(\lambda)'(RP)'(TP)'\Big[\{(TRA)\ \big[(TS)+(TU)(X_{16})'\big]+(\sum PX)\}\ (SPI)'\Big]'(p7)$: $[PC+1 \Rightarrow PC]$

$(\lambda)'(RP)'(TRA)\ \big[(TS)+(TU)(X_{16})'\big]\ (SPI)'(t7)$: $[1 \Rightarrow TOT]$

$(\lambda)'(RP)'\Big[\{(TRA)\ \big[(TS)+(TU)(X_{16})'\big]+(\sum PX)\}\ (SPI)'\Big]\ (p7)$: $[0 \Rightarrow X_{17-16}]$

$(\lambda)'(RP)'\Big[\{(TRA)\ \big[(TS)+(TU)(X_{16})'\big]+(\sum PX)\}\ (SPI)'\Big]\ (t7)$: $[TRU \Rightarrow IR]\ [1 \Rightarrow SPI]$

$[1 \Rightarrow SPIX]$

$(\lambda)'(IO)'(RP)(TP)'(XP)'(t7)$: $\big[(AR+I^4) \Rightarrow AR\big]$

$(\lambda)'(IO)'(RP)'(TP)'(F)'(t7)$: $\big[(AR+I^{\gamma}) \Rightarrow AR\big]$

$(\lambda)'(\sum BB^i)(t7)$: $[1 \Rightarrow \lambda][1 \Rightarrow \phi][1 \Rightarrow SGA^*]$

$(\lambda)'(TP+HT)(\sum CNV^i)(t7)$: $[1 \Rightarrow \lambda]$

8. $(\lambda)(\sum BB^i)(t8)$: $[1 \Rightarrow \phi]\ [1 \Rightarrow SGA^*]$

$(\lambda)'(NM)'(TF8)$: $[RE^n \Rightarrow 1]$

$(\lambda)'(p8)$: $[0 \Rightarrow OF][0 \Rightarrow T]$

$(\lambda)'\big[TP'+PR2\big](p8)$: $[IR \Rightarrow D]$

$(\lambda)'(TP)\big[PR0+(PR3)(ONC)\big](p8)$: $[HLT \Rightarrow D]$

$(\lambda)'(NM)'(t8)$: $[AR \Rightarrow MA^n]$

$(\lambda)'(SC)'(t8)$: $[AR \Rightarrow T]$

$(\lambda)'(TP)'(NM)(t8)$: $[1 \Rightarrow \lambda]$

CLEAR AND ADD (16μs)

(D = 10) : [MRA = 1][CLA = 1]

2. (MRA)(t2) : $[\mathrm{MOR}^n \Rightarrow A]$

ADD (16μs)

(D = 12) : [ADD = 1][AOS = 1][MRB = 1]

2. (MRB)(t2) : $[\mathrm{MOR}^n \Rightarrow B]$

3. $(\mathrm{ADD})(A_{sn} \oplus B_{sn})'(p3)$: $[0 \Rightarrow AS]$

   $(\mathrm{ADD})(A_{sn} \oplus B_{sn})(p3)$: $[1 \Rightarrow AS][B' \Rightarrow B]$

   (AOS)(t3) : $[1 \Rightarrow AOS^*]$

4. (AOS)(AS)(OF)'(p4) : $[0 \Rightarrow B][A' \Rightarrow A][A'_{sn} \Rightarrow A_{sn}]$

5. $(\mathrm{AOS})(X_{18})'(p5)$: $[0 \Rightarrow OA]$

   $(\mathrm{AOS})(\mathrm{AS})'(\mathrm{OF})(X_{16}' + X_{17}')(t5)$: $[1 \Rightarrow OA]$

   $(\mathrm{KEW})'(\mathrm{AOS})(\mathrm{AS})'(\mathrm{OF})(X_{16})'(t5)$: $[1 \Rightarrow TP]$

   (AOS)(AS)(OF)'(t5) : $[1 \Rightarrow AOS^*]$

ADD MAGNITUDE (16μs)

(D = 13) : [ADM = 1][AOS = 1][MRB = 1]

2. (MRB)(t2) : $[\mathrm{MOR}^n \Rightarrow B]$

3. $(\mathrm{ADM})(A_{sn})'(p3)$: $[0 \Rightarrow AS]$

   $(\mathrm{ADM})(A_{sn})(p3)$: $[1 \Rightarrow AS][B' \Rightarrow B]$

   (AOS)(t3) : $[1 \Rightarrow AOS^*]$

4. Same as ADD

5. Same as ADD

CLEAR ADD MAGNITUDE (16μs)

(D = 11) : [CAM = 1][MRA = 1][CLA = 1]

2. (MRA) (t2) : [MOR^n => A]

3. (CAM) (t3) : [0 · => A$_{sn}$]

CLEAR AND SUBTRACT (16μs)

(D = 14) : [CLS = 1][MRA = 1]

2. (MRA) (t2) : [MOR^n => A]

3. (CLS) (p3) : [A'$_{sn}$ => A$_{sn}$]

CLEAR AND SUBTRACT MAGNITUDE (16μs)

(D = 15) : [CSM = 1][MRA = 1]

2. (MRA) (t2) : [MOR^n => A]

3. (CSM) (t3) : [1 => A$_{sn}$]

SUBTRACT (16μs)

(D = 16) : [SUB = 1] [AOS = 1] [MRB = 1]

2. (MRB) (t2) : [MOR^n => B]

3. (SUB) $(A_{sn} \oplus B_{sn})'$ (p3) : [1 => AS][B' => B]

(SUB) $(A_{sn} \oplus B_{sn})$ (p3) : [0 => AS]

(AOS) (t3) : [1 => AOS*]

4. Same as ADD

5. Same as ADD

SUBTRACT MAGNITUDE (16μs)

(D = 17) : [SBM = 1] [AOS = 1][MRB = 1]

2. (MRB) (t2) : [MOR^n => B]

3. $(SBM)(A_{sn})'(p3)$: $[1 \Rightarrow AS][B' \Rightarrow B]$

$(SBM)(A_{sn})(p3)$: $[0 \Rightarrow AS]$

$(AOS)(t3)$: $[1 \Rightarrow AOS^{*}]$

4. Same as ADD

5. Same as ADD

MULTIPLE (86μs)

$(D = 20)$: $[MLY = 1][MRB = 1]$

2. $(MRB)(t2)$: $[MOR^{n} \Rightarrow B]$

3. $(MLY)(t3)$: $[A \Rightarrow Q][0 \Rightarrow AS]$

4. $(MLY)(p4)$: $[0 \Rightarrow A]$

$(MLY)(t4)$: $[1 \Rightarrow \lambda][(T + 1) \Rightarrow T]$

$(MLY)(Q_1)(t4)$: $[1 \Rightarrow AOS^{*}]$

5. $(MLY)(p5)$: $[1 \Rightarrow SRA^{*}][1 \Rightarrow SRQ^{*}][A_1 \Rightarrow Q_{36}]$

