SPERRY RAND

UNIVAC

9200/9200 II
9300/9300 II
9400
SYSTEMS

CARD PUNCH SUBSYSTEM

TYPES 0603-04, 0604-00, 0604-99

PROGRAMMERS REFERENCE

UP-7772

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format. This format provides a rapid and complete means of keeping recipients apprised of UNIVAC® Systems developments. The information presented herein may not reflect the current status of the product. For the current status of the product, contact your local Univac Representative.

The Univac Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware or software changes and refinements. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its Systems.

UNIVAC is a registered trademark of Sperry Rand Corporation.

PRINTED IN U.S.A.

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1. PURPOSE

The UNIVAC Card Punch Subsystem for the 9200/9200 II/9300/9300 II/9400 Systems provides the means for punching 80-column cards, which are used for bulk storage of input-output data.

The UNIVAC Card Punch Subsystem for the UNIVAC 9200/9200 II/9300/9300 II/9400 Systems comprises three types of card punch: 0603–04, 0604–00, and 0604–99. Each of these card punch types is used with the UNIVAC 9200 II, 9300, and 9300 II Systems. The UNIVAC 9200 System uses only the 0603–04 Card Punch, and the UNIVAC 9400 System uses only the 0604–99 Card Punch.

1.2. SCOPE

This manual contains information for the programming of the UNIVAC Card Punch Subsystem for the UNIVAC 9200/9200 II/9300/9300 II/9400 Systems. Referencing the programming information within this manual is unnecessary when appropriate software is available as an interface to the subsystem.

It is assumed that the programmer is already capable on the system level and need only be instructed in the use of the subsystem. Therefore, material already covered in the system manuals is not duplicated here.

This manual is divided into the following basic sections:

- Functional characteristics
- Programming

2. FUNCTIONAL CHARACTERISTICS

2.1. GENERAL

This section of the manual provides information concerning the functional and physical characteristics of UNIVAC 0603–04 and 0604 Card Punches. Included in this section are descriptions of the optional features available for each card punch, subsystem configuration, and interface between the processor and each card punch.

2.1.1. UNIVAC 0603–04 Card Punch

The UNIVAC 0603–04 Card Punch (serial punch) is a freestanding unit which is an integral part of the processor. (See Figure 2–1.) The punch mechanism is located in the card punch cabinet, and the control unit and associated operating controls are located in the processor cabinet.

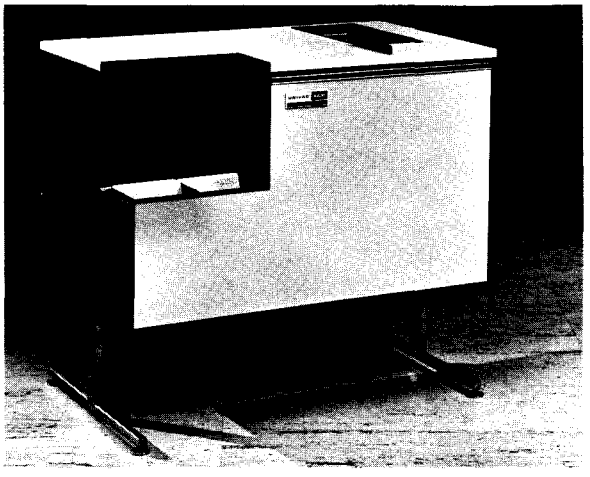


Figure 2–1. UNIVAC 0603–04 Card Punch

The punch mechanism is an 80-column serial punch that operates at a rate of 75 to 200 cards per minute (CPM). The punch rate is dependent on the number of columns punched in each card. If all 80 columns of each card are punched, the punch rate is 75 CPM. If a maximum of 14 columns of each card is punched, the punch rate increases to 200 CPM (see Figure 2–2).

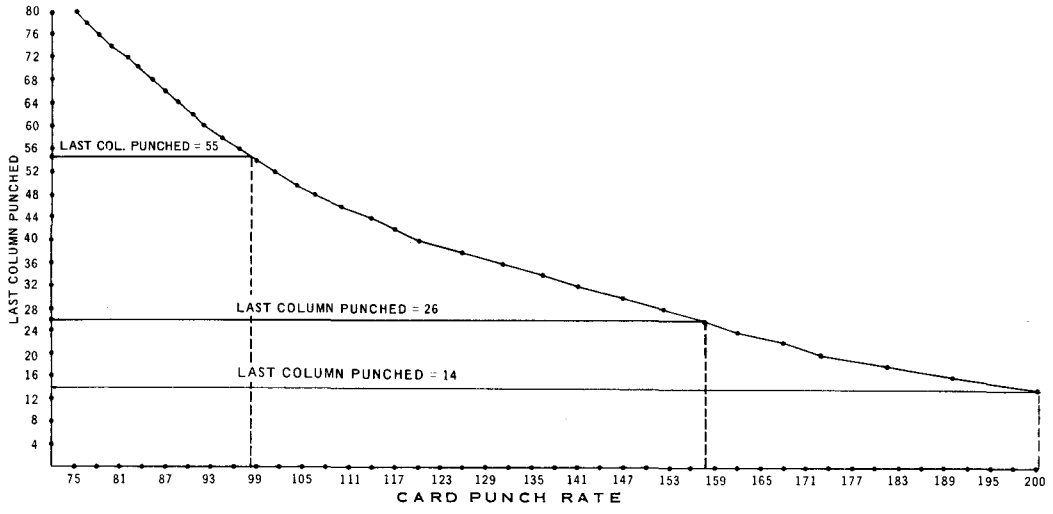


Figure 2–2. Card Punch Rate Versus Last Column Punched

After a card is punched, it is checked to ensure that it was correctly punched. Cards that are incorrectly punched are rejected, and the data is punched on the following blank card.

The serial punch is converted to a reader/punch when optional feature F0870–00 is installed. This feature permits reading data from a prepunched card prior to a punch operation. The card read rate is the same as the maximum punch rate. The serial punch is equipped with a card hopper at the input and two stackers (normal and select) at the output. Program controlled stacker selection is available when optional feature F0871–00 is installed.

Because the serial punch is an integral part of the processor, the control circuitry and associated operating controls and indicators are located in the processor. This area of the processor, designated the processor punch control section, regulates the flow of data and control signals to and from the punch mechanism. Data collected from the processor is checked for validity then sent to the punch mechanism in either the image mode or the compress mode of translation (see 3.3.1.5.1 and 3.3.1.5.2). To ensure that the data is punched in the card correctly, the processor punch control section senses (reads) the data punched on the card and compares it with the data initially sent to the punch mechanism.

Characteristics and salient features for the UNIVAC 0603–04 Card Punch are listed in Table 2–1.

FEATURE	CHARACTERISTICS
Card orientation	Loaded face forward, with 9 edge at bottom
Card rate (minimum)	75 CPM when all 80 columns are punched
Card rate (maximum)	200 CPM when a maximum of 14 columns is punched
Read station sensing	Column by column
Punch station punching	Column by column, two columns at a time
Read rate (optional feature)	200 CPM
Punch translation	Image mode: 160 six-bit bytes per card Compress mode: 80 bytes per card
Hopper capacity	1200 cards
Stacker capacity	
Normal stacker	750 cards
Reject stacker	750 cards

Table 2–1. UNIVAC 0603–04 Card Punch, Characteristics

2.1.2. UNIVAC 0604–00 Card Punch

The UNIVAC 0604–00 Card Punch (row punch) is a freestanding, self-contained unit controlled by commands from the processor through a multiplexer channel (see Figure 2–3). This card punch feeds and punches 80-column cards, row by row, at a maximum rate of 200 CPM. When installed, optional feature F0945–00 increases the maximum punch rate from 200 to 250 CPM. Data to be punched is transferred from the multiplexer channel to an 80-bit punch buffer in the control unit which regulates the flow of data to and from the punch mechanism. Data received by the control unit is checked for validity then sent in either the image mode or the compress mode of translation to the punch mechanism. To ensure that the card is punched correctly, the control unit senses the data punched on the card and compares it with the data initially sent to the punch mechanism. The punched cards are then directed by program control into one of the two card stackers at the output.

When installed, Read/Punch Feature F0875–00 is a prepunch read unit that reads data from a prepunched card, row by row, into the control unit. Additional data can then be punched into the card when the command is received and transferred to the control unit for a comparison check.

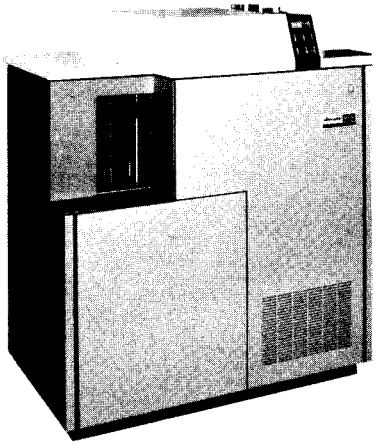


Figure 2–3. UNIVAC 0604–00 or 0604–99 Card Punch

2.1.3. UNIVAC 0604–99 Card Punch

The UNIVAC 0604–99 Card Punch (row punch) is similar to the UNIVAC 0604–00 Card Punch (see Figure 2–3) except that the punch rate is 250 CPM. Characteristics and salient features for the UNIVAC 0604–00 or 0604–99 Card Punches (hereafter also referred to as UNIVAC 0604 Card Punch) are listed in Table 2–2.

FEATURE	CHARACTERISTICS
Card orientation	Fed face down, with 9 edge leading
Punch rate	
Type 0604–00 Card Punch	200 CPM (maximum)
Type 0604–99 Card Punch	250 CPM (maximum)
Read station sensing	Row by row
Punch station punching	Row by row
Read rate (optional feature)	Same as maximum punch rate
Punch translation	Image mode: 160 six-bit bytes per card Compress mode: 80 bytes per card
Hopper capacity	1000 cards
Stacker capacity	
Normal stacker	1000 cards
Select stacker	1000 cards

Table 2–2. UNIVAC 0604 Card Punch, Characteristics

2.2. CONFIGURATION

The card punch subsystem consists of the UNIVAC 0603–04, 0604–00, or 0604–99 Card Punch. Figure 2–4 illustrates the functional arrangment of the subsystem components with the UNIVAC 9200/9200 II/9300/9300 II/9400 Systems. For the following card punch types there are two variations; they differ only in frequency of electrical supply:

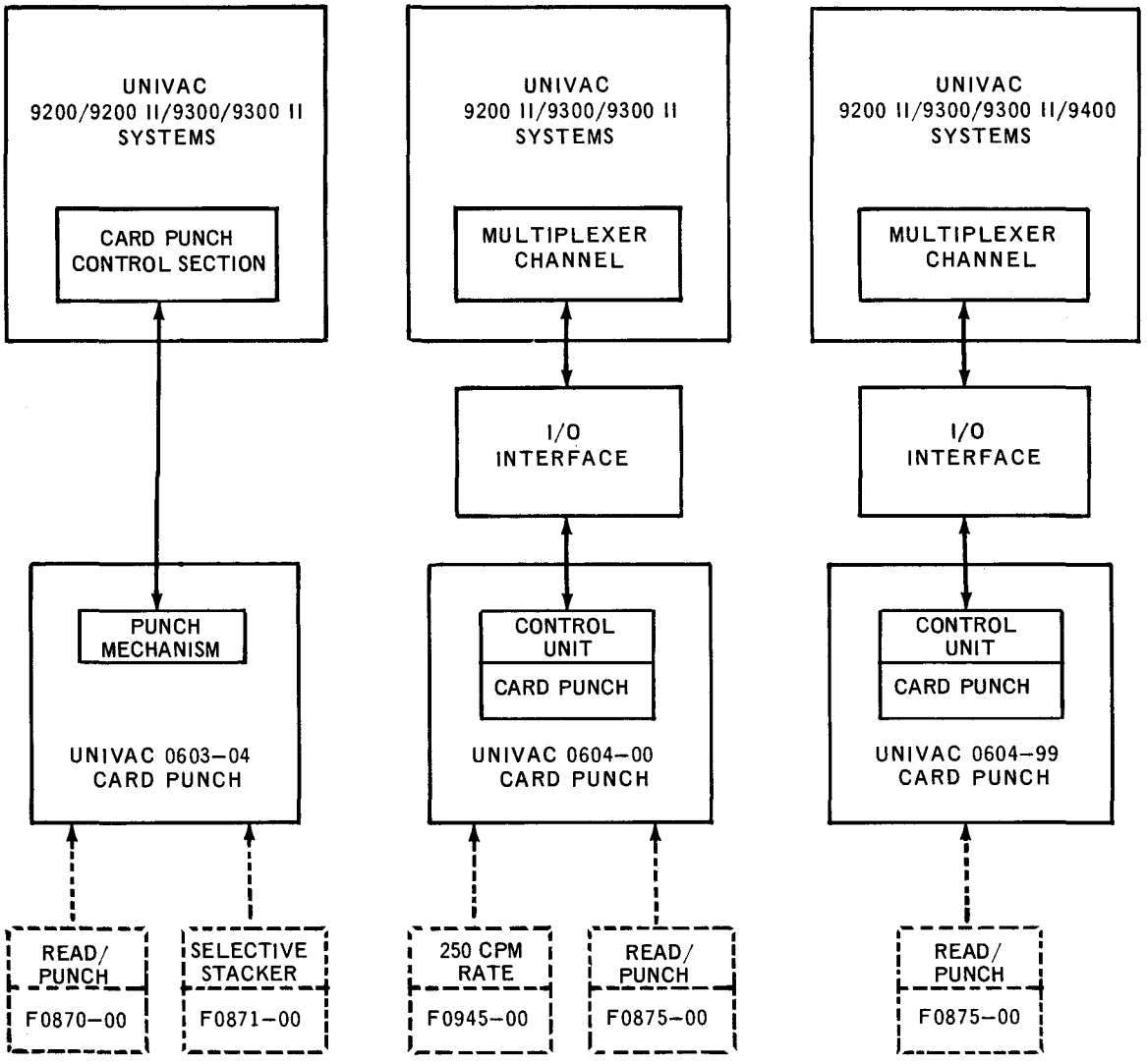
- Card Punch, Type 0603–04 (60 hertz) or Type 0603–05 (50 hertz)
- Card Punch, Type 0604–00 (60 hertz) or Type 0604–01 (50 hertz)
- Card Punch, Type 0604–99 (60 hertz) or Type 0604–98 (50 hertz)

2.3. SUBSYSTEM COMPONENTS

The card punch subsystem components, described in the following paragraphs, consist of the UNIVAC 0603–04, 0604–00, and 0604–99 Card Punches.

2.3.1. UNIVAC 0603–04 Card Punch

The UNIVAC 0603–04 Card Punch (see Figure 2–1) is an 80-column serial-feed card punch. The subsystem component description of the serial punch is divided into two parts: the card punch mechanism and the processor punch control section.



NOTE: Dashed lines indicate optional features.

Figure 2-4. Card Punch Subsystem Configurations

2.3.1.1. Card Punch Mechanism

Functional units of the serial punch mechanism are as follows:

- input station
- ready station
- read station (optional)
- wait-and-advance station
- punch station
- punch check unit
- output station

A description of each unit and its relationship to card flow follows. The card punch feed path is shown in Figure 2–5.