$(MLY)(T = 36)(p5)$: $[0 \Rightarrow \lambda]$

$(MLY)(t5)$: $[(T + 1) \Rightarrow T]$

$(MLY)(\lambda)(Q_1)(t5)$: $[1 \Rightarrow AOS^{*}]$

$(MLY)\sum(BB^{i})(T \leq 30)(t5)$: $[1 \Rightarrow \phi][1 \Rightarrow SGA^{*}]$

$(MLY)(T = 35)(t5)$: $[PC \Rightarrow MA^{n}]$

$(MLY)(T = 35)(TF5)$: $[RE^{n} = 1]$

6. $(MLY)(B_{sn})(p6)$: $[(A_{sn})' \Rightarrow A_{sn}]$

$(MLY)(t6)$: $[A_{sn} \Rightarrow Q_{sn}]$

MULTIPLY AND ROUND (86µs)

(D = 21) : [MLR = 1][MLY = 1][MRB = 1]

2. Same as Multiply

3. Same as Multiply

4. Same as Multiply

5. Same as Multiply

6. (MLY)(B_{sn})(p6) : $[A'_{sn} \Rightarrow A_{sn}]$

(MLR)(Q_{36})(p6) : [0 => B][1 => AS]

(MLR)(Q_{36})(t6) : [1 => AOS*]

(MLY)(t6) : $[A_{sn} \Rightarrow Q_{sn}]$

DIVIDE (88µs; 18µs IF OVERFLOW)

(D = 22) : [DVD = 1][MRB = 1]

2. (MRB)(t2) : $[MOR^n \Rightarrow B]$

(DVD)(t2) : $[0 \Rightarrow AS][0 \Rightarrow Q_{sn}]$

3. (DVD)$(\lambda)'$(p3) : $[1 \Rightarrow \lambda]$

(DVD)$(Q_{sn})'$(p3) : $[(AS \oplus OF)' \Rightarrow AS]$

(DVD)$(Q_{sn})'(OF)'$ (p3) : [B' => B]

(DVD)$(\lambda)(Q_{sn})'$(p3) : [1 => SLA*][1 => SLQ*]

$[(AS \oplus OF)' \Rightarrow Q_1]$

(DVD(SLA*) : $[A_{36} \Rightarrow OF]$

(DVD)(t3) : [1 => AOS*][(T + 1) => T]

(DVD)(T = 0)(C_{36})(t3) : $[1 \Rightarrow Q_{sn}]$

(DVD)(Q_{sn})(p3) : $[0 \Rightarrow \lambda]$[B' => B][0 => AS]

$(DVD)(Q_{sn})'(T = 0)'\sum(BB^j)(T \leq 30)(t3)$: $\left[1 => \phi\right]\left[1 => SGA^*\right]$

$(DVD)(T = 36)p3)$: $\left[0 => \lambda\right]$

4. $(DVD)(Q_{sn})'(p4)$: $\left[1 => SLQ^*\right]\left[(AS \oplus OF)' => Q_1\right]$

5. $(DVD)(X_{18})'(p5)$: $\left[0 => OA\right]$

$(DVD)(Q_{sn})(X_{16}' + X_{17}')(t5)$: $\left[1 => OA\right]$

$(DVD)(KEW)'(Q_{sn})(X_{16})'(t5)$: $\left[1 => TP\right]$

$'DVD)(Q_{sn})'(AS)(p5)$: $\left[B' => B\right]\left[0 => AS\right]$

$(DVD)(Q_{sn})'(Q_1)'(t5)$: $\left[1 => AOS^*\right]$

6. $(DVD)(Q_{sn})'(t6)$: $\left[(A_{sn} \oplus B_{sn}) => Q_{sn}\right]$

$(DVD)(Q_{sn})(t6)$: $\left[A_{sn} => Q_{sn}\right]$

DIVIDE LONG (88μs; 18μs IF OVERFLOW

$(D = 23)$: $\left[DVL = 1\right]\left[DVD = 1\right]\left[MRB = 1\right]$

2. Same as Divide

3. $(DVL)(\lambda)(Q_{sn})'(p3)$: $\left[Q_{36} => A_1\right]$

Otherwise same as Divide

4-6 Same as Divide

ADD BETA (26μs)

IR = 24 : $\left[F = 1\right]$

$(D = 24)$: $\left[ADB = 1\right]\left[ASB = 1\right]$

1. $(ASB)(T = 0)(pl)$: $\left[1 => \lambda\right]$

$(ASB)(tl)$: $\left[(T + 1) => T\right]$

$(ASB)(T = 0)(tl)$: $\left[A => Q\right]$

$(ASB)(T = 1)(t1) \quad : [X => B_{1-12}]$

$(ASB)(T = 2)(t1) \quad : [MOR^{n} => A]$

$(ADB)(A_{sn} \oplus B_{sn})'(T = 3)(p1) \quad : [0 => AS]$

$(ADB)(A_{sn} \oplus B_{sn})(T = 3)(p1) \quad : [1 => AS][B' => B]$

$(ASB)(T = 3)(t1) \quad : [1 => AOS^{*}]$

$(ASB)(AS)(OF)'(T = 4)(p1) \quad : [0 => B][A' => A][A'_{sn} => A_{sn}]$

$(ASB)(AS)(OF)'(T = 4)(t1) \quad : [1 => AOS^{*}]$

$(ASB)(T = 4)(t1) \quad : [0 => MO^{n}]$

$(ASB)(T = 5)(p1) \quad : [0 => \lambda]$

$(ASB)(T = 5)(t1) \quad : [A => MOR^{n}][A_{1-12} => I^{\gamma}]$

2. No Action

3. $(ASB)(\sum BB^{i})(t3) \quad : [1 => \lambda]\ [1 => \phi][1 => SGA^{*}]$

4. $(ASB)(\sum BB^{i})'(t4) \quad : [0 => \lambda]$

SUBTRACT BETA (26μs)

$(IR = 25) \quad : [F = 1]$

1. $(D = 25) \quad : [SBB = 1]\ [ASB = 1]$

Same as ADB, except replace (ADB) etc. with:

$(SBB)(A_{sn} \oplus B_{sn})'(T = 3)(p1) \quad : [1 => AS][B' => B]$

$(SBB)(A_{sn} \oplus B_{sn})(T = 3)(p1) \quad : [0 => AS]$

2. Same as ADB

3. Same as ADB

4. Same as ADB

SHIFT LEFT ($a \leq 9$: 16μs; $a > 9$: 8 + a - a (mod 2) μs)

	(IR = 30)	: [SC = 1]
	(D = 30)	: [SHL = 1] [SH = 1] [SLC = 1]
1-5	(SH) (T = 0)' (H)' (TF8)' (p + t)	: [(T - 1) => T]
	(SLC) (T = 0)' (H)' (TF8)' (p + t)	: [1 => SLA*]
	(SH) (A_{36}) (SLA*)	: [1 => OF]
3.	(SHL) (T $\leq$ 4)' (t3)	: [1 => λ]
4.	(SH) (λ) ($\sum BB^i$) (t4)	: [1 => ϕ] [1 => SGA*]
	(SHL) (T $\leq$ 4) ($\sum BB^i$)' (t4)	: [0 => λ]
5.	(SHL) (X_{18})' (p5)	: [0 => OA]
	(SHL) (OF) ($X_{16}' + X_{17}'$) (t5)	: [1 => OA]
	(SHL) (KEW)' (OF) (X_{16})' (t5)	: [1 => TP]

LONG SHIFT LEFT ($a \leq 9$: 16μs; $a > 9$: 8 + a - a (mod 2) μs)

	(IR = 31)	: [SC = 1]
	(D = 31)	: [SLL = 1] [SH = 1] [SLC = 1]
		: [SHL = 1]
1-5	(SH) (T = 0)' (H)' (TF8)' (p + t)	: [(T - 1) => T]
	(SLC) (T = 0)' (H)' (TF8)' (p + t)	: [1 => SLA*]
	(SLL) (T = 0)' (H)' (TF8)' (p + t)	: [1 => SLQ*] [Q_{36} => A_1]
	(SH) (A_{36}) (SLA*)	: [1 => OF]

3 5 Same as Shift Left

SHIFT RIGHT ($\alpha \leq 14$: 16μs; $\alpha > 14$: $2 + \alpha + \alpha$ (mod 2) μs)

(IR = 32) : [SC = 1]

(D = 32) : [SHR = 1][SH = 1]

1-7 (SH)(T = 0)'(H)'(TF8)'(p + t) : [(T - 1) => T]

(SHR)(T = 0)'(H)'(TF8)'(p + t) : [1 => SRA*]

3. (SH)(SHL)'($T \leq 9$)'(t3) : [1 => λ]

4. (SH)(λ)($\sum BB^i$)(t4) : [1 => φ][1 => SGA*]

(SH)(SHL)'($T \leq 9$)($\sum BB^i$)'(t4) : [0 => λ]

LONG SHIFT RIGHT ($\alpha \leq 14$: 16μs; $\alpha > 14$: $2 + \alpha + \alpha$ (mod 2) μs)

(IR = 33) : [SC = 1]

(D = 33) : [SRL = 1][SHR = 1][SH = 1]

1-7 (SH)(T = 0)'(H)'(TF8)'(p + t) : [(T - 1) => T]

(SHR)(T = 0)'(H)'(TF8)'(p + t) : [1 => SRA*]

(SRL)(T = 0)'(H)'(TF8)'(p + t) : [1 => SRQ*][A_1 => Q_{36}]

3-4 Same as Shift Right

CYCLE SHORT ($\alpha \leq 14$: 16μs; $\alpha > 14$: $2 + \alpha +$ (mod 2) μs)

(IR = 34) : [SC = 1]

(D = 34) : [CYS = 1][SH = 1][SLC = 1]

1-7 (SH)(T = 0)'(H)'(TF8)'(p + t) : [(T - 1) => T]

(SLC)(T = 0)'(H)'(TF8)'(p + t) : [1 => SLA*]

(CYS)(SLA*) : [A_{36} => A_1]

3-4 Same as Shift Right

CYCLE LONG ($a \leq 14$: 16μs; $a > 14$: $2 + a + a$ (mod 2) μs)

(IR = 35) : [SC = 1]

(D = 35) : [CYL = 1] [SH = 1] [SLC = 1]

1-7 (SH)(T = 0)'(H)'(TF8)'(p + t) : [(T - 1) => T]

(SLC)(T = 0)'(H)'(TF8)'(p + t) : [1 => SLA*]

(CYL)(T = 0)'(H)'(TF8)'(p + t) : [1 => SLQ*] [A_{36} => A_{sn}]

[A_{sn} => Q_1] [Q_{36} => Q_{sn}]

[Q_{sn} => A_1]

3-4 Same as Shift Right

NORMALIZE (18 + n-n (mod 2) μs, n = number of shifts performed)

(IR = 37) : [NM = 1]

(D = 37) : [NRM = 1]

1. (NRM)(A_{36})'(T = 36)'(p1'+ t1) : [1 => SLA*] [(T + 1) => T]

(NRM)(λ)($\sum BB^i$)(t1) : [1 => φ] [1 => SGA*]

(NRM)(λ) [A_{36} + (T = 36)] ($\sum BB^i$)'(t1) : [0 => λ]

(NRM)(λ)'(t1) : [AR => MA^n] [1 => λ]

(NRM)(λ)'(TF1) : [RE^n = 1]

2. (NRM)(λ)(t2) : [T => MOR^n] [0 => λ]

STORE (16μs)

(D = 50) : [STR = 1]

1. (STR)(t1) : [A => MOR^n]

LOAD (18μs)

	(D = 51)	:	[LOD = 1][MRB = 1]
2.	(MRB)(t2)	:	[MOR^n => B]
	(LOD)(t2)	:	[1 = λ]
3.	(LOD)(λ)(t3)	:	[X => AR][0 => λ][1 => AR_{13}] [1 => AR_{14}][1 => AR_{15}]
	(LOD)(λ)' (t3)	:	[B => REG^n]

MOVE (26μs)

	(IR = 52)	:	[F = 1]
	(D = 52)	:	[MOV = 1][MRB = 1]
2.	(MRB)(t2)	:	[MOR^n => B]
	(MOV)(λ)'(p3)	:	[1 => λ]
	(MOV)(t3)	:	[(T + 1) =>T]
	(RP)(MOV)(T = 0)(RP)(t3)	:	[AR => Q]
	(MOV)(T = 1)(t3)	:	[G => AR_{13-15}][X => AR_{1-12}]
	(MOV)(T = 2)(t3)	:	[AR => MA^n]
	(MOV)(T = 2)(TF3)	:	[RE^n = 1]
	(MOV)(T = 3)(t3)	:	[B => MOR^n]
	(MOV)(T = 4)(t3)	:	[AR + I^2 => AR]
	(MOV)(T = 4)(TF3)	:	[WR^n = 1]
	(MOV)(T = 5)(p3)	:	[0 => λ]