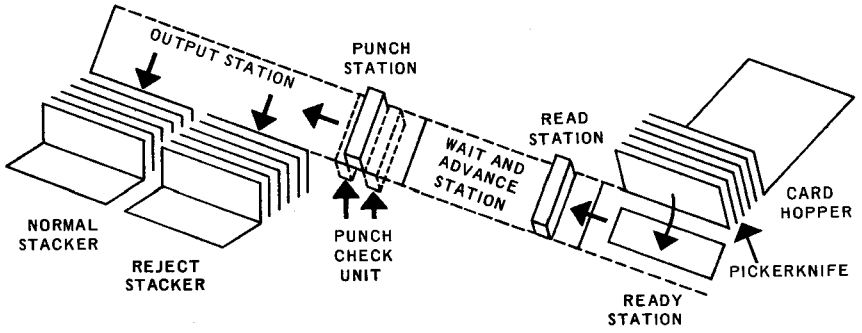


Figure 2–5. Card Feed Path for UNIVAC 0603–04 Card Punch

2.3.1.1.1. Input Station

The card hopper, which has a capacity of 1200 standard-thickness cards, and a pickerknife mechanism comprise the input station. Cards are loaded into the hopper face forward, with the 9 edge at the bottom (the broad-side feed method is used). The pickerknife mechanism feeds cards one at a time from the bottom of the card hopper into the first pair of feed rollers. The operation of the pickerknife mechanism is controlled by a solenoid that is energized by the feed signal.

2.3.1.1.2. Ready Station

The ready station holds one card. Application of a control (feed) signal to a pinch roller causes the card to move in a column-by-column manner into continuously running feed rolls, which transfer the card to the wait-and-advance station. The feed signal that is applied to the pinch roller is the same signal that is used to initiate the transfer of a card from the card hopper to the ready station. The ready station contains a photocell for controlling the time at which a card enters and leaves the ready station to be checked. This photocell functions as a jam detection sensor.

2.3.1.1.3. Read Station

The sensing unit and the sprocket drum are the major components of the read station.

- The *sensing unit* consists of an exciter lamp and 12 photodiode assemblies. As a card passes through the read station, the information punched in the card permits the light to pass through the holes and then activate the corresponding photodiodes. The signals generated by the photodiodes are applied to the processor punch control section.

- The *sprocket drum*, which is constantly rotating while the card punch mechanism is energized, provides timing signals for synchronizing data sensing and data transmission. The angular velocity of the drum is proportional to the linear velocity of the card traveling through the read station.

2.3.1.1.4. Wait-and-Advance Station

The wait-and-advance station controls the card for one cycle. The card is advanced from the wait-and-advance station through the punch station in two-column increments by the card-pusher mechanism. The card-pusher mechanism consists of 41 spring-steel pusher blades attached to an oscillating platform. The wait-and-advance station contains two photocell diodes. The first photocell diode, which can be used as a jam detection sensor, is energized as the trailing edge of the card enters the ready station. In response to a signal from the first photocell diode, the processor punch control section provides a signal to energize the card-brake solenoid. The second photocell diode is energized after the first advance operation to provide a timing signal for the first call for punch information. In response to the signal from the second photocell diode, the processor supplies the necessary punch-data signals, which represent the data to be punched in columns 1 and 2.

2.3.1.1.5. Punch Station

The information to be punched into a card is received from the processor in a column-serial fashion. The data received from the processor is translated into punched holes by 24 punch-actuator interposers that are mechanically interjected between the punch striker bar (which runs constantly) and the specified punch. The information is punched into the card in a column-by-column manner, two columns at a time. Punching accuracy is ±0.007 inch from the nominal hole location. A punch sprocket signal is supplied to the processor punch control section by the punch mechanism once per punch cycle to enable synchronization of the punch mechanism and the processor punch control section.

2.3.1.1.6. Punch Check Unit

The punch check unit, which provides the capability of sensing and verifying that all punches have physically penetrated the card, consists of 24 proximity transducers. Each of these transducers generates an amplitude-modulated signal when the corresponding cutting punch penetrates the card. The output signals from these transducers are detected and interpreted by the processor punch control section.

2.3.1.1.7. Output Station

The postpunch area and two card stackers comprise the output station.

- The postpunch area is the area immediately following the punch station. Located in the postpunch area are a pinch roll and a photocell diode. A signal applied to the pinch roll from the processor punch control section causes the card to be ejected rapidly from the punch station. This signal may be applied to the pinch roll any time after the 14th column has been punched. The photocell diode is used as a jam detection sensor; also, the signal generated by this photocell diode indicates to the processor punch control section that the energizing signal applied to the pinch roll should be removed.

- Two card stackers are provided to receive cards exiting from the punch station; these are the normal stacker and the reject stacker. Each of these stackers has a capacity of 750 standard-thickness cards. A switch located at the bottom of the stacker assembly generates the appropriate control signal when either stacker becomes full. All cards that are punched correctly are routed to the normal output stacker; those punched incorrectly are routed to the reject stacker.

2.3.1.2. Processor Punch Control Section

The punch control section, which is a part of the general I/O structure of the processor, controls and monitors the operation of the serial punch mechanism. Comprising control logic and data paths, this section is designed to perform the following functions:

- to supply signals which prepare the punch mechanism for punch operations in one of two modes of translation;
- to synchronize the flow of data to and from the punch mechanism;
- to detect and interpret signals, both normal and abnormal, from the punch mechanism.

If the serial punch is conditioned for a punch operation and cards have advanced from station to station until a card is ready to enter the punch station (see Figure 2–5) on command from the processor, the processor punch control section initiates and controls operations in this sequence:

(1) Data is transferred in either the image mode or the compress mode of translation from the processor main storage to the punch mechanism through data/control paths of the processor punch control section. The transferred data is punched on a card fed from the wait-and-advance station. To ensure that the card is punched correctly, the processor punch control section receives and interprets signals from the punch check unit as the card is being punched.

(2) At the completion of the punch operation, the processor punch control section causes the punched card to be transferred to either the normal stacker or the reject stacker.

(3) If the read/punch feature (F0870–00) is installed and cards are to be read, a card is fed from the ready station through the read station to the wait-and-advance station. As the card passes through the read station, data is read in either the image mode or the compress mode and applied to the processor punch control section, which transfers the data to the assigned processor memory storage area.

(4) The next card is fed from the hopper to the ready station, and the above process is repeated for each card in turn.

2.3.2. UNIVAC 0604 Card Punch

The UNIVAC 0604 Card Punch is an 80-column row-by-row card punch. The subsystem component description of the row punch is divided into two parts: the card punch mechanism and the control unit.

2.3.2.1. Card Punch Mechanism

Functional units of the card punch mechanism are as follows:

- input station
- wait station 1
- prepunch read station (optional)
- wait station 2
- punch station
- postpunch read station
- output station

A description of each unit and its relationship to card flow follows. The card feed path is shown in Figure 2–6.

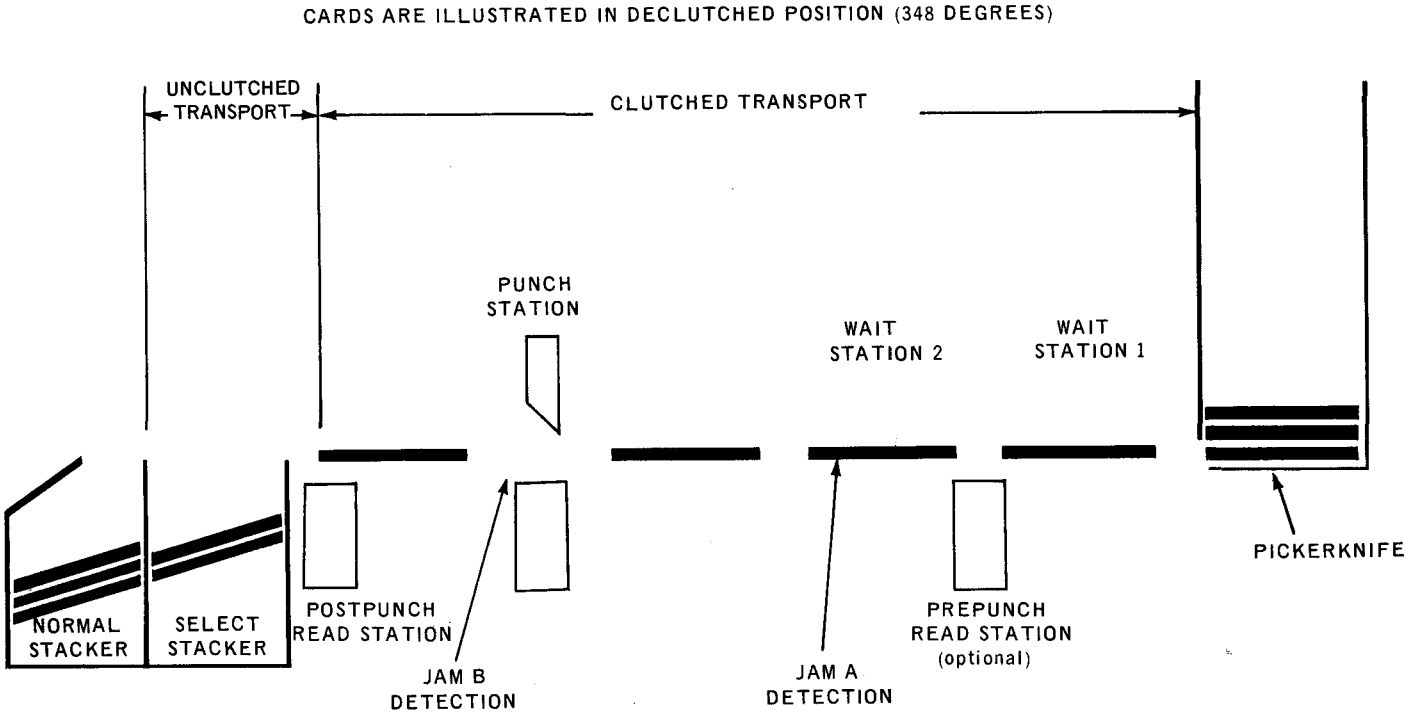


Figure 2–6. Card Feed Path for UNIVAC 0604 Card Punch

2.3.2.1.1. Input Station

The card hopper and the pickerknife mechanism constitute the input station. The hopper has a capacity of 1000 cards, which are stacked face down in a vertical column with the 9 edge leading. The pickerknife mechanism feeds one card at a time from the bottom of the card hopper into the first pair of feed rolls.

2.3.2.1.2. Wait Station 1

Wait station 1, which includes the first pair of feed rolls, holds the card for one cycle of the clutched transport system.

2.3.2.1.3. Prepunch Read Station

The prepunch read station provides row-by-row sensing of a card as it leaves wait station 1. Sensing is accomplished by means of flexible brushes which protrude through the punched holes of the card to make contact with the sensing roll. As each successive card row passes over the brushes, the control unit applies a probe pulse to the sensing roll. Where a hole is present in a particular row, the brush will protrude through the hole and make contact with the sensing roll. The same action takes place simultaneously for every hole detected in that particular row. An interlock switch prevents operation of the read/punch function if the brushes are incorrectly positioned.

2.3.2.1.4. Wait Station 2

In wait station 2, two functions are performed: The card is held for another cycle of the clutched transport system, and the position of the card in the card path is checked. Upon entering wait station 2, the leading edge of the card is sensed by a photocell diode, and the trailing edge is sensed as the card leaves wait station 2. This is known as A jam check. Any malfunction at this station lights the A JAM half of the MAN FEED A JAM/B JAM switch/indicator on the operator's control panel.

2.3.2.1.5. Punch Station

Data punched in a card at this station is received from the multiplexer channel and loaded into an 80-bit punch buffer. When the punch-activating mechanism is set to receive the new data, the punch buffer is unloaded and the data is translated into punched holes (row by row) by punch-activating interposers. These interposers are mechanically interjected between the punch drive bar (which runs constantly) and the specified punch. This process continues until all 12 rows of the card have been punched. As the card emerges from the punch station, it is checked by the B jam photocell diode at the leading and trailing edges of the card for position or card jam. Any malfunction at this station will light the B JAM half of the MAN FEED A JAM/B JAM switch/indicator on the operator's control panel.

2.3.2.1.6. Postpunch Read Station

In the postpunch read station, the same type of operation as was performed at the prepunch read station (see 2.3.2.1.3) is performed. The information sensed is loaded into an 80-bit read buffer where it is made available to the multiplexer channel. The read stations are located so that when one row of a card is being sensed at the postpunch read station, the sensing brushes of the prepunch read station are halfway between rows of another card. This half-cycle offset allows a single read buffer to be time shared between the stations.

2.3.2.1.7. Output Station

The output station consists of two card stackers designated normal and select. Each has a capacity of 1000 standard cards. The output station is equipped with a card deflector which, when energized, intercepts and delivers a card to the select stacker. The normal stacker accepts all cards not delivered to the select stacker. When either stacker is filled to capacity, a switch located at the base of the stacker assembly shuts off the main drive.

2.3.2.2. Control Unit

The control unit which coordinates the operations within the row punch, comprises buffer storage, control logic, and data paths. The internal hardware is designed to perform the following functions:

- to receive commands from the processor and prepare the row punch for different modes of operation and for data handling;
- to synchronize the flow of data between the row punch and the multiplexer channel; and
- to interpret signals, both normal and abnormal, from the row punch and to notify the processor of conditions within the unit.

If the row punch is conditioned for a punch operation and cards have advanced from station to station until a card is ready to enter the punch station (Figure 2–6), on command from the processor the control unit initiates a feed and punch signal. The command causes data to be transferred from an 80-bit punch buffer to the punch-activating mechanism. The data is punched in the card being fed from wait station 2. After each row is punched, it is read (sensed) and the information is stored in the read buffer. To verify that the data was punched into the card accurately, the sensed data is transferred to a postread hole counter for a hole count check. If the sensed data does not pass the hole count check, appropriate bits in the status byte and sense data byte will be set and card punch operations will stop (see 3.3.1.4 and 3.3.1.5).

At the normal completion of the punch operation, a command from the processor causes the control unit to direct the punched card into either the normal stacker or the select stacker.

If the read/punch feature (F0875–00) is installed and a read command is issued to the control unit from the processor, a card is fed from wait station 1 into the prepunch read station. As the card passes through the prepunch read station, each row of punched information is sensed and transferred to the read buffer for subsequent presentation to the multiplexer channel.

2.4. OPTIONAL FEATURES

Optional features are available for each card punch type included in the card punch subsystem. Table 2–3 lists each optional feature, the feature number, the card punch type with which each feature is used, and a brief description of each feature.

OPTIONAL FEATURE	FEATURE NUMBER	CARD PUNCH TYPE	DESCRIPTION
Read/Punch	F0870–00	0603–04	Permits prepunch reading of 80-column cards. Read/punch speed is 75–200 CPM; read only speed is 200 CPM.
Selective Stacker	F0871–00	0603–04	Permits program-controlled stacker selection.
250 CPM Rate	F0945–00	0604–00	Increases the card feed rate from 200 CPM to 250 CPM.
Read/Punch	F0875–00	0604–00, 0604–99	Permits prepunch reading of 80-column cards.

Table 2–3. UNIVAC Card Punch Subsystem, Optional Features

2.5. INTERFACE BETWEEN PROCESSOR AND UNIVAC 0603–04 CARD PUNCH

The UNIVAC 0603–04 Card Punch, used with UNIVAC 9200/9200 II/9300/9300 II Systems, is under direct control of the processor punch control section. The interface between the processor and the serial punch mechanism is a data/control path which consists of a number of lines from the processor punch control section to the punch mechanism, and from the punch mechanism to the processor punch control section.