(MOV)(T = 5)($\sum$BB^i)(t3) : [1 => λ][1 => ϕ][1 => SGA*]

(MOV)(T = 5)(t3) : [AR$_{1-12}$ => X][$\dot{\text{AR}}_{13-15}$ => G]

4. (MOV)($\sum$BB^i)'(t4) : [0 => λ]

5. (MOV)(t5) : [Q => AR]

REPEAT (16μs)

(D = 01) : [RPT => 1]

2. (RPT)(t2) : [X => I^4]

3. (RP)'(RPT)(t3) : [AR => I^3]

7. (RPT)(t7)(IO)' : [1 => RP]

TRANSFER UNCONDITIONAL (16μs)

(IR = 40) : [TU = 1]

(D = 40) : [TRU => 1]

2. (TRU)(SPIX)(X$_{16}$)'(t2) : [PC => B]

3. (TRU)(p3) : [0 => PC]

(TRU)(SPIX)' + (X$_{16}$)(t3) : [AR => PC]

4. (TRU)(TRA)(X$_{17}$)(X$_{16}$)(t4) : [0 => TRA]

(TRU)(CTMW)'(X$_{16}$)(X$_{17}$)'(t4) : [1 => TRA]

(TRU)(t4) : [0 => SPIX]

TRANSFER TO PCS (16μs)

(IR = 42) : [TS = 1]

(D = 42) : [TRS = 1]

3. (TRS)(t3) : [PCS => PC]

TRANSFER AND LOAD PCS (16μs)

(IR = 41) : [TS = 1] [F = 1]

(D = 41) : [TRL = 1]

2. (TRL)(t2) : [PC => PCS]

3. (TRL)(t3) : [AR => PC]

5. (TRL)(t5) : $[X => I^{\gamma}]$

TRANSFER ON NEGATIVE (16μs)

(IR = 46) : [TS = 1]

(D = 46) : [TRN = 1]

3. $(TRN)(A_{sn})(t3)$: [AR => PC]

TRANSFER ON POSITIVE (16μs)

(IR = 44) : [TS = 1]

(D = 44) : [TRP = 1]

3. $(TRP)(A_{sn})'(t3)$: [AR => PC]

TRANSFER ON ZERO (16μs)

(IR = 45) : [TS = 1]

(D = 45 : [TRZ = 1]

3. (TRZ)(ZT)(t3) : [AR => PC]

TRANSFER ON INDEX (22μs)

(IR = 43) : [TS = 1] [F = 1]

(D = 43) : [TRX = 1]

1. $(TRX)(I^{\gamma+1} = 0)'(t1)$: $[(I^{\gamma+1} - 1) => I^{\gamma+1}]$

```
2.  (TRX)(t2)                                   : [AR => B]
3.  (TRX)(T = 0)($I^{\gamma+1}$ = 0)'(p3)       : [1 => λ]
    (TRX)(t3)                                   : [(T + 1) => T]
    (TRX)(T = 0)($I^{\gamma+1}$ = 0)'(t3)       : [X => AR]
    (TRX)(T = 1)(t3)                            : [(AR + $1^{\gamma}$) => AR]
    (TRX)(T = 2)(t3)                            : [AR =>$I^{\gamma}$][0 => λ]
    (TRX)(λ)'($I^{\gamma+1}$ =0)'(t3)           : [B => PC]
      COMPARE (22μs)
    (D = 47)                                    : [TRC = 1][MRB = 1]
2.  (MRB)(t2)                                   : [$MOR^n$ => B]
    (TRC)(t2)                                   : [1 => λ]
3.  (TRC)(T = 0)($A_{sn} \oplus B_{sn}$)'(p3)   : [1 => AS][B' => B]
    (TRC)(T = 0)($A_{sn} \oplus B_{sn}$)(p3)    : [0 =>AS]
    (TRC)(t3)                                   : [(T + 1) => T]
    (TRC)(T = 0)(t3)                            : [A => Q]
    (TRC)(T = 0)(t3)                            : [1 => AOS*]
    (TRC)(T = 1)(AS)(OF)'(p3)                   : [0 => B][A' => A]
    (TRC)(T = 1)(AS)(OF)'(t3)                   : [1 => AOS*]
    (TRC)(T = 2)(ZT)(p3)                        : [0 => $A_{sn}$][(PC + 1) => PC]
    (TRC)(T = 2)($A_{sn}$ + ZT)(p3)             : [0 => RP]
    (TRC)(T = 2)(t3)                            : [0 = λ]
```

(TRC)(λ)'(t3) : [AR => B]

(TRC)(λ)'(RP)'(A_{sn})'(p3) : [(PC + 1) => PC]

5. (TRC)(t5) : [Q => A]

LOAD INDEX (16μs)

(IR = 53) : [F = 1]

(D = 53) : [LDX = 1]

2. (LDX)(t2) : [AR => $I^{\gamma+1}$]

3. (LDX)(t3) : [X => I^{γ}]

REPLACE ADDRESS (22μs)

(D = 54) : [RPA = 1][RPL = 1]

1. (RPL)(tl) : [A => B] [1 => λ]

2. (RPL)(t2) : [(T + 1) => T]

(RPL)(T = 0)(t2) : [MOR^n => A]

(RPA)(T = 1) : [RA = 1]

(RPL)(T = 1 (t2) : [B => A][0 => MO^n]

(RPL)(T = 2)(t2) : [A => MOR^n][0 => λ]

5. (RPL)(t5) : [B => A]

REPLACE THROUGH MASK (22μs)

(D = 55) : [MSK = 1][RPL = 1]

1. (RPL)(tl) : [A => B][1 => λ]

2. (RPL)(t2) : [(T + 1) => T]

(RPL) (T = 0) (t2) : [MOR^{n} => A]

(MSK) (T = 1) : [RM = 1]

(RPL) (T = 1) (t2) : [B => A] [0 => MO^{n}]

(RPL) (T = 2) (t2) : [A => MOR^{n}] [0 => λ]

5. (RPL) (t5) : [B => A]

LOGICAL MULTIPLY (16μs)

(D = 02) : [LGM = 1] [MRB = 1]

2. (MRB) (t2) : [MOR^{n} => B]

3. (LGM) (t3) : [1 => LGM^{*}]

HALT

(IR = 00) : [HT = 1]

(D = 00) : [HLT = 1]

3. (HLT) (t3) : [1 => H]

SENSE (16μs)

(IR = 05) : [TS = 1]

(D = 05) : [SEN = 1]

3. (SEN) (SNL^{β}) t3) : [AR => PC]

SENSE AND SET

(IR = 06) : [TS = 1]

(D = 06) : [SNS = 1]

3. (SNS) (SNL^{β})' (t3) : [AR => PC]

4. (SNS) (t4) : [1 => AFF^{β}]

SENSE AND RESET (16μs)

(IR = 07) : [TS = 1]

(D = 07 : [SNR = 1] [SEN = 1]

3. (SEN) (SNL$^{\beta}$) (t3) : [AR => PC]

4. (SNR) (t4) : [0 => AFF$^{\beta}$]

LOGICAL ADD (16μs)

(D = 03) : [LGA = 1] [MRB = 1]

2. (MRB) (t2) : [MOR^{n} => B]

3. (LGA) (t3) : [B => A]

Note: This is a transfer of B into A at t3 without clearing A at p3.

LOGICAL NEGATION (16μs)

(D = 04) : [LGN = 1] [MRA = 1]

2. (MRA) (t2) : [MOR^{n} => A]

4. (LGN) (p4) : [A' => A] [A'_{sn} => A_{sn}]

LOGICAL MECHANIZATION FOR CONSOLE FUNCTIONS

The console function mechanization list logically describes the automatic and manual switching operations of the console.

INITIATING SWITCHES

$$\mathrm{STW + COW + PRW + ROW + RIW + MIW} + \sum \mathrm{CMW}^n \qquad : \left[\sum \mathrm{W} = 1\right]$$

$$\mathrm{(STW)'\,(COW)'\,(PRW)'\,(ROW)'\,(RIW)'\,(MIW)'}\left(\sum \mathrm{CMW}^n\right) \qquad : \left[\sum \mathrm{W'} = 1\right]$$

SINGLE GATE CIRCUIT

$$\left(\sum \mathrm{W}\right)\mathrm{(H)(MR')(XT')(SG')(t)} \qquad : \left[1 \Rightarrow \mathrm{SG}\right]$$

$$\mathrm{(SG)(t)} \qquad : \left[1 \Rightarrow \mathrm{XT}\right] \left[0 \Rightarrow \mathrm{SG}\right]$$

$$\left(\sum \mathrm{W'}\right)\mathrm{(t)} \qquad : \left[0 \Rightarrow \mathrm{XT}\right]$$

SINGLE PULSE CIRCUIT

$$\mathrm{(XG')}\left(\sum \mathrm{W}\right)\mathrm{(SP')(SIP)(CLOCK)} \qquad : \left[1 \Rightarrow \mathrm{SP}\right]$$

$$\mathrm{(SP)(SIP)(CLOCK)} \qquad : \left[0 \Rightarrow \mathrm{SP}\right]$$

$$\mathrm{(SP)(CLOCK)} \qquad : \left[1 \Rightarrow \mathrm{XG}\right]$$

$$\left(\sum \mathrm{W'}\right)\mathrm{(CLOCK)} \qquad : \left[0 \Rightarrow \mathrm{XG}\right]$$

$$\mathrm{PRO' + SIP' + SP} \qquad : \left[\mathrm{SPL} = 1\right]$$

OPERATING MODES

A. Run; Start

(STW)(SG)(t) : [ASR => PC]

(STW)(SG)(te) : [0 => H]

B. One Operation: Start

Same as A. Plus

.(CM')(XP')(ONC)(t_5) : [1 => TP]

C. Single Pulse; Start

Same as A.

D. Run; Continue

(COW)(SG)(te) : [0 => H]

E. One Operation; Continue

Same as D. Plus

(CM')(XP')(ONC)(t_5) : [1 => TP]

F. Single Pulse; Continue

Same as D.

G. Run; Program Read-In

(SIP')(PRW)(SG)(t) : [1 => PR1] [1 => TP] [1 => CLRCIS^1]

(SIP')(PRW)(SG)(t_e) : [0 => H]

(PR1)(t_4) : [ROK, K, J, => CIS^1] j = 20, k = 2

(PR1)(t_5) : [1 => CNV^1] [1 => GS^1]

Included in in-out memory cycle

$(TC2)(\phi)(PR1)(t)$: $\left[BFR^{1} \Rightarrow IR, G, X, AR\right] \left[i \Rightarrow PR2\right]$

$(TC2)(\phi)(PR2)(t)$: $\left[BFR^{1}_{15-1} \Rightarrow PC\right]$

Included in basic cycle

$(\lambda')(TP' + PR2)(P8)$: $\left[IR \Rightarrow D\right]$

$(\lambda') \left[PRO + (PR3)(ONC)\right]$ $(TP)(P8)$: $\left[HLT \Rightarrow D\right]$

Included in in-out

$(CSL^{i})(t_2)$: $\left[1 \Rightarrow CLRCIS^{i}\right]$

$(CSL^{i})(t_3)$: $\left[IR \Rightarrow ISR^{i}\right]$ etc.

$(CSL^{i})(IMO^{i'})(t_5)$: $\left[1 \Rightarrow CNV^{i}\right] \left[1 \Rightarrow GS^{i}\right]$

$(\lambda')(PR2)(t_8)$: $\left[1 \Rightarrow PR3\right]$

$(\lambda')(PR3)(t_8)$: $\left[1 \Rightarrow PRO\right] \left[0 \Rightarrow IR\right]$

H. One Operation; Program Read-In

Same as G. Plus

$(\lambda')(PR3)(ONC')(t_8)$: $\left[0 \Rightarrow TP\right]$

I. Run; Manual Instruction

$(MIW)\ (SG)(t)$: $\left[1 \Rightarrow MI\right]$

$(MIW)(SG)(t_e)$: $\left[0 \Rightarrow H\right]$

Included in basic cycle

$(\lambda')(MI)(t_6)$: $\left[WSR_{37-31} => IR\right] \left[WSR_{30-28} => G\right] \left[WSR_{27-16} => X\right]$

$\left[WSR_{15-1} => AR\right]$

$(TP)(t_6)$: $\left[0 => MI\right]$

J. One Operation; Manual Instruction

Same as I. Plus

$(CM')(XP')(ONC)(t_5)$: $\left[1 => TP\right]$

K. Single Pulse; Manual Instruction

Same as I.

L. Run; Manual Read-In

$(RIW)(SG)(t)$: $\left[1 => RI\right] \left[1 => MR\right] \left[1 => TCX\right]$

$(MR)(TCX)(P)$: $\left[0 => AR\right]$

$(MR)(TCX)(t)$: $\left[ASR => AR\right] \left[0 => TCX\right]$

$(MR)(TC1)$: $\left[RE^n = 1\right]$

$(MR)(TC1)(t)$: $\left[AR => MA^n\right]$

$(MR)(TC2)(RI)(t)$: $\left[WSR => MOR^n\right]$

$(MR)(TC3)$: $\left[WR^n = 1\right]$

$(HAW)(TC4)$: $\left[0 => MR\right] \left[0 => RI\right]$

M. One Operation; Manual Read-In

Same as L. Plus

$(ONC)(TC4)(t)$: $\left[0 => RI\right] \left[0 => MR\right]$

N. Single Pulse; Manual Read-In

Same as L.