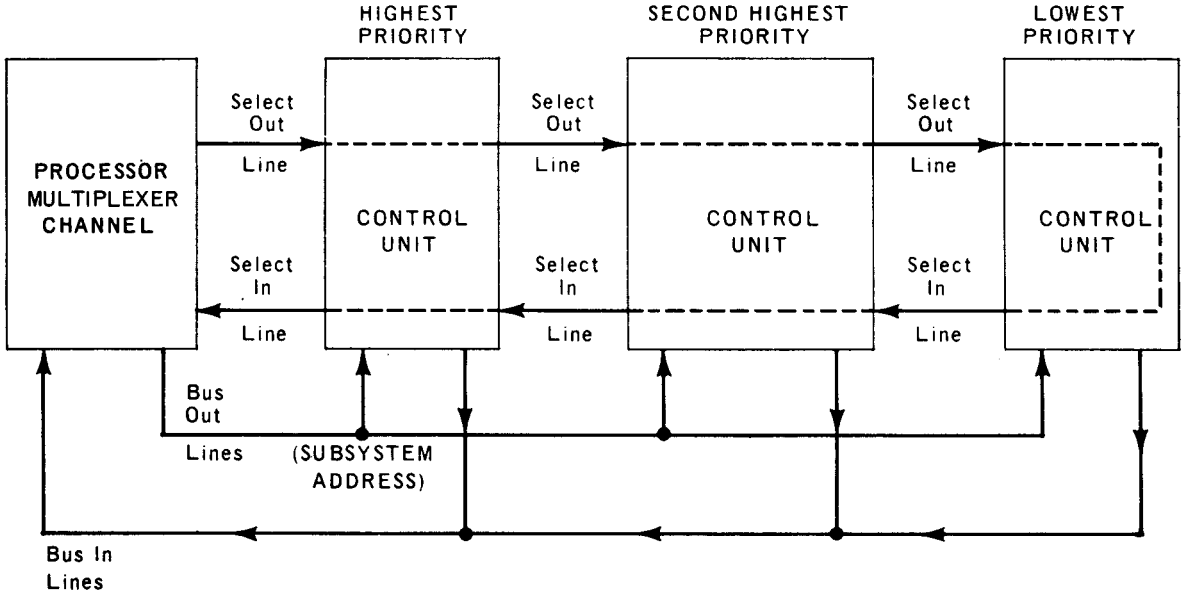
2.6. INTERFACE BETWEEN PROCESSOR AND UNIVAC 0604 CARD PUNCH

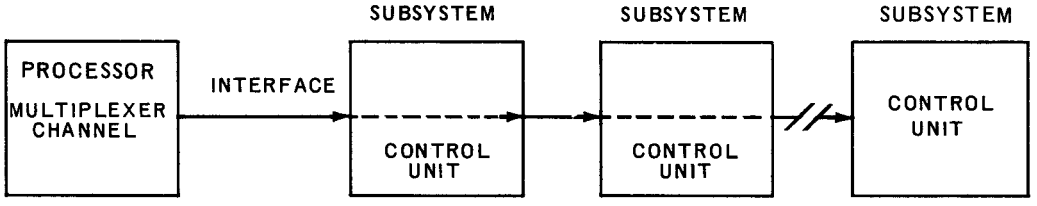
The interface between the processor and the UNIVAC 0604 Card Punch adapts the control signals supplied by the multiplexer channel to the form required by the card punch. The following paragraphs describe the interface lines and the operation of the interface lines in the different sequences required for input/output (I/O) instructions.

2.6.1. Interface

The I/O interface (Figure 2–7) provides the means for time sharing communications between each processor multiplexer channel (UNIVAC 9200 II/9300/9300 II/9400 Systems) and associated control units. The interface consists of four groups of lines: I/O busses, control lines, priority lines, and interlock lines.

When an operation is initiated, the multiplexer channel applies a signal to the select out line. The select out signal is sequentially applied to each control unit (passes serially through all control units) in order of priority.





NOTE: 1. A maximum of eight control units can be connected to the interface.
2. The normal system configuration allots one of these control units to the card punch subsystem.

Figure 2-7. Interface to Multiplexer Channel (Part 1 of 2)

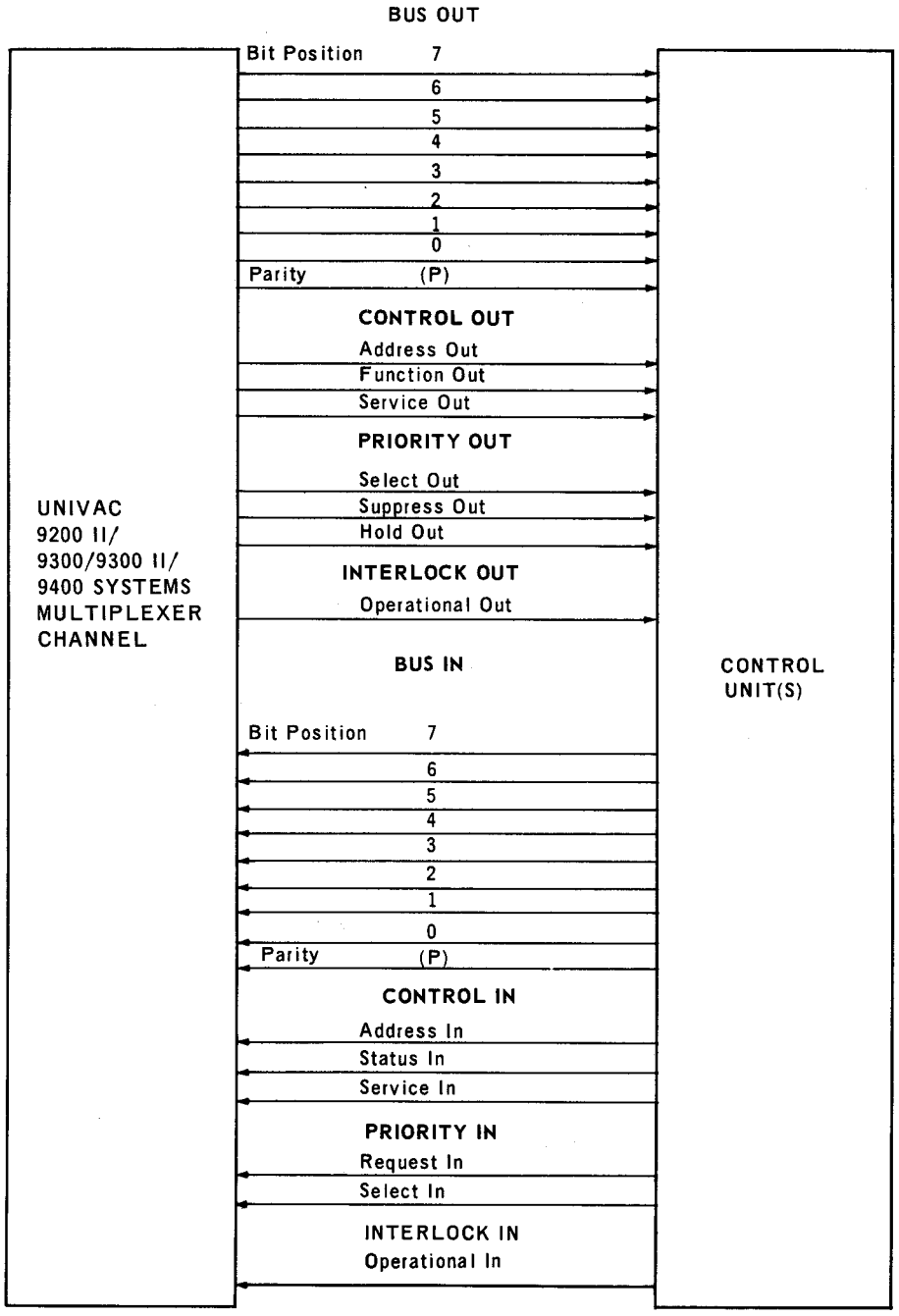


Figure 2–7. Interface to Multiplexer Channel (Part 2 of 2)

The select out signal is applied to the subsystem with the highest priority. If the address is not that of the addressed unit or the unit does not require service, the select out signal is applied to the subsystem with the next highest priority. This operation continues until the select out signal is applied to the addressed or requesting subsystem. If none of the control units recognizes the address or requires servicing, the lowest priority control unit sends the select in signal to the multiplexer channel.

When a subsystem accepts the address or is the requesting unit, it captures the interface by applying a signal to the operational in line and thereby inhibits passage of the select out signal to the next lower priority subsystem. Normally, the subsystem retains control of the interface for a short interval (execution of immediate commands or transfer of a single byte of data) and relinquishes control by passing the select out signal to the next lower priority unit. During the next initiation of the select out signal, the operation is repeated; thus, another data byte is transferred. This process is repeated until the transfer operation is complete and the channel is notified of this status.

2.6.1.1. Bus Lines

The bus out lines carry addresses, commands, and data from the multiplexer channel to the subsystem control units. The bus in lines carry addresses, data, status, and sense information from a subsystem control unit to the multiplexer channel. The type of information contained on the bus out and bus in lines is identified by a simultaneous signal on the appropriate control line. (See Figure 2–7.)

The bus lines consist of nine bus in and nine bus out lines (each group consists of eight data lines and one parity line). Except for control signals, all data transmitted between the processor and subsystems is carried on these lines. Any signal provided by the processor multiplexer channel is common to all control units. However, only one control unit at a time can be logically connected to the multiplexer channel.

Bit position 7 on a bus is designated the low-order value of a byte, and bit position 0 is designated the high-order value, with intervening bits in descending order. When it is necessary to transmit less than eight data bits, the bits must be located in the highest number adjoining bit positions of the bus. All unused lines must be the low numbered lines (bit position 0 and adjoining bit positions). The parity bit must always be the result of odd parity.

2.6.1.2. Control, Priority, and Interlock Lines

When these lines are active, they identify the data on the bus lines and in some cases permit an operation to be performed. The signal name, the origin of the signal, its purpose, and the effect of each control and priority line and the interlock line are given in Table 2–4.

SIGNAL NAME	ORIGIN	PURPOSE
CONTROL LINES		
Address in	Control unit	Indicates that address of currently selected control unit requiring service is on bus in line.
Status in	Control unit	indicates that a status byte is present on bus in lines.
Service in	Control unit	Signals multiplexer channel when selected control unit is prepared to transmit or receive a byte of information. The type of information depends on the operation. The multiplexer channel responds with a service out, command out, or an address out (during a disconnect) signal.
Address out	Miliplexer channel	Signals control units to decode control unit address present on bus out lines. The addressed control unit responds with an operational in signal.
Command out	Multiplexer channel	Indicates that a command byte is present on bus out lines. The command byte notifies the selected control unit to start, continue, or stop an operation. Response to an address in signal means *start* the operation. Response to a service in signal means *stop* the operation. Response to a status in signal means stack (retain status information at control unit until next selection sequence).
Service out	Multiplexer channel	Acknowledges receipt of a service in or status in signal from a control unit. Indicates to the selected control unit that the multiplexer channel has received a data byte.

Table 2-4. Control, Priority, and Interlock Lines Between the Processor and the UNIVAC 0604 Card Punch (Part 1 of 2)

SIGNAL NAME	ORIGIN	PURPOSE
PRIORITY LINES		
Request in	Control unit	Indicates that control unit is prepared to transmit status or data, and requests a selection sequence.
Select out	Multiplexer channel	Establishes communications with control unit whose address is on bus in lines.
Select in	Control unit	Indicates address byte is not recognized by any control unit (completes path of select out signal).
Suppress out	Multiplexer channel	Informs control units not to generate service requests in an attempt to seize the interface. This signal suppresses any subsequent data or status transfer.
Hold out	Multiplexer channel	Indicates that select out signal will begin. This signal is present only for the time the select out signal is present and terminates when a control unit seizes the interface.
INTERLOCK LINES		
Operational in	Control unit	Indicates that a control unit has been selected and is to be present for the entire period of time that data is being transmitted to the multiplexer channel.
Operational out	Multiplexer channel	Allows operations between control units and multiplexer channel. When this signal is not present, all lines to the channel are rendered inoperable.

Table 2–4. Control, Priority, and Interlock Lines Between the Processor and the UNIVAC 0604 Card Punch (Part 2 of 2)

2.6.2. Operation of Interface

The multiplexer channel, by means of the interface, controls the transfer of command, data, sense, and status information. The interface accomplishes the transfers by permitting the operational sequences listed here. These operational sequences are also illustrated in Figure 2–8:

- Initial section
- Control unit request
- Status transfer
- Data transfer
- Control unit busy

2.6.2.1. Initial Selection Sequence

An initial section sequence is started when the multiplexer channel addresses a control unit during the execution of a start-I/O instruction (see Figure 2–8). The format for the address portion of the start-I/O instruction is shown in Figure 2–9.

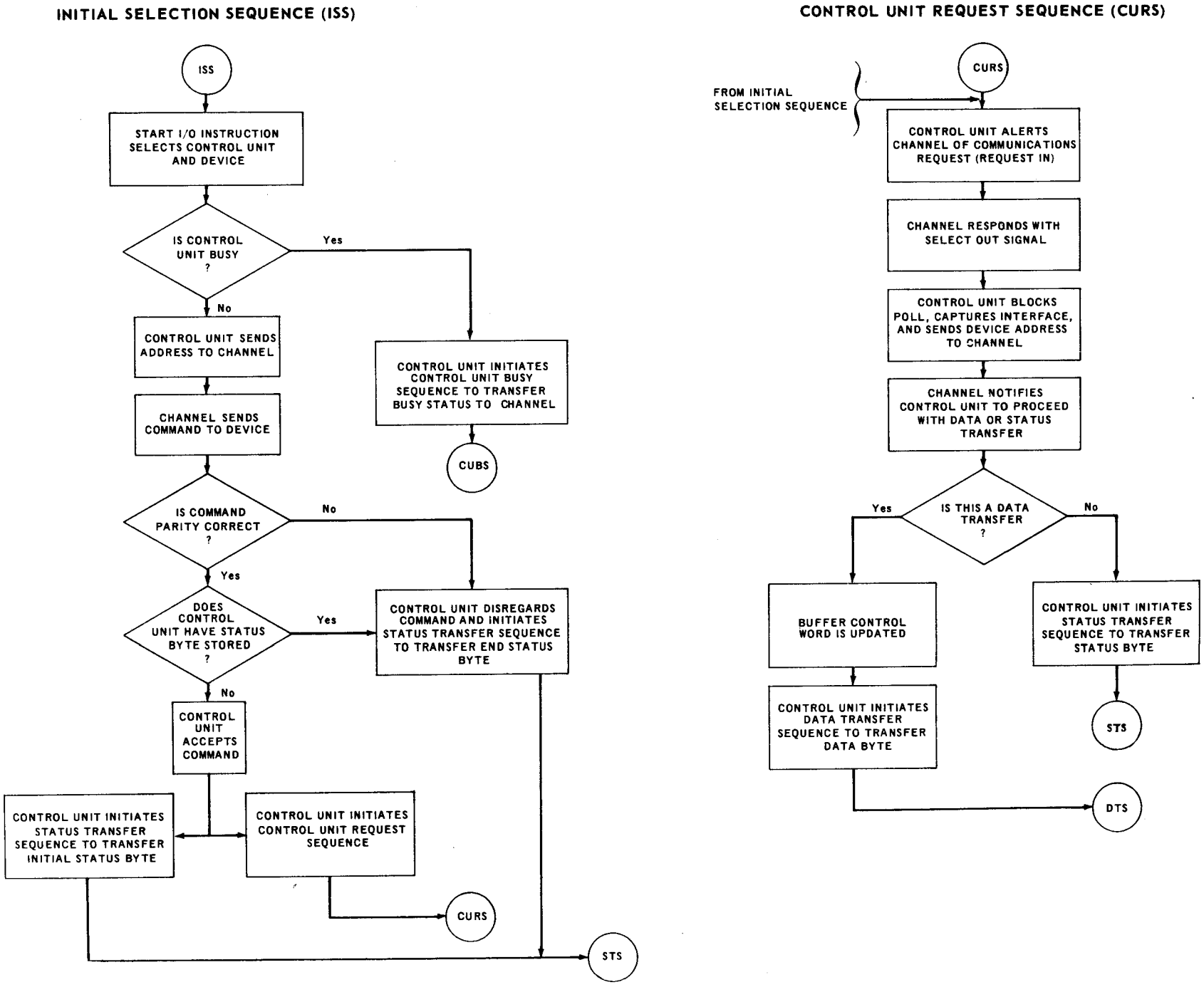
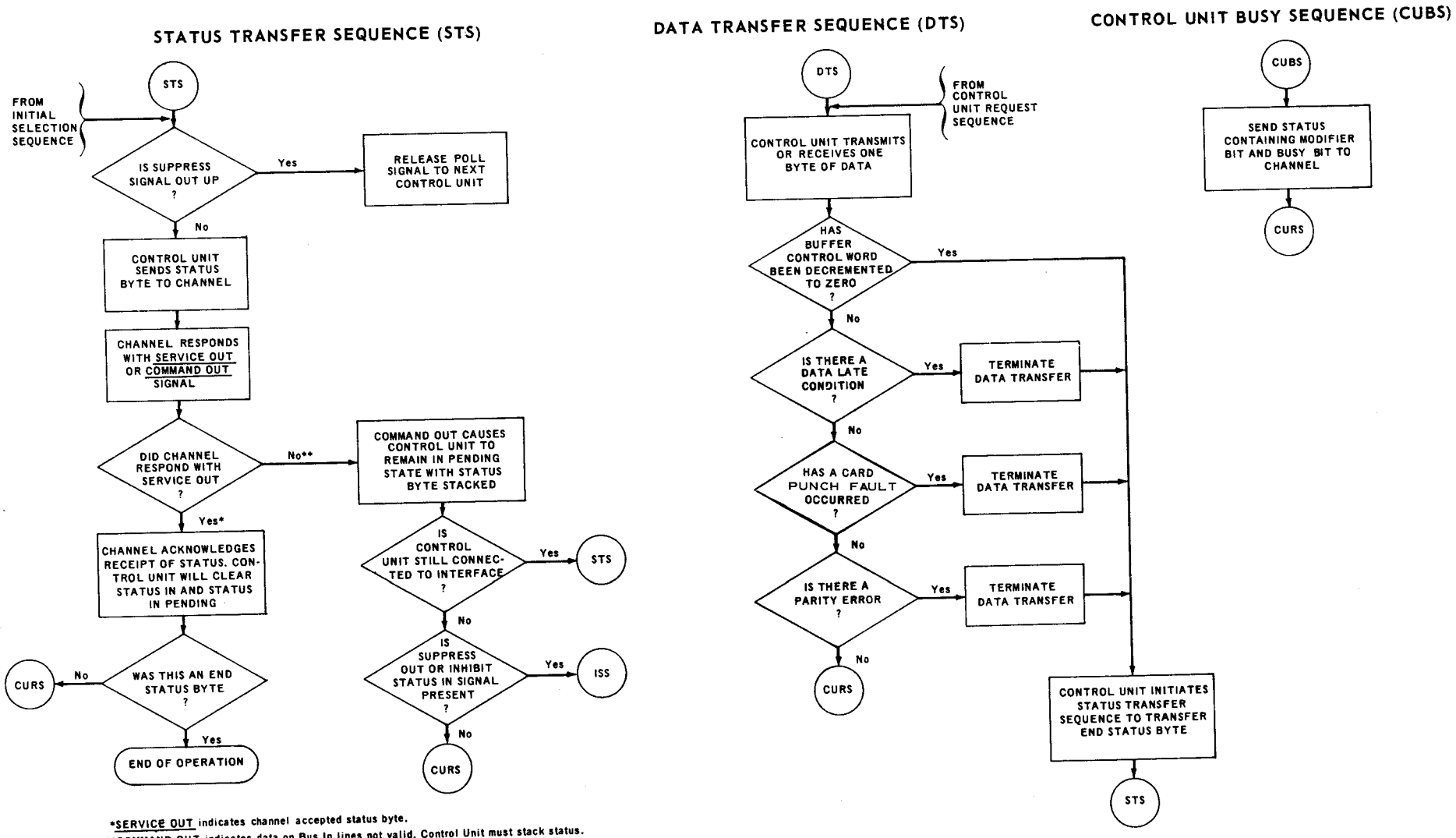