O. Run; Manual Read-Out

(ROW)(SG)(t) : [1 => MR] [1 => TCX]

(MR)(TCX)(P) : [0 => AR]

(MR)(TCX)(t) : [ASR => AR] [0 => TCX]

(MR)(TC1) : [RE^n => 1]

(MR)(TC1)(t) : [AR => MA^n]

(MR)(TC3) : [WR^n => 1]

(HAW)(TC4) : [0 => MR]

P. One Operation; Manual Read-Out

Same as O. Plus

(ONC)(TC4)(t) : [0 => MR]

Q. Single Pulse: Manual Read-Out

Same as O.

SPECIAL PURPOSE CONTROLS

A. Halt (TP - Stop Flip-flop)

1) Halt Switch

(HAW)(XP)' (t5) : [1 => TP]

(HAW)(TC4)(t) : [0 => MR] [0 => RI]

2) Alarm Halt

$$(\mathrm{KEW})'\ \left[\mathrm{CTP}^{i} + \sum \mathrm{MPE} + \mathrm{NXML} + \mathrm{NXI}\right] (\mathrm{XP})'(t5) : \left[1 \Rightarrow \mathrm{TP}\right]$$

3) Total TP Flip-flop logical Conditions (This is included here because the complete expression appears in no other mechanization)

$$(\mathrm{PRW})(\mathrm{SG})(t) + \Big[(\mathrm{ONC})(\mathrm{CM})' + \mathrm{HAW} + (\mathrm{RP})'(\mathrm{CM}) + (\mathrm{KEW})' \left\{\mathrm{CTP} + \sum \mathrm{MPE}\right.$$

$$\left. + \mathrm{NXML} + \mathrm{NXI}\right\} + (\mathrm{KEW})' \left\{X_{16}'\ (Q_{sn})(\mathrm{DVD}) + (\mathrm{OF}) \left[(\mathrm{AS})'(\mathrm{AOS}) + \mathrm{SLL} + \mathrm{SHL}\right]\right\}\Big]$$

$$(\mathrm{XP})'(t5) \quad : \left[1 \Rightarrow \mathrm{TP}\right]$$

$$(\mathrm{SG})(\mathrm{PRW})'(t) \quad : \left[0 \Rightarrow \mathrm{TP}\right] \left[1 \Rightarrow \mathrm{XP}\right]$$

$$(t3) \quad : \left[0 \Rightarrow \mathrm{XP}\right]$$

B. Emergency Halt

$$(\mathrm{EHW})(t) \quad : \left[\pi_{i=1}^{n} (0 \Rightarrow \mathrm{CNV}^{i})\right] \left[1 \Rightarrow \mathrm{PRO}\right] \left[1 \Rightarrow \mathrm{TP}\right]$$

C. Clear Memory

$$(\sum \mathrm{CMW}^{n})(\mathrm{SG})(t_{e}) \quad : \left[0 \Rightarrow \mathrm{H}\right]$$

$$(\sum \mathrm{CMW}^{n})(\mathrm{SG})(\mathrm{P}) \quad : \left[0 \Rightarrow \mathrm{A}\right] \left[0 \Rightarrow \mathrm{IR}\right] \left[0 \Rightarrow \mathrm{AR}\right]$$

$$: \left[0000 \Rightarrow \mathrm{I}^{3}\right]$$

$$(\sum \mathrm{CMW}^{n})(\mathrm{SG})(t) \quad : \left[1 \Rightarrow \mathrm{RP}\right] \left[1 \Rightarrow \mathrm{CM}\right] \left[\mathrm{STR} \Rightarrow \mathrm{IR}\right]$$

$$\left[0001 \Rightarrow \mathrm{I}^{4}\right]$$

$$(\sum \mathrm{CMW}^{n})(\mathrm{SG})(t) \quad : \left[n \Rightarrow \mathrm{AR}_{15\text{-}13}\right]$$

$$(\mathrm{RP})'(\mathrm{CM})(t5) \quad : \left[1 \Rightarrow \mathrm{TP}\right] \left[0 \Rightarrow \mathrm{CM}\right]$$

FLIP-FLOP CONTROLS

A. Sense Flip-Flop

1) NHC

(SNPW)(H)(t) : [1 => NHP]

(CNPW)(H)(t) : [0 => NHP]

2) ISN

(SISW)(H)(t) : [1 => ISN]

(CISW)(H)(t) : [0 => ISN]

3) TRA

(STMW)(H)(t) : [1 => TRA]

(CTMW)(H)(t) : [0 => TRA]

4-11) SFF^{1} - SFF^{16}

(S1W)(t) : [1 => SFF^{1}]

(C1W)(t) : [0 => SFF^{1}]

12) TPE

(SIPW)(H)(t) : [1 => TPE]

(CIPW)(H)(t) : [0 => TPE]

B. Alarm Flip-Flops

1) MPE^n

$(CLAW)(H)(P)$: $\left[0 \Rightarrow MPE^n\right]$

2) $IOA^i - DVA^i - IPE^i - ISE^i - TRE^i - IMO^i$ $\left[\text{All } CNV^i \text{ Alarm Flip-Flops}\right]$

$(CLAW)(H) + (SNR)(TF4)(X_{16-22}(IOA^i)$: $\left[CIA^i \Rightarrow 1\right]$

$(CIA^i) + (CCAW^i) + (CLR)\ (t)$: $\left[\text{Clear All } CNV^i \text{ Alarm F.Fs.}\right]$

3) OA

$(CLAW)(H)(t)$: $\left[0 \Rightarrow OA\right]$

4) NXM

$(CLAW)(H)(t)$: $\left[0 \Rightarrow NXM\right]$

5) NXI

$(CLAW)(H)(t)$: $\left[0 \Rightarrow NXI\right]$

BUS INDICATOR CIRCUIT

$(\sum CMW^{n'})(CLR')(H)(MR')$: $\left[BI = 1\right]$

$(BI)(BWR^n)$: $\left[TROREG^n = 1\right]$

SECTION VII

VAN LAYOUT

The layout and weight distribution of MOBIDIC 7A equipment in the two computer vans is shown on Figures 1-1 and 1-2. The equipment has been functionally grouped. Due consideration has been given to human engineering and weight distribution factors.

The arrangement of the equipment and the system interconnections and design are such that the equipment in van 1 can be operated as a separate computer, independent of the equipment in van 2. In addition, van 2 equipment can be operated as an independent off-line data processing system.

A status, or alarm, panel and associated equipment has been included in van 1 to allow the operator in that van to monitor and control the equipment in van 2. The status panel is mounted above the console. The height and angle at which the panel is mounted make it easy for the operator to see the various indicators and operate the controls. Indicator lights display the status of all equipment in van 2. Switches are included for reserving shared devices (those in-out devices capable of either on-line or off-line use), all of which are in van 2. The arrangement of the indicators and controls is shown on Figue 7-1. The indicators and their lens colors are listed on Table 7-1.

Two-position toggle switches are used to reserve the shared devices. Five such switches are provided: one for each of the two magnetic tape transports, one for each of the card reader-punches, and one for the line printer. Each switch has a RELEASE and a SELECT position. When the switch associated with a shared device is in the SELECT position, that device is reserved for on-line use: When the switch is in the RELEASE position, that device may be used off-line. Action of the on-line - off-line interlock provisions is described is Section II.

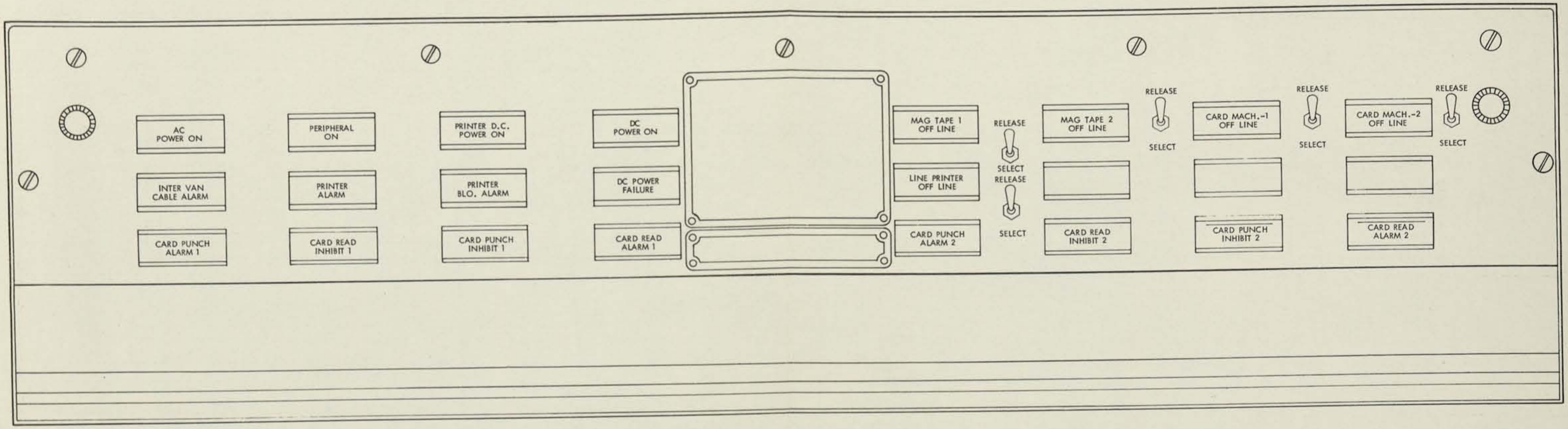


Figure 7-1. Status Panel Layout, Van 1

TABLE 7-1. STATUS PANEL INDICATORS

Indicator	Lens Color
AC POWER ON	Green
PERIPHERAL ON	Yellow
PRINTER D. C. POWER ON	Green
DC POWER ON	Green
INTER VAN CABLE ALARM	Red
PRINTER ALARM	Red
PRINTER BLO. ALARM	Red
DC POWER FAILURE	Red
CARD PUNCH ALARM 1	Red
CARD READ INHIBIT 1	Red
CARD PUNCH INHIBIT 1	Red
CARD READ ALARM 1	Red
MAG TAPE 1 OFF LINE	Yellow
MAG TAPE 2 OFF LINE	Yellow
CARD MACH. -1 OFF LINE	Yellow
CARD MACH. -2 OFF LINE	Yellow
LINE PRINTER OFF LINE	Yellow
CARD PUNCH ALARM 2	Red
CARD READ INHIBIT 2	Red
CARD PUNCH INHIBIT 2	Red
CARD READ ALARM 2	Red

SECTION VIII

ELEMENT TESTER

8.1 GENERAL

An element tester is included in MOBIDIC 7A to permit rapid testing of MOBIDIC computer elements to detect both existing and incipient malfunctions. A quick-look indication of overall element performance is given by displaying the element output visually, either on an oscilloscope for comparison with an "acceptable performance" reference pattern on a transparent oscilloscope face-mask or on direct-reading meters. These tests enable the test operator to locate and reject malfunctioning elements rapidly and accurately; they do not indicate which component is causing the malfunction. By the use of simulated worst-case computer conditions and by applying high or low off-bias voltage, a marginal or questionable element can be detected. The element tester can, through these tests, detect and predict faults in MOBIDIC computer elements.

8.2 ELECTRICAL DESCRIPTION

The element tester draws approximately 6ØØ watts of input power. Its test-result outputs are:

1. Oscilloscope-calibrated waveform.
2. Element-under-test waveform.
3. Meter "A" reading.
4. Meter "B" reading.
5. Test-position indicator lights.

A crystal-controlled clock triggers a pulse generator to generate pulses at the MOBIDIC clock-rate of 1 mcps. These trigger pulses complement a flip-flop, which in turn produces a square wave. The pulse generator output is ANDED with the output of a counter to trigger the oscilloscope and is used for other test purposes. A set of power supplies furnishes all voltage levels necessary for element testing. The test operator can monitor voltages by means of a 12-position rotary-type test selector switch which displays the test voltage on the power test meter.

During testing a test set-up card is inserted in a Cardmatic switch. These cards are made of a rigid vinyl material and are punched with holes in various locations. There is a separate card, with its own individual configuration of punched holes, for each MOBIDIC element type. Inside the Cardmatic switch are two decks of 187 contacts each. Inserting a test set-up card into the Cardmatic switch establishes a closed circuit between corresponding contacts in the upper and lower decks, if the hole position controlling that pair of contacts is not punched. These card-controlled switch circuits route test input current and voltage to the appropriate element input terminals and route the element output to an appropriate test-result display device.

A transparent mask is placed over the face of the oscilloscope tube. On this mask are clear and red areas which have been so laid out that the output waveform for a properly functioning element of the type to be tested will fall completely within the clear area. The oscilloscope-calibrated waveform for this type of element is displayed on the oscilloscope and adjusted until it falls within green lines on the mask which define the appropriate waveform.