Figure 2-8. Type 0604 Card Punch Control Unit Operational Sequences (Part 1 of 2)



*SERVICE OUT indicates channel accepted status byte.
**COMMAND OUT indicates data on Bus In lines not valid. Control Unit must stack status.

Figure 2–8. Type 0604 Card Punch Control Unit Operational Sequences (Part 2 of 2)

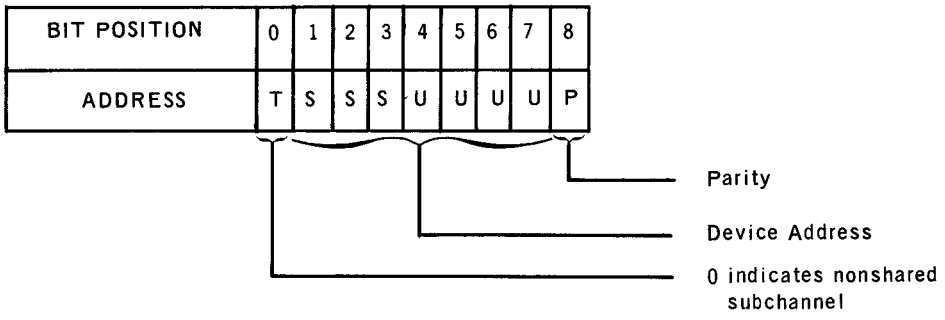


Figure 2–9. Address Format

Prior to performing the incoming command, the control unit checks for the following conditions:

- invalid address
- control unit busy
- command parity incorrect
- stored status

If any of these conditions are present, the control unit rejects the command and sends an appropriate status byte to the multiplexer channel (see 3.3.1.4).

For a start-I/O instruction, the control unit initiates the following:

- status transfer sequence to indicate equipment status at the initiation of a specific command (initial status byte);
- control unit request sequence (see 2.6.2.2);
- status transfer sequence at end of the initial section sequence to indicate acceptance of command (end status byte).

Execution of a sense or read command means that two status bytes (initial status byte and end status byte) must be generated. When the multiplexer channel indicates that the initial status byte is processed, the control unit then executes the command. At the conclusion of the command, another status transfer sequence is initiated. The end status byte and a device end bit are sent to the multiplexer channel indicating equipment status if the command has been successfully completed. When the status byte is accepted by the multiplexer channel, the status-in-pending state is cleared and the status register is cleared to binary 0. The control unit is disconnected from the interface at the end of every data transfer or status transfer sequence.

2.6.2.2. Control Unit Request Sequence

The control unit request sequence is initiated when the control unit is required to communicate with the multiplexer channel and is disconnected from the interface. For read, sense, or end status byte operations, the control unit sends the request in signal to the multiplexer channel, and the channel responds only if the external interrupts are not locked out. The control unit captures the interface with the operational in signal and identifies the control unit requesting attention. The multiplexer channel responds with the command out signal, the buffer control word is decremented, and the control unit then proceeds to perform a data transfer to status transfer sequence as required.

2.6.2.3. Data Transfer Sequence

A data transfer sequence (to or from channel) is normally initiated at the conclusion of a control unit request sequence for operations which require the transfer of data or sense data bytes.

For an input operation (control unit to multiplexer channel), the multiplexer channel initiates either a service out or command out signal. The service out (proceed) signal causes the control unit to transmit another data byte during the next data transfer sequence. The command out (stop) signal is generated by the multiplexer channel when the buffer control word is decremented to 0. The control unit then proceeds to initiate a status transfer sequence.

An output operation (processor to control unit) is similar to an input operation except the control unit samples the output data byte on the bus out lines during the time the service out signal is present. The data transfers terminate when the command out signal, in response to the service in signal, is received at the control unit.

During the data transfer sequence, any of the abnormal conditions listed here can be detected in the addressed tape unit.

- Data Late

  If the multiplexer channel does not respond with a service out or command out signal within a specific time frame, the current operation is aborted. The end status byte contains a 1 bit in the unit check bit position (bit 6), and a sense data byte (in response to a sense command) is transmitted to the multiplexer channel (with appropriate bit set) to indicate a data late condition (see 3.3.3).

- Parity Error

  If a parity error is detected on the bus out lines, the current operation is aborted. The end status byte contains a 1 bit in the unit check bit position (bit 6), and a sense data byte 1 (in response to a sense command) is transmitted to the multiplexer channel (with appropriate bit set) to indicate a parity error condition (see 3.3.3).

2.6.2.4. Status Transfer Sequence

Depending on conditions, the status transfer sequence causes the control unit to send either the initial status or end status byte (see 3.3.2) to the multiplexer channel in such a way that:

- The *initial* status byte is sent to the multiplexer channel during this sequence after successful initiation of the current operation.
- The *end* status byte is generated during this sequence to indicate completion of current operation.

2.6.2.5. Control Unit Busy Sequence

The control unit busy sequence is entered from an initial selection sequence if at time of selection the control unit is executing a previously initiated operation or contains a status byte for a tape unit being addressed. A status byte containing a 1 bit in the status modifier and busy bit positions is transmitted to the multiplexer channel.

2.6.2.6. System Reset Operation

This operation is performed when the SYSTEM RESET switch on the console is pressed. This switch is pressed:

(1) when power is initially turned on;

(2) when the multiplexer channel has been disconnected from the interface; and

(3) as part of the initial program-loading procedure.

A system reset operation is indicated at the interface:

(1) by the concurrent absence of the operational out and suppress out signals (see Table 2–4); and

(2) when the control unit is in the online mode.

This condition causes the operational in signal to be absent and the control unit and its status to be reset. The control units will be in a busy status for the duration of the reset procedure.

2.6.2.7. Selective Reset Operation

This operation terminates data transfers and resets the selected control unit when a malfunction is detected at the multiplexer channel.

A selective reset operation is indicated at the interface by the presence of the select out signal and by the absence of the operational out signal (see Table 2–4). This condition causes the operational in signal to be absent and the operating control unit and its status to be reset.

2.6.2.8. Interface Disconnect

In this operation, the selected control unit completes its current operation and releases the interface.

An interface disconnect operation is indicated at the interface by the presence of the address out signal and by the absence of the select out signal prior to completion of any signal sequence.

When the control unit is disconnected from the interface, it responds by removing all signals (with the exception of request in) from the interface. If the control unit was effecting an input operation at time of disconnect, the data on bus in lines need not be valid. If the disconnect occurred during an output operation, the data on the bus out lines must be valid until the absence of the service in or operational in signal. When the control unit reaches the normal ending point it attempts to obtain selection on the interface to present any generated status to the multiplexer channel.

If the interface disconnect occurred prior to the acceptance of the initial status or after the acceptance of the device end status byte for an operation by the multiplexer channel, the control unit does not generate a status byte as a result of the interface disconnect.

3. PROGRAMMING

3.1. GENERAL

This section of the manual provides information concerning input/output (I/O) control of the UNIVAC 0603–04 Card Punch (which is an integral part of the UNIVAC 9200/9200 II/9300/9300 II Systems) and of the UNIVAC 0604 Card Punch (which is used with the UNIVAC 9200 II/9300/9300 II/9400 Systems by way of a multiplexer channel).

3.2. UNIVAC 0603–04 CARD PUNCH I/O OPERATION

The UNIVAC 0603–04 Card Punch responds to commands specified by bits 27 through 31 of the execute-I/O (XIOF) instruction (UNIVAC 9200/9200 II/9300/9300 II Systems).

EXECUTE–I/O (XIOF) INSTRUCTION

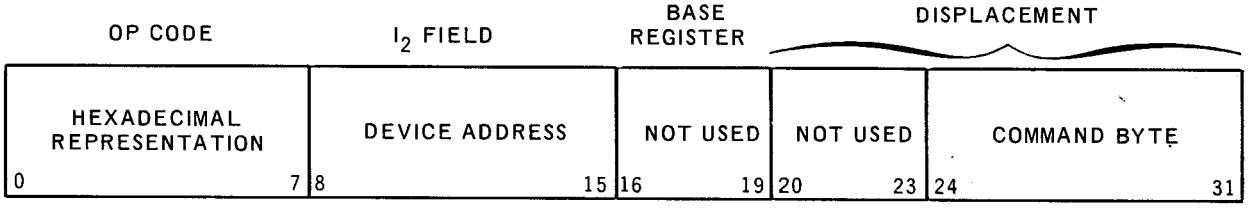
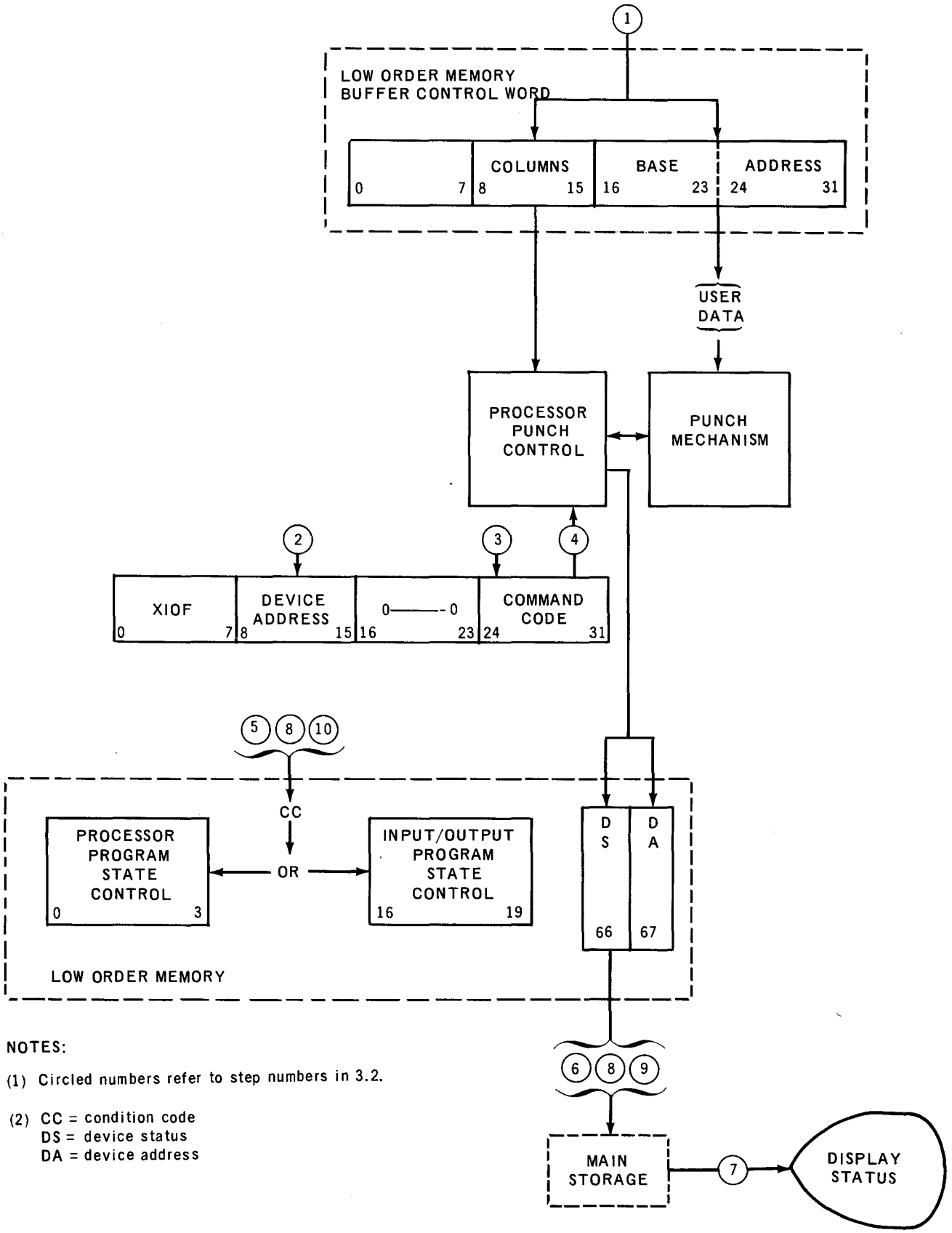


Figure 3–1 shows the functional relationship between the processor and the serial punch when a command is to be executed. When the concurrent operating system (COS) or the nonconcurrent operating system (NCOS) is in use, the execution of commands for the serial punch occurs only when the processor is in the input/output program state.



NOTES:

(1) Circled numbers refer to step numbers in 3.2.

(2) CC = condition code
DS = device status
DA = device address

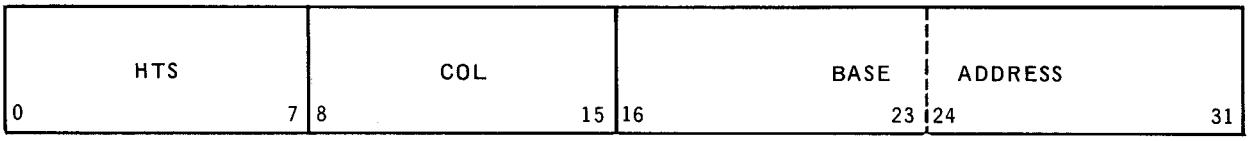
Figure 3–1. Command Function Relationship Between UNIVAC 9200/9200 II/9300/9300 II Systems and UNIVAC 0603–04 Card Punch

When the minimum operating system (MOS) is in use, all commands for the serial punch are executed in the processor program state. A knowledge of the *UNIVAC 9200/9200 II/ 9300/9300 II Systems Card Assembler Programmers Reference, UP-4092* (current version) is necessary when using the information that follows.

Listed below are the initial steps required to execute a command for the UNIVAC 0603–04 Card Punch by use of the UNIVAC 9200/9200 II/9300/9300 II Systems.

(1) Load the proper buffer control word. (For reading, load locations 72, 73, 74, and 75; for punching, load locations 76, 77, 78, and 79.

The buffer control word for the card punch contains:



where: HTS = Hardware temporary storage reserved for the reader. This byte should not be loaded by the program.

COL = The number of columns to be read or to be punched. For reading, this number must always be 80; for punching, any even nonzero number less than 81 may be be specified. At the end of a card operation, this count is decremented to zero.

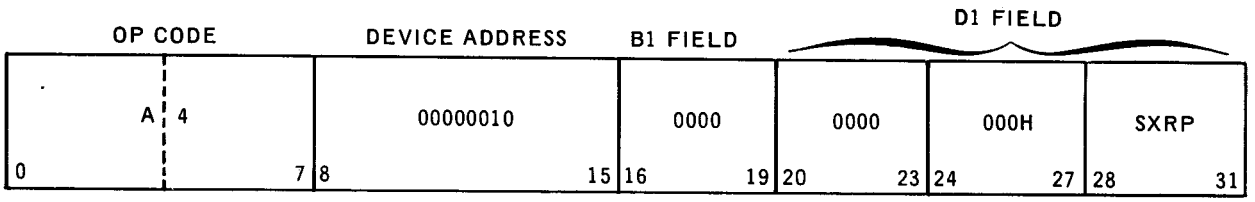
BASE ADDRESS = The address of the most significant half-word (even-numbered address) of the card read area in memory. Upon completion of the operation, this address is one greater than the address of the last byte into which information was read, or this address will point to the byte immediately following the last byte of information punched.

(2) Load the device address into bit positions 8 through 15 of the XIOF instruction.

(3) Load the command byte into bit positions 24 through 31 of the XIOF instruction. If the MOS is used, all interrupts must be inhibited (bit 27 must be a 1).

If the COS or NCOS is used, interrupts must not be inhibited (bit 27 is a 0).

(4) Issue XIOF instruction.



where: H = 1 Inhibit all interrupts (for dedicated I/O device).

P = 1 Punch a card.

R = 1 Read a card.

X = 0 Read and/or punch a card in compressed mode (see 3.3.1.5.2).

X = 1 Read and/or punch a card in image mode (see 3.3.1.5.1).

S = 1 Select stacker. Effective only if the program stacker select feature is installed. Otherwise, this specification is ignored.

Either the R or P bit must be 1 (both cannot be 0). All other bits shown as 0's must be 0's, or an error may result.

Feeding with no reading or punching is done by specifying the punching of two blank columns.

The second punch stacker is an error stacker and is selected on punch errors. This stacker is program selectable. However, errors will cause this stacker to be selected regardless of program choice. Stacker selection is given for the card in the punch wait station in the same instruction that causes it to be punched.

(5) Check the condition code (CC) setting to determine if the instruction was accepted.

CC

00 = command accepted

10 = busy

11 = command rejected – invalid device address or device offline

(6) Store and test the status of the serial punch after the operation is completed to determine if the operation was successful.

Examples of the statements of steps (1) through (6) are:

LABEL	OPERATION	OPERAND	COMMENTS
	MVC	B(CW+1(3),CON	LOAD BCW ASSIGNED TO SERIAL PUNCH
	MVI	XF+1,X'02'	LOAD DEVICE ADDRESS INTO XIOF INSTRUCTION
	MVI	XF+3,X'05'	LOAD COMMAND BYTE INTO XIOF INSTRUCTION
	XIOF	X'02',X'05'	PUNCH A CARD IN IMAGE MODE
	BC	8,EXIT	IF CC=0, BRANCH TO EXIT
	TIO	4000(15),X'02'	CC≠00; STORE STATUS
			CODING FOR TESTING STATUS
CON	DC	X'50006000'	

When COS or NCOS is in use, the supervisor request call (SRC) instructs the supervisor to execute the instructions shown in the above example in the input/output program state.

(7) In case of an error, display the status byte and instruct operator to take appropriate action according to error recovery message. A status byte is stored in location 66 (DS) of low order main storage and contains information pertaining to the result of the last issued order or to the next-to-last issued order. The bits which comprise the status byte are designated in Figure 3–2 and described in Table 3–1.

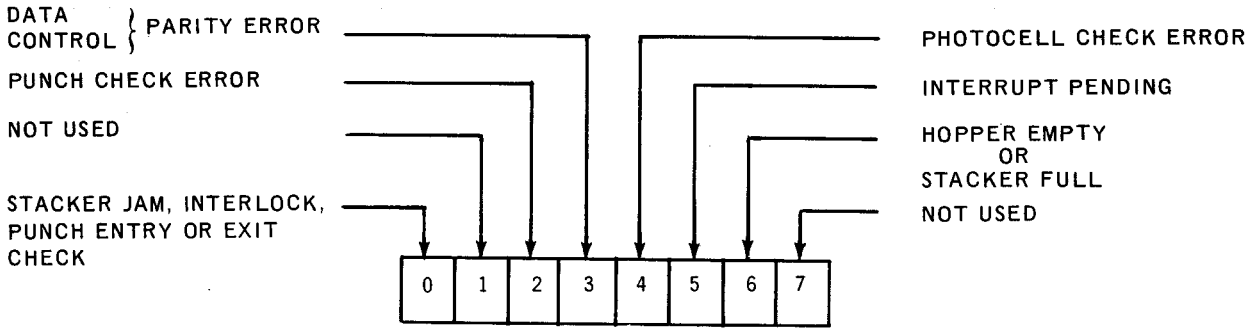


Figure 3–2. UNIVAC 0603–04 Card Punch, Status Byte Format

If the status byte contains all 0's, the function is performed as specified.

BIT POSITION	BIT DESIGNATION	DEFINITION
0	Stacker jam, interlock, punch entry, or exit check	Set to indicate stacker jam, interlock broken, punch entry, or exit check, or any other condition that may require operator intervention.
1	Not used	This bit position contains a 0.
2	Punch check error	Set to indicate that the last card punched was in error. An interrupt occurs, and the card being punched automatically goes to the reject stacker.
3	Data parity or control parity error	Set to indicate that a card at the read station or the punch station may be in error. An immediate interrupt occurs upon recognition of the error and the XIOF is terminated. A card passing through the punch station will automatically go to the reject stacker.
4	Photocell check error	Set to indicate read photocells error or possible card jam. This status bit is set at the end of an XIOF. The last XIOF issued should be assumed in error.
5	Interrupt request pending	Set only if the TIO instruction clears a pending interrupt before it is accepted. This status bit does not indicate an error.
6	Hopper empty or stacker full	Set to indicate an empty card hopper or a full stacker. Also indicates that the last XIOF instruction was terminated before it was executed. To recover from this early termination, the XIOF must be reissued after condition is corrected.
7	Not used	This bit position contains a 0.

Table 3–1. UNIVAC 0603–04 Card Punch, Status Byte Definition

(8) If all interrupts are inhibited (MOS in use), store the status and check for successful execution of the XIOF instruction before issuing the next XIOF. If an error exists, repeat step 7.

Example:

```
LABEL     OPERATION     OPERAND                    COMMENTS
1         10            16
LOOP      TIO           4,0,0,(1,5),,X'02'         STORE STATUS
          BC            3,,LOOP                    BRANCH TO LOOP IF CC=1,0 OR 1,1
          .                                        }
          .                                        } CODE FOR TESTING STATUS
          .                                        }
*         STATUS IS STORED; DISPLAY STATUS BYTE. IF ERROR EXISTS,
          INSTRUCT OPERATOR TO TAKE APPROPRIATE ACTION.
```

NOTE: Instructions are executed in processor program state.

(9) If interrupts are not inhibited, the program must provide wait instructions for an interrupt. Store the status byte by moving it from low-order reserved storage (device status (DS) byte, location 66_{10}). Check for successful execution of the previous XIOF instruction before issuing the next XIOF. If an error exists, repeat step 7.

Example:

```
LABEL     OPERATION     OPERAND                    COMMENTS
1         10            16
START     DS            CL1
          MVC           STAT,66(0)                 STORE STATUS BYTE
          .                                        }
          .                                        } TEST STATUS BYTE
          .                                        }
*         DISPLAY STATUS BYTE. IF ERROR EXISTS, INSTRUCT
          OPERATOR TO TAKE APPROPRIATE ACTION.
```

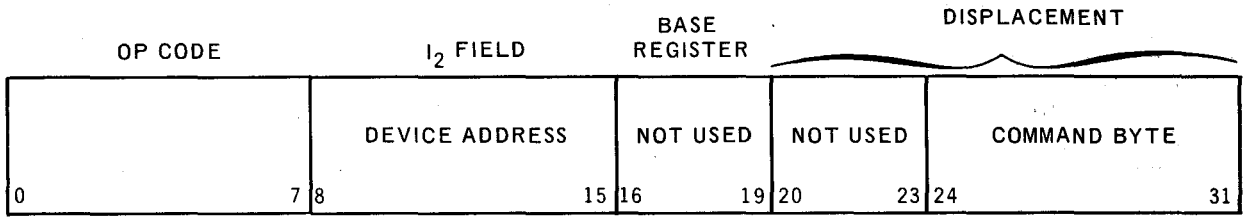
NOTE: Instructions are executed in input/output program state after the interrupt has occurred.

3.3. UNIVAC 0604 CARD PUNCH I/O OPERATION

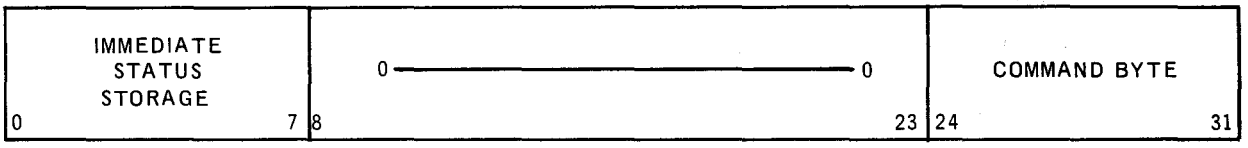
The following paragraphs provide information pertaining to the command, status, and sense data bytes utilized in all input/output transactions between the processor and the UNIVAC 0604 Card Punch.

The UNIVAC 0604 Card Punch responds to the command bytes listed in Table 3–2 and described in 3.3.1 through 3.3.1.7. The command byte is located in bit positions 24 through 31 of the execute-I/O (XIOF) instruction (UNIVAC 9200 II/9300/9300 II Systems) and the channel address word (UNIVAC 9400 System).

EXECUTE–I/O (XIOF) INSTRUCTION



CHANNEL ADDRESS WORD



3.3.1. Command Bytes

The following paragraphs discuss the command bytes issued to the row punch by the processor. All commands are checked for vertical parity and validity before being accepted by the row punch control unit. Table 3–2 summarizes the command bytes.

COMMAND BYTE	BIT POSITION
	0 1 2 3 4 5 6 7
TEST*	X X 1 1 0 0 0 0, or X X 0 0 0 0 0 0
SET INHIBIT STATUS	X X 0 1 0 0 0 0
RESET INHIBIT STATUS	X X 1 0 0 0 0 0
SENSE	X X X X 0 1 0 0
CONTROL (used for nondata transfer commands)	A X C D E X 1 1
LOAD BUFFER (write)	A X C D E F 0 1
UNLOAD BUFFER (read)	A X X X X F 1 0

NOTES:

(1) Bit position 7 is the least significant bit position.

(2) X is not used (disregarded by control unit).

(3) *Test-input/output-status (TIO) instruction is used with 9200 II/9300/9300 II Systems.

(4) A, C, D, E = detail bits for control command.

A = 0 denotes normal operation.

A = 1 indicates transfer postpunch read data to the punch buffer (this function is a maintenance feature only).

C = 1 functions to advance the cards one station (feed a card) and to place the card punched on the previous punch order into the select stacker.

D = 1 functions to feed and punch a card.

E = 1 functions to feed a card.

(5) A, C, D, E, F = detail bits for load-buffer command.

A = 0 functions to load the punch buffer.

A = 1 functions to load the read buffer (read buffer test function).

C, D, and E detail bits perform the same functions as described above for C, D, and E of the control command byte.

F = 0 functions to cause cards to be punched in compress mode.

F = 1 functions to cause cards to be punched in image mode.

Table 3–2. UNIVAC 0604 Card Punch, Command Bytes (Part 1 of 2)

(6) A, F = detail bits for unload-buffer command.

A = 0 functions to unload the read buffer.

A = 1 functions to unload the punch buffer.

F = 0 functions to read data punched in the compress mode.

F = 1 functions to read data punched in the image mode.

(7) The following command bytes are invalid and will be rejected by the control unit:

| X | X | X | X | 1 | X | 0 | 0 |

and

| X | X | 0 | 0 | 0 | X | 1 | 1 |

Table 3-2. UNIVAC 0604 Card Punch, Command Bytes (Part 2 of 2)

3.3.1.1. Test

| X | X | 1 | 1 | 0 | 0 | 0 | 0 |

or

| X | X | 0 | 0 | 0 | 0 | 0 | 0 |

24 31

X = Not used (disregarded by control unit)

The test command tests the status of the unit specified by the address field of the start-I/O instruction (UNIVAC 9400 System) or the TIO instruction (UNIVAC 9200 II/9300/9300 II Systems). A status byte is returned to the multiplexer channel at the conclusion of the initial selection sequence. If the control unit is busy, this command is aborted and a busy condition status byte is sent to the channel by the control unit busy sequence. If the control unit is not busy, the status byte will indicate that the addressed unit is ready for operation or will contain the store status byte from a previous operation.

3.3.1.2. Set Inhibit Status

| X | X | 0 | 1 | 0 | 0 | 0 | 0 |

24 31

X = Not used (disregarded by control unit)

The set-inhibit-status command performs the same function as the test command (see 3.3.1.1). In addition, it sets the inhibit-status-in condition at a control unit to prevent the control unit from generating a request sequence signal for presenting a status byte to the multiplexer channel. This condition remains set until cleared by a system reset or selective reset operation, or by a reset-inhibit-status command (see 3.3.1.3).

3.3.1.3. Reset Inhibit Status

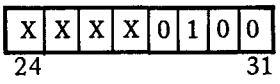
| X | X | 1 | 0 | 0 | 0 | 0 | 0 |

24 31

X = Not used (disregarded by control unit)

The reset-inhibit-status command performs the same function as the test command (see 3.3.1.1). In addition, it clears any inhibit-status-in condition (see 3.3.1.2) at the control unit.