The element to be tested is placed in the test jig and the waveform of the element under test is displayed on the oscilloscope. The test selector switch is then manually stepped through a prescribed sequence of positions. Indicator lights on the console show the switch position. If the waveform of the element under test remains within the clear area of the mask, the element is acceptable, if any of the pattern falls in the red area, the element is not acceptable. The "A" and "B" meters give a quick-look indication of capacitor leakage current and, indirectly, of resistor values when passive (that is, transistorless) elements are under test.

The element tester simulates worst case operating conditions, including the electrical effects of environmental conditions and component aging. A worst-case waveform is simulated by so loading down the test signal that it has the maximum allowable rise and fall time and the minimum acceptable voltage. The element being tested is then subjected to high and low marginal checking voltages.

The high voltage forces into failure those transistors with a low or marginal beta; it also shows up any circuits whose turn-on time is too slow. The low voltage prevents those circuits with high transistor I_{co} or badly-drifted component values from turning off; it also shows up those transistors with marginal "hole storage" capacity. Throughout the tests maximum load conditions are applied to the element circuits. The combination of electrical stress conditions insures that elements with marginal transistors or incorrect-value components will be rejected. The test results of a given element can be analyzed so that repair is possible.

8.3 MECHANICAL DESCRIPTION

The tester is a self-contained, console-style unit. It consists of a rack-type cabinet of welded aluminum angle construction covered with 3/32" thick aluminum sheet skins. Overall dimensions are approximately 62" high x 2Ø-1/2" wide x 26" deep. The desk, which is completely detachable from the cabinet, protrudes a distance of 15" to the front. The physical appearance of the element tester is illustrated in the sketch shown in Figure 8-1. The cabinet is shock-mounted at the base and stabilized at the top.

The desk top is about 3Ø" from the floor level. It contains the power test meter, power switch, Cardmatic switch, test jigs, and test set-up cards. Jigs are recessed below the desk surface to minimize the possibility of damage to an element during testing.

The oscilloscope is mounted close to the top of the cabinet with the center of the oscilloscope tube approximately at eye-level in respect to a test operator seated at the tester. The oscilloscope itself is easily removable for maintenance, calibration, and use as a portable test instrument.

Test meters are mounted directly above the oscilloscope. The test-position indicator lights are mounted directly below the meters and above the oscilloscope tube.

A small bin to the right of the oscilloscope tube provides storage for oscilloscope face masks. An access cover below the oscilloscope allows use of an external probe for element testing.

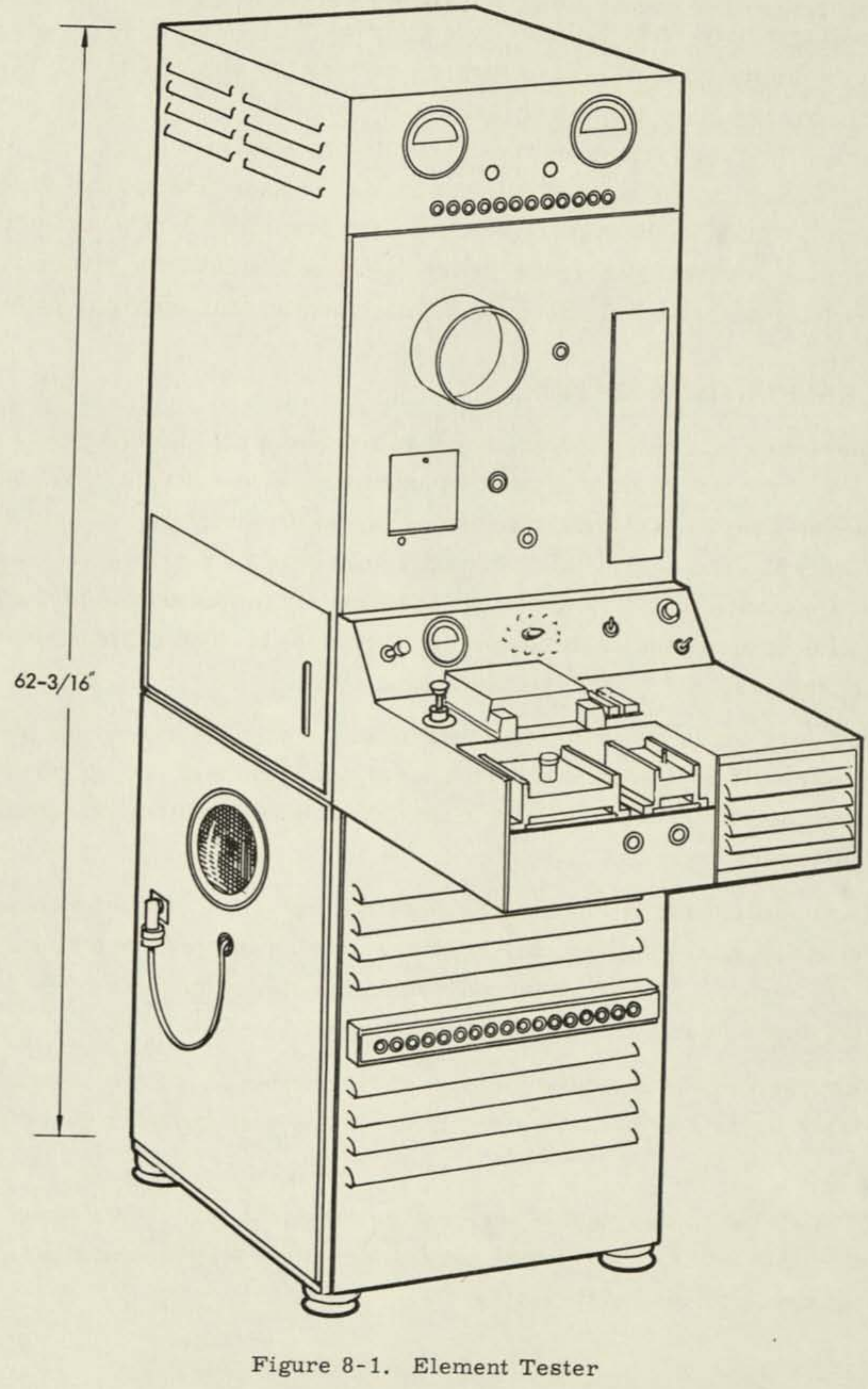


Figure 8-1. Element Tester

The fan-cooled power supplies are chassis-mounted below the desk in the cabinet. Mounted on ball bearing slides, this chassis can be pulled out to the front of the cabinet for maintenance purposes. Power fuses are accessible from the front of the cabinet below the desk.

SYLVANIA ELECTRONIC SYSTEMS

A unit of
GENERAL TELEPHONE & ELECTRONICS

63 SECOND AVENUE, WALTHAM, MASS.