3.3.1.4. Sense



X = Not used (disregarded by control unit)

The sense command requests the control unit to test and store the current status of the card punch then to transfer sense data bytes (see 3.3.3) to the multiplexer channel. These bytes describe the current status of the control unit and any unusual conditions encountered in its last operation. The bytes are sent in data transfer sequence at the nominal data rate. The end status byte contains the status as a result of the sense command operation; normally only the channel end and device end bits are set in this status byte (see 3.3.2).

3.3.1.5. Load Buffer

X = Not used (disregarded by control unit)

The load-buffer (write) command contains instructions which:

- specify whether the read buffer or punch buffer is to be loaded;
- specify to the control unit the mode of translation in which cards are to be punched;
- issue feed and punch orders.

Loading the read buffer is a buffer test function. When the load-buffer command specifies that the read buffer is to be loaded, detail bit D should not be a 1 (see Table 3–2). The mode of translation specified by the load-buffer command is either the image mode or the compress mode (see 3.3.1.5.1, 3.3.1.5.2, and Table 3–2). If the image mode is specified, data transfer time from the multiplexer channel to the punch buffer should not exceed 320 microseconds per byte.

The load-buffer command also contains feed and punch orders for the control unit. Once these orders are accepted, loading of the punch buffer occurs. However, the punch buffer may be loaded before issuing a feed and punch order. When this occurs, the control command (see 3.3.1.7) normally is issued following a load-buffer command. At the completion of the data transfer, the control unit returns an end status byte to the multiplexer channel.

3.3.1.5.1. Image Mode

Punching cards in the image mode of translation permits two bytes of binary data to be represented on each column of an 80-column card (see Figures 3–3 and 3–4). To punch two bytes on one column, the two most significant bits of each data byte received from the multiplexer channel are disregarded. The remaining least significant six bits of each byte are punched on half of a column. For example, data byte 1 would be represented on the lower half of a column, and data byte 0 would be represented on the upper half of the same column. Thus, 80 columns of a punched card can represent a total of 160 bytes of binary data. A card punched in the image mode contains a one-to-one representation of binary data (that is, a punch represents a binary 1 and no punch represents a binary 0).

If a prepunch read feature is used, one column of data read is assembled from two six-bit bytes into two eight-bit bytes. To present a complete eight-bit byte, zeros are added to bits 0 and 1.

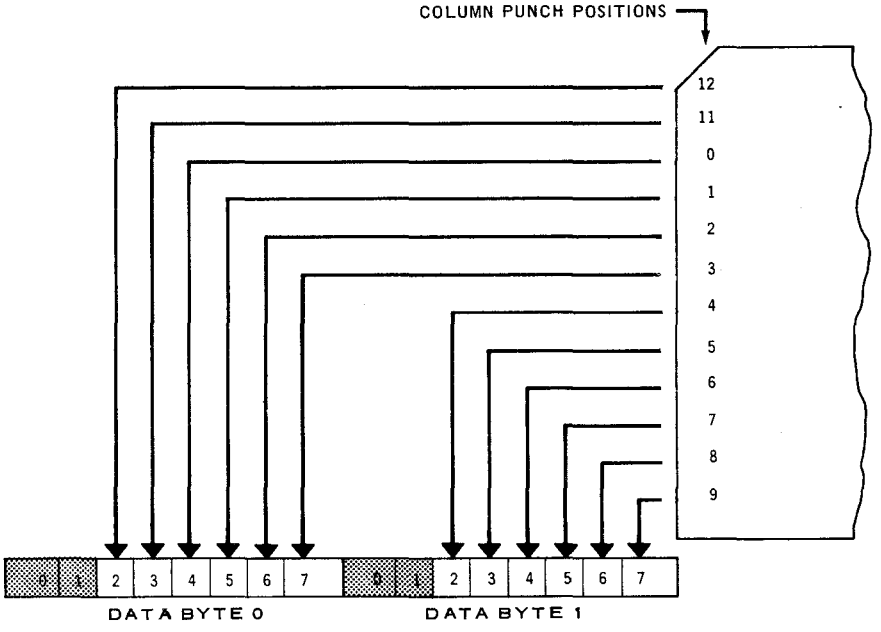


Figure 3–3. Image Mode Translation

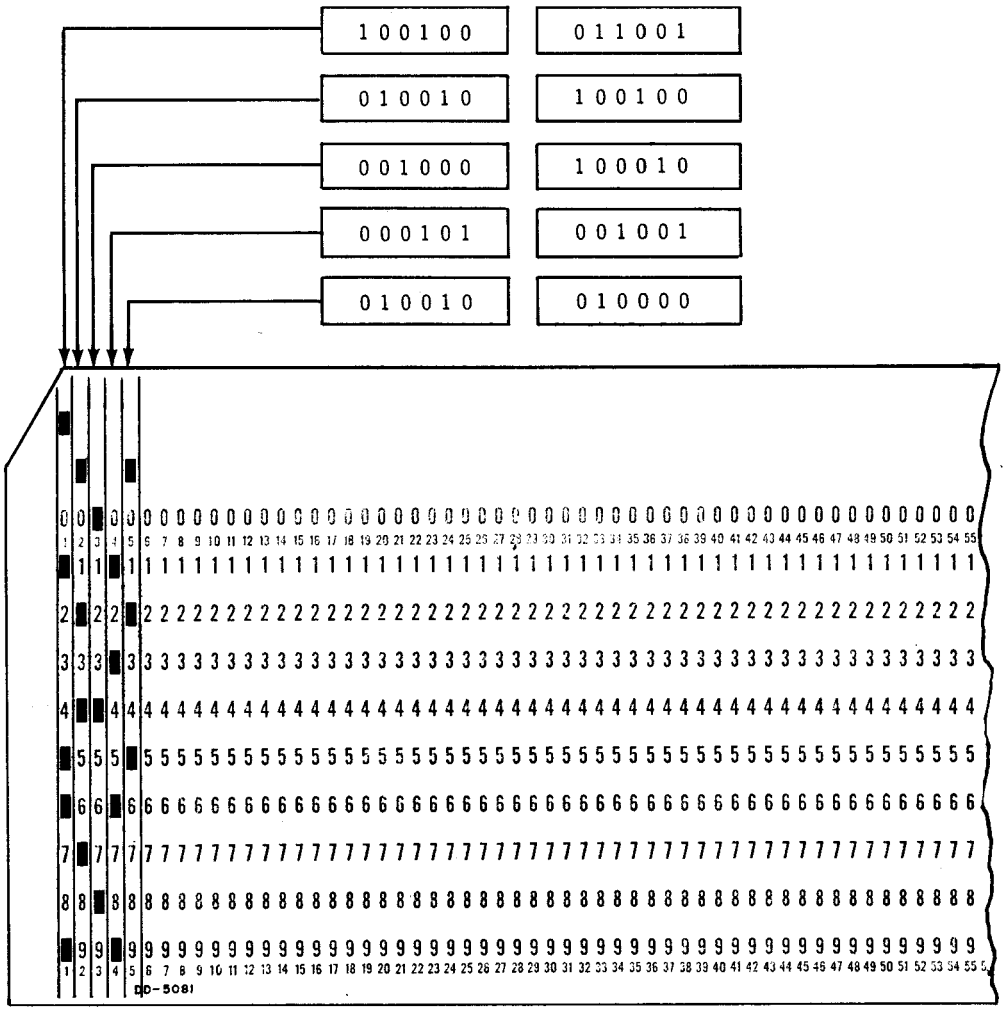
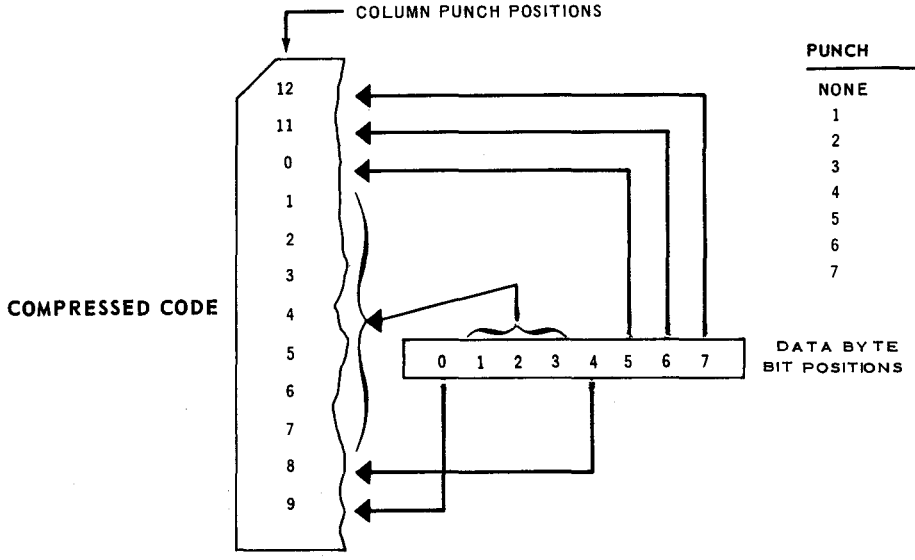


Figure 3–4. Relationship of Data Bytes to Card Punches (Image Mode)

3.3.1.5.2. Compress Mode

In the compress mode of translation, one eight-bit byte is punched on each column of a card (see Figures 3–5 and 3–6 and Table 3–3). Each byte represents a hexadecimal character that is composed of two digits. Of the eight-bit byte, bits 0, 1, 2, and 3 are the binary representation of the most significant hexadecimal digit; bits 4, 5, 6, and 7 are the binary representation of the least significant hexadecimal digit. As a result, any column of a card can represent any hexadecimal character from 00 through FF.



NOTE: PUNCH POSITIONS 1 THROUGH 7 ARE INDICATED IN BITS 1 THROUGH 3 ACCORDING TO THE ABOVE TABLE.

Figure 3–5. Compress Mode Transmission

When punching cards in the compress mode, a single punch in any one of rows 1 through 7 represents bits 1, 2, and 3 of the data byte (see Figure 3–5). All other rows are a direct binary representation (for example, a logical 1 in bit position 7 of a data byte is represented by a punch in row 12). If more than one punch per column is present in rows 1 through 7, the codes that these punches represent will be OR gated when the column is read. For example, a punch in row 2 and in row 5 of a column is read as a punch in row 2.

	COMPRESSED CARD CODE PUNCHES		BINARY REPRESENTATION	HEXADECIMAL REPRESENTATION	EBCDIC CODE EQUIVALENT
DATA BYTE 1	9,7	0	1 1 1 0 0 1 0 0	E4	U
DATA BYTE 2	9,2	12,0	1 1 0 1 0 1 0 1	D5	N
DATA BYTE 3	9,5	8,12	1 1 0 0 1 0 0 1	C9	I
DATA BYTE 4	9,7	12,0	1 1 1 0 0 1 0 1	E5	V
DATA BYTE 5	9,5	12	1 1 0 0 0 0 0 1	C1	A
DATA BYTE 6	9,5	12,11	1 1 0 0 0 0 1 1	C3	C

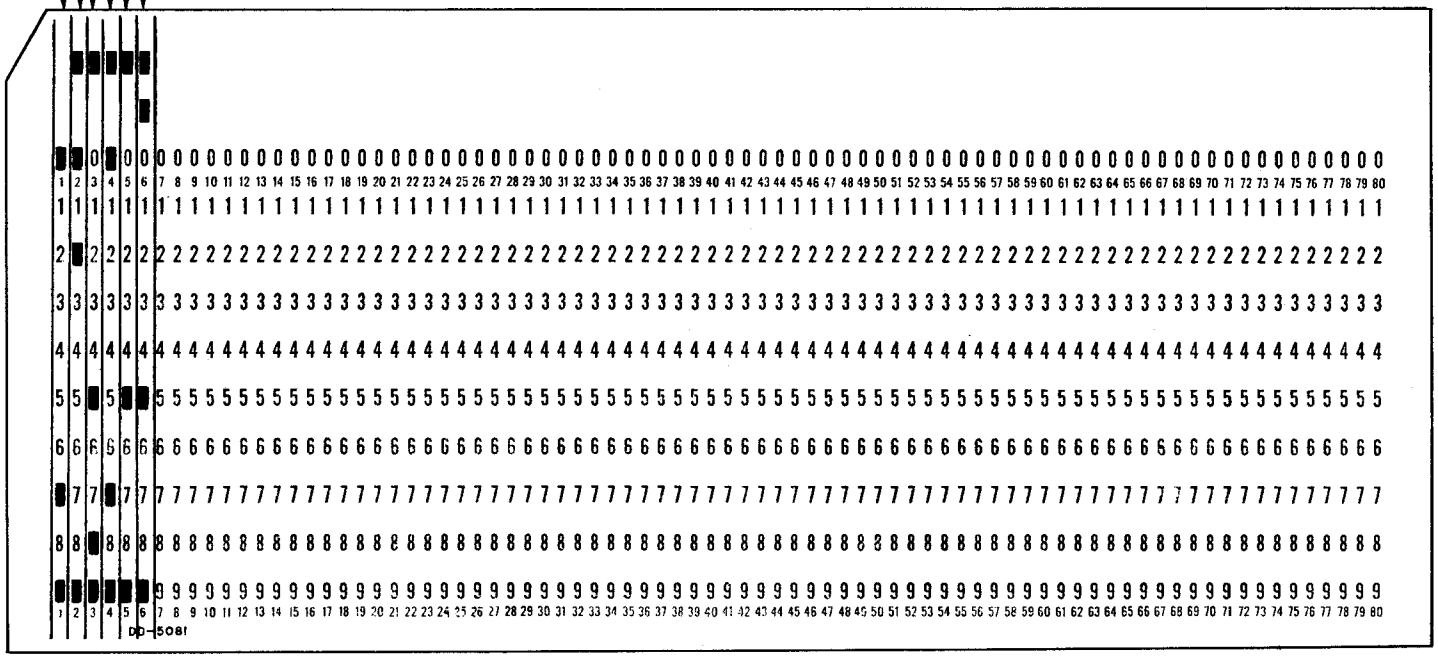


Figure 3–6. Relationship of Data Bytes to Card Punches (Compress Mode)

DECIMAL	HEXA-DECIMAL	EBCDIC	COMPRESSED CARD CODE *
0	00	NUL	NO PUNCH
1	01		3 12
2	02		4 11
3	03		1 12,11
4	04	PF	5 0
5	05	HT	2 12,0
6	06	LC	7 11,0
7	07	DEL	6 12,11,0
8	08		9 8
9	09		9,3 8,12
10	0A		9,4 8,11
11	0B		9,1 8,12,11
12	0C		9,5 8,0
13	0D		9,2 8,12,0
14	0E		9,7 8,11,0
15	0F		9,6 8,12,11,0
16	10		↑ ↑
17	11		
18	12		
19	13		BIT BIT
20	14	RES	
21	15	NL	
22	16	BS	POSITIONS 0, 1, 2, 3. POSITIONS 4, 5, 6, 7
23	17	IL	
24	18		
25	19		
26	1A		
27	1B		
28	1C		
29	1D		
30	1E		
31	1F		
32	20	DS	
33	21	SOS	
34	22	FS	
35	23		
36	24	BYP	
37	25	LF	
38	26	EOB	
39	27	PRE	
40	28		
41	29		
42	2A	SM	
43	2B		
44	2C		
45	2D		
46	2E		
47	2F		
48	30		
49	31		
50	32		
51	33		
52	34	PN	
53	35	RS	
54	36	UC	
55	37	EOT	
56	38		
57	39		
58	3A		
59	3B		
60	3C		
61	3D		
62	3E		
63	3F		
64	40	SP	
65	41		
66	42		
67	43		
68	44		
69	45		
70	46		
71	47		
72	48		
73	49		
74	4A	¢	
75	4B	.	
76	4C	<	
77	4D	(	
78	4E	+	
79	4F	| (Vert. Bar)	
80	50	&	
81	51		
82	52		
83	53		
84	54		
85	55		

DECIMAL	HEXA-DECIMAL	EBCDIC
86	56	
87	57	
88	58	
89	59	
90	5A	!
91	5B	$
92	5C	*
93	5D	)
94	5E	;
95	5F	¬ (Not)
96	60	–
97	61	/
98	62	
99	63	
100	64	
101	65	
102	66	
103	67	
104	68	
105	69	
106	6A	\
107	6B	, (Comma)
108	6C	%
109	6D	_ (Underscore)
110	6E	>
111	6F	?
112	70	
113	71	
114	72	
115	73	
116	74	
117	75	
118	76	
119	77	
120	78	
121	79	
122	7A	:
123	7B	#
124	7C	@
125	7D	' (Prime or Apos)
126	7E	=
127	7F	" (Quotes)
128	80	
129	81	a
130	82	b
131	83	c
132	84	d
133	85	e
134	86	f
135	87	g
136	88	h
137	89	i
138	8A	
139	8B	
140	8C	
141	8D	
142	8E	
143	8F	
144	90	
145	91	j
146	92	k
147	93	l
148	94	m
149	95	n
150	96	o
151	97	p
152	98	q
153	99	r
154	9A	
155	9B	
156	9C	
157	9D	
158	9E	
159	9F	
160	A0	
161	A1	
162	A2	s
163	A3	t
164	A4	u
165	A5	r
166	A6	w
167	A7	x
168	A8	y
169	A9	z

DECIMAL	HEXA-DECIMAL	EBCDIC
170	AA	
171	AB	
172	AC	
173	AD	
174	AE	
175	AF	
176	B0	
177	B1	
178	B2	
179	B3	
180	B4	
181	B5	
182	B6	
183	B7	
184	B8	
185	B9	
186	BA	
187	BB	
188	BC	
189	BD	
190	BE	
191	BF	
192	C0	PZ
193	C1	A
194	C2	B
195	C3	C
196	C4	D
197	C5	E
198	C6	F
199	C7	G
200	C8	H
201	C9	I
202	CA	
203	CB	
204	CC	
205	CD	
206	CE	
207	CF	
208	D0	MZ
209	D1	J
210	D2	K
211	D3	L
212	D4	M
213	D5	N
214	D6	O
215	D7	P
216	D8	Q
217	D9	R
218	DA	
219	DB	
220	DC	
221	DD	
222	DE	
223	DF	
224	E0	
225	E1	
226	E2	S
227	E3	T
228	E4	U
229	E5	V
230	E6	W
231	E7	X
232	E8	Y
233	E9	Z
234	EA	
235	EB	
236	EC	
237	ED	
238	EE	
239	EF	
240	F0	0
241	F1	1
242	F2	2
243	F3	3
244	F4	4
245	F5	5
246	F6	6
247	F7	7
248	F8	8
249	F9	9
250	FA	
251	FB	
252	FC	
253	FD	
254	FE	
255	FF	

*Punch patterns for bit positions 0, 1, 2, and 3 are used to store the most significant hexadecimal digit; punch patterns for bit positions 4, 5, 6, and 7 are used to store the least significant hexadecimal digit.

Table 3–3. Compress Mode Card Code to Processor Code

3.3.1.6. Unload Buffer

A	X	X	X	X	F	1	0
24							31

X = Not used (disregarded by control unit)

The unload-buffer (read) command causes the control unit to transfer data to the multiplexer channel. Data bytes stored in the read buffer after a card-punch-and-check operation are transferred to the multiplexer channel. Data bytes stored in the punch buffer are transferred to the multiplexer channel to provide buffer memory and translator tests. If the prepunch read feature is used, data bytes are stored in the read buffer and transferred to the multiplexer channel. At the completion of the data transfer, an end status byte is presented to the multiplexer channel.

To maintain the 200-card-per-minute rate when the read/punch feature is in use, the data transfer resulting from the unload-buffer command must be completed within 25 milliseconds after a status byte (with the device end bit set) is presented. If the 250-card-per-minute feature is in use, the allowable time for transferring data is reduced; therefore, functions of the unload-buffer command must be completed within 20 milliseconds.

Data transferred to the multiplexer channel is in either the compress mode or the image mode of translation. If cards are punched in the image mode, the transfer time between the read buffer and the processor via the multiplexer channel should not exceed 230 microseconds per byte. This transfer time is necessary for maintaining the 200-card-per-minute rate. If the 250-card-per-minute feature is in use, the transfer time should not exceed 125 microseconds per byte.

3.3.1.7. Control

A	X	C	D	E	X	1	1
24							31

X = Not used (disregarded by control unit)

The control command is used for operations which do not require the transfer of data. Detail bits of this command can direct the card punch to perform any of the operations defined in Table 3–2 for the control command.

3.3.2. Status Byte

A status byte is stored in the upper eight bits (0 through 7) of the subchannel control word (UNIVAC 9400 System)

SUBCHANNEL CONTROL WORD

DEVICE STATUS	FLAGS	BUFFER CONTROL WORD INDEX
0 7	8 14	15 31

or in the low order main storage location DS (UNIVAC 9200 II/9300/9300 II Systems). This byte supplies information concerning conditions of the last operation. A status byte is presented to the multiplexer channel at the following times:

- the end of the initial selection sequence;
- the completion of data transfer resulting from either a load-buffer or an unload-buffer command;
- the completion of card motion cycles.

The status register will be cleared when the presented status byte is accepted by the multiplexer channel. The bits which comprise the status byte are designated in Figure 3–7 and described in Table 3–4.

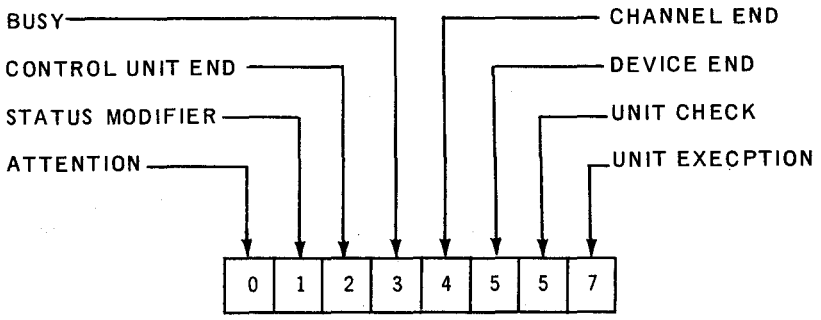


Figure 3–7. UNIVAC 0604 Card Punch, Status Byte Format

BIT POSITION	BIT DESIGNATION	DEFINITION
0	Attention	Not used. This bit position contains a 0.
1	Status modifier	Set to indicate that the subsystem is being used. The status modifier bit and busy bit are both set to indicate that the subsystem is busy.
2	Control unit end	Not used. This bit position contains a 0.
3	Busy	Set only during initial selection sequence to indicate that the control unit cannot accept a command because of one of the following conditions: ■ The busy bit and status modifier bit are set to indicate that the control unit is executing a previously initiated operation. ■ The control unit has pending status conditions from a previous command. If test command or TIO instruction is received, busy bit will not be set.
4	Channel end	Set for the following conditions: ■ To indicate the completion of data transfers resulting from a punch (load-buffer or control) command. ■ To indicate acceptance of a control command. ■ The channel end bit is set with the device end bit to indicate the completion of data transfers resulting from an unload-buffer command. ■ To indicate early termination due to a data parity error.

Table 3–4. UNIVAC 0604 Card Punch, Status Byte Definition (Part 1 of 2)

BIT POSITION	BIT DESIGNATION	DEFINITION
5	Device end	Set to indicate the completion of a command initiated by the multiplexer channel. The device end bit is set for the following conditions: ■ To indicate the completion of a punch (load-buffer or control) command. ■ To indicate the completion of a control command. ■ The device end bit is set with the channel end bit to indicate the completion of data transfers resulting from an unload-buffer command. ■ To indicate occurrence of an error which will prevent the completion of the punch command.
6	Unit check	Set for the following conditions: ■ To indicate a parity error in the command byte or an invalid command. ■ To indicate that the card punch was not ready when a test command or a TIO instruction was presented during the initial selection sequence. The test command or TIO instruction is rejected and no end status byte is generated. ■ To indicate that a bit was set in sense data byte 1.
7	Unit exception	Unit exception bit is set with device end bit to indicate a hole count error on the previous card punch cycle.

Table 3–4. UNIVAC 0604 Card Punch, Status Byte Definition (Part 2 of 2)

3.3.3. Sense Data Bytes

Sense data bytes are stored data sent in response to a sense command (see 3.3.1.4). The data describes unusual conditions detected in the last operation and the current status of the UNIVAC 0604 Card Punch. Sense data bits that are set as a result of error or fault conditions during an operation remain set until cleared by initiation of a new command (except for a sense or test command). Execution of a sense or test command causes additional sense data bits to change state and thereby to modify the sense data byte. Sense data bits that indicate current status of the UNIVAC 0604 Card Punch change if the current status changes. See Figure 3–8 and Table 3–5 for a definition of sense data bytes 1 and 2.

SENSE DATA BYTE 1

BIT 0	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7
COMMAND REJECT	INTER-VENTION REQUIRED	BUS OUT CHECK	CARD JAM	DATA CHECK	DATA LATE	NONREPEAT ABNORMAL	INHIBIT STATUS IN

SENSE DATA BYTE 2

BIT 0	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7
NOT USED	NOT USED	HCRPA	HCRPB	HCRPC	HCR2A	HCR2B	HCR2C

Figure 3–8. UNIVAC 0604 Card Punch, Sense Data Bytes

BIT POSITION	BIT DESIGNATION	DEFINITION
SENSE DATA BYTE 1		
0	Command reject	Set when an unspecified command is issued. If an incorrect parity is detected during the transfer of the command code, this bit is suppressed. Neither the channel end bit (4) nor device end bit (5) in the status byte is set for this condition.
1	Intervention required	Set to indicate that an abnormal condition (other than a hole count error) was detected during the previous operation. In all cases, the error requires manual intervention to correct (that is, empty hopper, stacker full, and so on).
2	Bus out check	Set when a command byte or data byte parity error is detected during the initial selection sequence. A command byte parity error causes the control unit to terminate operation immediately and to suppress the invalid command. If the control unit is holding a pending status when the command byte parity error is detected, the command byte is disregarded and the stored status is transferred to the multiplexer channel during the status transfer sequence. Neither the channel end nor the device end bit in the status byte is set for this condition. A data byte parity error during data transfer causes the control unit to terminate operation immediately. Channel end bit and device end bit in the status byte are set.
3	Card jam	Set to indicate that a card transport error has occurred. A jam in the prepunch station causes the A JAM half of the MAN FEED A JAM/B JAM switch/indicator on the operator's control panel to light. A jam in the postpunch station causes the B JAM half of the MAN FEED A JAM/B JAM switch/indicator on the operator's control panel to light.

Table 3–5. UNIVAC 0604 Card Punch, Sense Data Bytes, Bit Definition (Part 1 of 2)

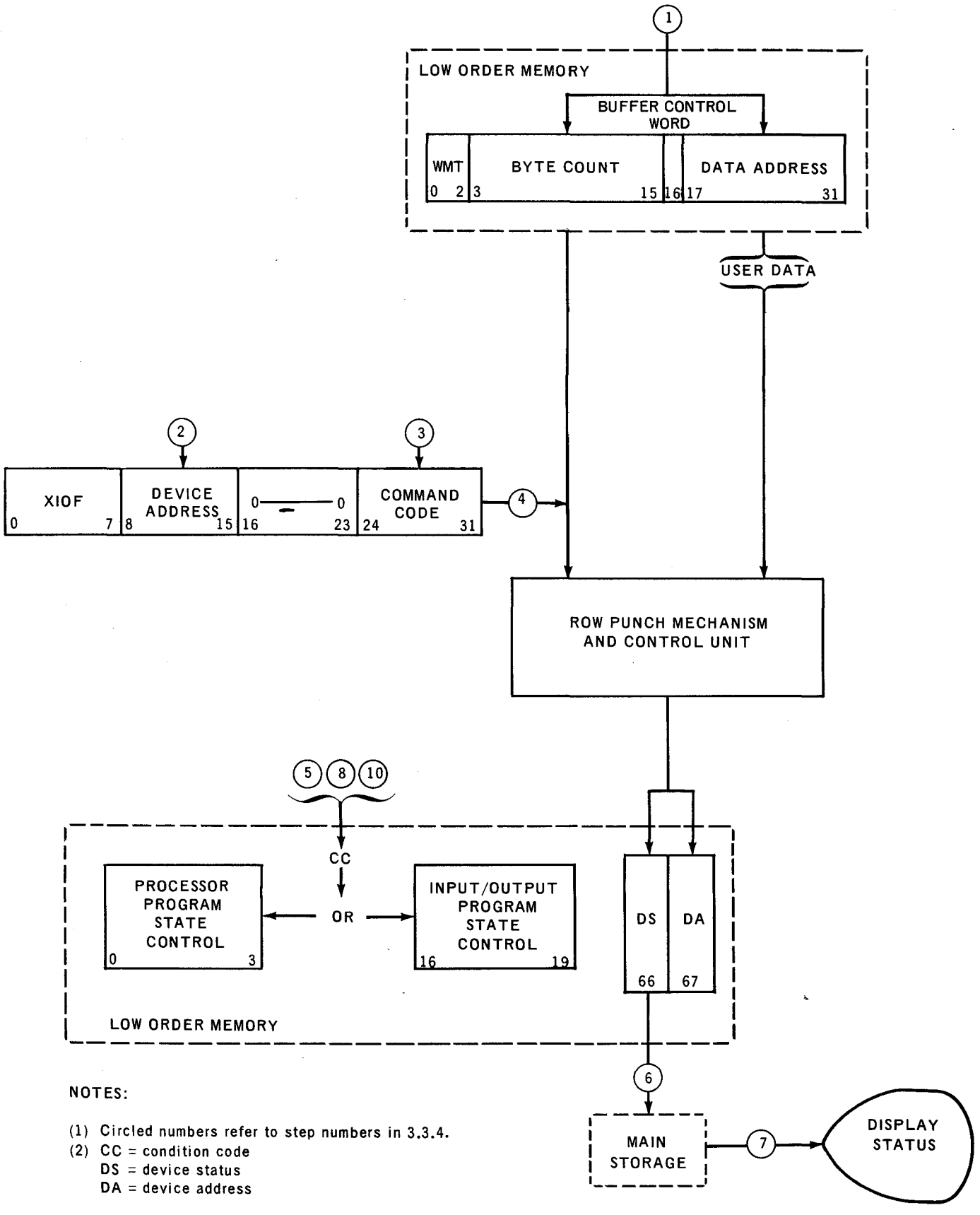
BIT POSITION	BIT DESIGNATION	DEFINITION
4	Data check	Not used. This bit position contains a 0.
5	Data late	Set to indicate that the punch buffer has not been loaded prior to the initiation of a punch operation.
6	Nonrepeat abnormal	Set to indicate that one of the following conditions was detected during the previous operation: ■ Hopper empty ■ Chips (box full or not in place) ■ Stacker jam ■ Stacker full ■ Interlock error
7	Inhibit	Set to indicate that the inhibit status bit in flip-flop is set (prevent a status byte from being sent to the multiplexer channel).
SENSE DATA BYTE 2		
0	Not used	Sense data byte 2 is used for maintenance analysis functions in conjunction with failure routines. The status of various flip-flops in the control unit is transferred to this byte.
1	Not used	
2	HCRPA Punch	
3	HCRPB hole	
4	HCRPC counter	
5	HCR2A Postread	
6	HCR2B hole	
7	HCR2C counter	

Table 3–5. UNIVAC 0604 Card Punch, Sense Data Bytes, Bit Definition (Part 2 of 2)

3.3.4. UNIVAC 9200 II/9300/9300 II Systems I/O Operation

When the concurrent operating system (COS) or the nonconcurrent operating system (NCOS) is in use, the execution of commands for the row punch occurs only when the processor is in the input/output program state. When the minimum operating system (MOS) is in use, all commands for the row punch are executed in the processor program state. A knowledge of the *UNIVAC 9200/9300 System Central Processor and Peripherals Programmers Reference, UP-7546* (current version) is necessary when using the information that follows.

Figure 3–9 shows the functional relationship between the processor and the row punch.



NOTES:

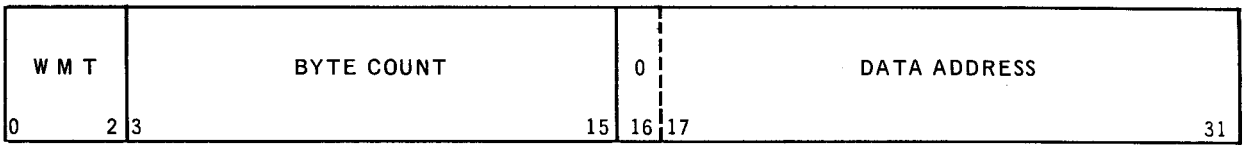
(1) Circled numbers refer to step numbers in 3.3.4.
(2) CC = condition code
DS = device status
DA = device address

Figure 3–9. Command Function Relationship Between UNIVAC 9200 II/9300/9300 II Systems and 0604 Card Punch

Listed below are the initial steps required to execute a command for the UNIVAC 0604 Card Punch by use of the UNIVAC 9200 II/9300/9300 II Systems.

(1) Load the proper buffer control word (four bytes).

BUFFER CONTROL WORD



where:

W = data direction bit

W = 1 for load buffer or write operations

W = 0 for unload buffer or read operations

M = addressing mode bit

M = 0 for all card punch operations (under hardware control)

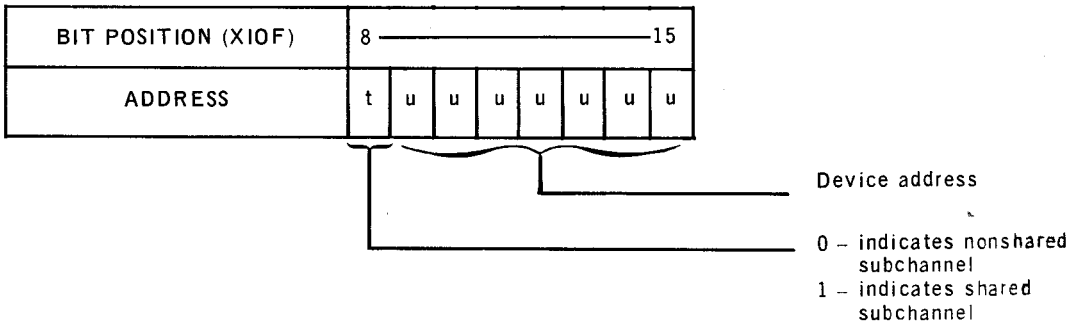
T = termination bit

T = 0 for all card punch operations (under hardware control)

Byte Count – specifies the number of bytes required for load-buffer (write) operations or for unload-buffer (read) operations. For unload-buffer operations, the number of bytes specified must be 160 (image mode) or 80 (compress mode). For load-buffer operations, the number of bytes specified may be less than 160 (image mode) or less than 80 (compress mode).

Data Address – used by the multiplexer channel as the storage address for the current byte of data. When operations terminate, this field indicates the storage address where the next byte would have gone or come from had the operation continued.

(2) Load the device address into bit positions 8 through 15 of the XIOF instruction.



(3) Load the command byte into bit positions 24 through 31 of the XIOF instruction (see Table 3–2 and 3.3.1).

(4) Issue XIOF instruction.

NOTE: If the MOS is used, all interrupts must be inhibited. The XIOF instruction with the set-inhibit-status command must therefore be executed first. Interrupts must not be inhibited if the COS or NCOS is used.

(5) Check the condition code (CC) setting to determine if the instruction was accepted.

CC

00 = command accepted

01 = command rejected

10 = busy

11 = not operational

(6) Store and test the status of the row punch after the operation is completed to determine if the operation was successful.

```
LABEL    OPERATION    OPERAND                  COMMENTS
1        10           16
         MVC          BCW(4),CON               LOAD BCW ASSIGNED TO ROW PUNCH
         MVI          XF+1,X'06'               LOAD DEVICE ADDRESS INTO XIOF INSTRUCTION
         MVI          XF+3,X'05'               LOAD COMMAND BYTE INTO XIOF INSTRUCTION
XF       XIOF         X'06',X'05'              PUNCH A CARD IN IMAGE MODE
         BC           8,EXIT                   IF CC=00, BRANCH TO EXIT
         BC           1,ERR                    IF CC=11, BRANCH TO ERROR DISPLAY ROUTINE
         BC           2,XF                     IF CC=10, BRANCH TO XF
         TIO          4000(15),X'02'           CC=01; STORE STATUS
         .                                     }
         .                                     } TEST STATUS BYTE
         .                                     }
         .
CON      DC           X'80EE1A'
```

When COS or NCOS is in use, the supervisor request call (SRC) instructs the supervisor to execute the instructions shown in the above example in the input/output program state.

(7) In case of an error, display the status byte and instruct operator to take appropriate action according to error recovery message.

If the status byte is displayed (MOS in use) and corrective action requires the operator to press the CLEAR HOPPER/CHIPS switch/indicator on the row punch control panel, the circuits of the row punch are cleared. The next XIOF instruction must contain the set inhibit status command to ensure that all interrupts are inhibited.

(8) If all interrupts are inhibited (MOS in use), check for successful execution of the XIOF instruction before issuing the next XIOF. If an error exists, repeat step (7).

Example:

```
LABEL      OPERATION      OPERAND                    COMMENTS
1          10             16
LOOP       TIO            4000(15),X'0002'           STORE STATUS
           BC             3,LOOP                     BRANCH TO LOOP IF CC=10 OR 11
                                                     CODING FOR TESTING STATUS
*          STATUS IS STORED; DISPLAY STATUS BYTE. IF ERROR EXISTS,
           INSTRUCT OPERATOR TO TAKE APPROPRIATE ACTION.
```

NOTE: Instructions are executed in processor program state.

(9) If interrupts are not inhibited (COS or NCOS in use), the program must provide wait instructions for an interrupt. When the interrupt occurs, status will then be stored.

(10) Check status byte for error condition after first interrupt occurs; display status byte if error condition exists. If error condition does not exist, the program must wait for second interrupt. After second interrupt occurs (device end bit set), test status byte for errors (see 3.3.2).

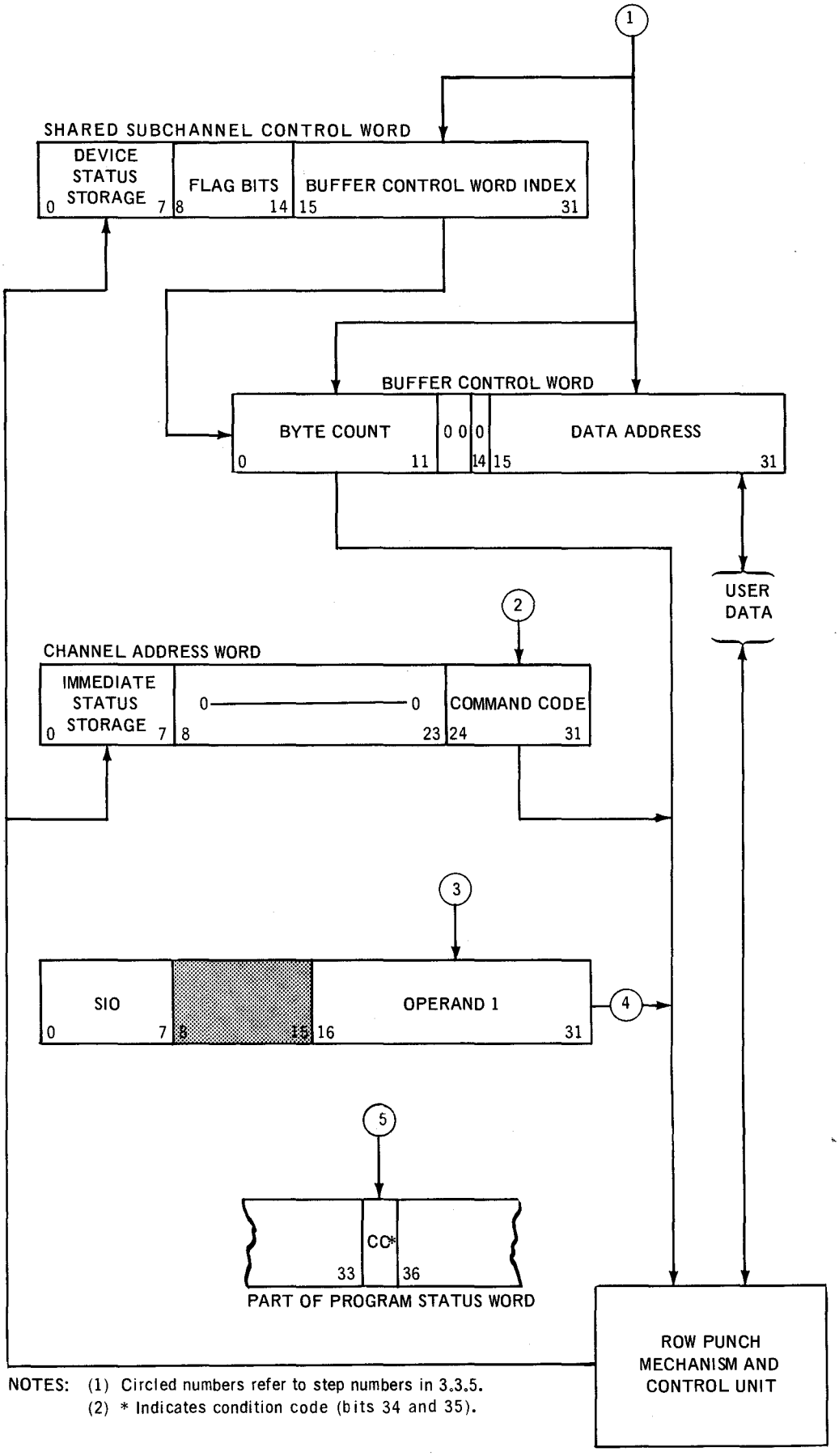
Example:

```
LABEL      OPERATION      OPERAND                    COMMENTS
1          10             16
           MVC            4000(11),66                STORE STATUS BYTE
                                                     TEST STATUS BYTE
*          DISPLAY STATUS BYTE. IF ERROR EXISTS, INSTRUCT
           OPERATOR TO TAKE APPROPRIATE ACTION.
```

NOTE: Instructions are executed in input/output program state after the interrupt has occurred.

3.3.5. UNIVAC 9400 System I/O Operation

The initial steps required to execute a command for the UNIVAC 0604 Card Punch by use of the UNIVAC 9400 System are listed here. Figure 3–10 illustrates the functional relationship between the processor and the row punch when a command is to be executed. It should be noted that this operation is privileged (can be executed only in the supervisor mode) and is utilized only when writing I/O dispatcher routines which are in the supervisor. A knowledge of the *UNIVAC 9400 System Supervisor Programmers Reference, UP-7689* (current version) and the *UNIVAC 9400 Assembler Central Processor Unit Programmers Reference, UP-7600* (current version) is necessary when using the information that follows.

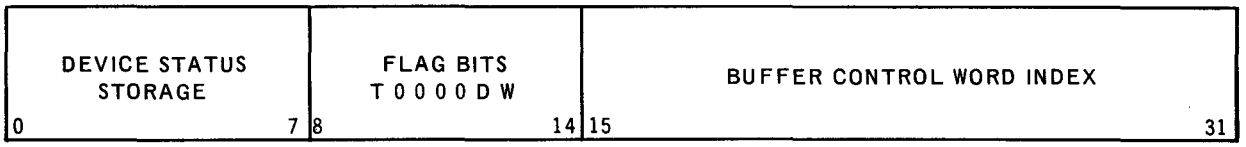


NOTES: (1) Circled numbers refer to step numbers in 3.3.5.
(2) * Indicates condition code (bits 34 and 35).

Figure 3-10. Command Function Relationship Between UNIVAC 9400 Processor and 0604 Card Punch

(1) Load the subchannel control word (SCW) and buffer control word (BCW).

SHARED SUBCHANNEL CONTROL WORD



Device status storage: Device status storage on interrupts for shared subchannel.

Flag bits: Terminate flag (T bit):

Upon completion of the input/output operation using the BCW, the multiplexer channel sets the T flag to 1 when either a stop-operation code (00) or a stop-immediate code (11) is contained in the current BCW. Normal completion of a BCW is caused when the byte count is decremented from 1 to 0.

Decrement flag (D bit):

If the D flag is a 1 bit, the data address is decremented by 1 for each data byte transferred.

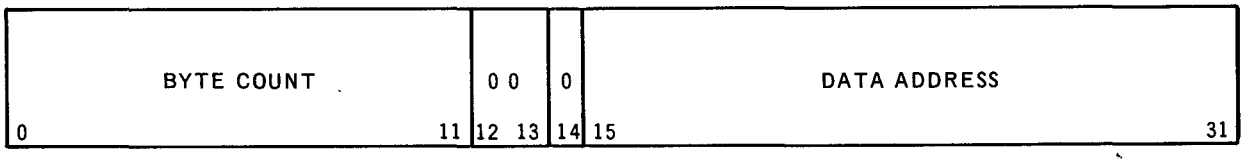
If the D bit is a 0 bit, the data address is incremented by 1 for each data byte transferred.

Write output flag (W bit):

If the W flag is a 1 bit, it specifies an output operation to the channel.

If the W flag is a 0 bit, it specifies an input operation to the channel.

SHARED BUFFER CONTROL WORD



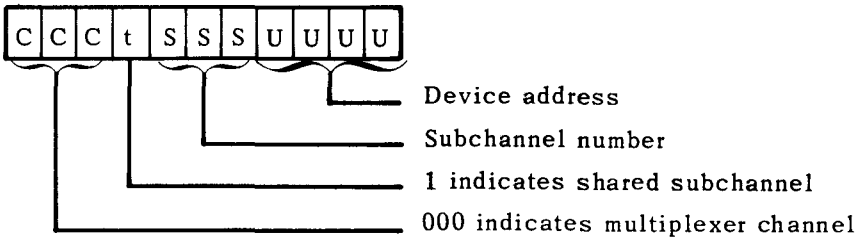
Byte count: Specifies the number of bytes required for load-buffer (write) operations or for unload-buffer (read) operations. For unload-buffer operations, the number of bytes specified must be 80 (image mode) or 160 (compress mode). For load-buffer operations, the number of bytes specified may be less than 80 (image mode) or 160 (compress mode).

Bits 12, 13, 14: Must be zero.

Data address: Used by the multiplexer channel as the storage address for the current byte of data. When operations terminate, this field indicates the storage address where the next byte would have gone or come from had the operation continued.

(2) Load the command byte into bit positions 24 through 31 of the channel address word (CAW). (See Table 3–2)

(3) Load the channel and device address into operand 1 portion of start-I/O (SIO) instruction.



(4) Issue SIO instruction. After the SIO instruction is issued, the row punch control unit accesses the SCW and CAW, which are fixed locations in main storage. The SCW contains the buffer control word index (BCWI) which points to the BCW.

(5) Test condition code stored in bit positions 34 and 35 of the program status word. The condition code is used for decision making. Each possible resulting condition code, along with its meaning, is:

00 – normal completion

01 – status stored

10 – channel busy

11 – not operational

UP-7772