TEKTRONIX®

21 AND 31 CALCULATOR

INTERFACING INFORMATION

INSTRUCTION MANUAL

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97005

Serial Number ____________________

070-1695-00

First Printing JULY 1974

DISCLAIMER

This manual documents the interfacing requirements for the TEKTRONIX 21 and 31 Calculators. Tektronix believes that the information contained herein is accurate and complete to the extent necessary for designing an interface in normal interfacing applications. However, Tektronix assumes no liability for interface circuitry which has not been designed and built by Tektronix, Inc. Furthermore, any damage to TEKTRONIX calculators and/or peripherals which results from the use of non-Tektronix designed interfaces will nullify the warranty on such equipment.

Specifications and price change privileges reserved.

TABLE OF CONTENTS

Fig. 1-0. TEKTRONIX 21 and 31 Calculators.

Section 1

GENERAL

DOCUMENTATION NOTE

This manual provides basic interfacing information on the TEKTRONIX 21 and 31 Calculators. The discussions are directed toward the design engineer using the manual as a reference while designing an interface to his peripheral device. The technician will also find the manual a good source of information on calculator operations.

The term "interface" refers to any hardware circuitry required to complete the communication link between the calculator and an external device. The terms "device," "external device," and "peripheral" refer to any piece of equipment which can be interfaced to the calculator.

The circuit diagrams provided are only examples. Circuit design will vary in each application. The discussions dealing with alphanumeric interchange and Direct Memory Access (DMA) apply only to the TEKTRONIX 31 Calculator.

Section 1, GENERAL, gives a description of overall interfacing requirements, explains the use of the REMOTE key on the calculator keyboard, provides signal definitions and pin locations on the INPUT/OUTPUT connector, and gives details on signal line termination, bus receivers, and bus drivers.

Section 2, SYSTEM DESCRIPTION, describes the TEKTRONIX 31 Calculator System Block Diagram and the TEKTRONIX 21 Calculator System Block Diagram. These diagrams are found in section 6 on pullout sheets. It is recommended that the reader keep the appropriate diagram extended for use as a reference while reading the manual.

Section 3, SYSTEM TIMING AND CONTROL, provides information on the timing required to synchronize an external device to the calculator's clocking system. System control signals and the remote address signal lines are discussed in detail.

Section 4, DATA BUS, covers the use of the calculator's primary path for data exchange. Timing diagrams and examples of typical interfacing circuits are provided.

Section 5, EXTERNAL BUS, gives information on how an external device can operate the calculator from a remote location, exchange ASCII character codes, and directly access the 31 Calculator's internal memory. Timing diagrams and examples of typical interfacing circuits are provided.

Section 6, DIAGRAMS, contains the System Block Diagrams discussed in section 2, the examples of Data Bus interface circuits discussed in section 4, and the examples of DMA interface circuits discussed in section 5.

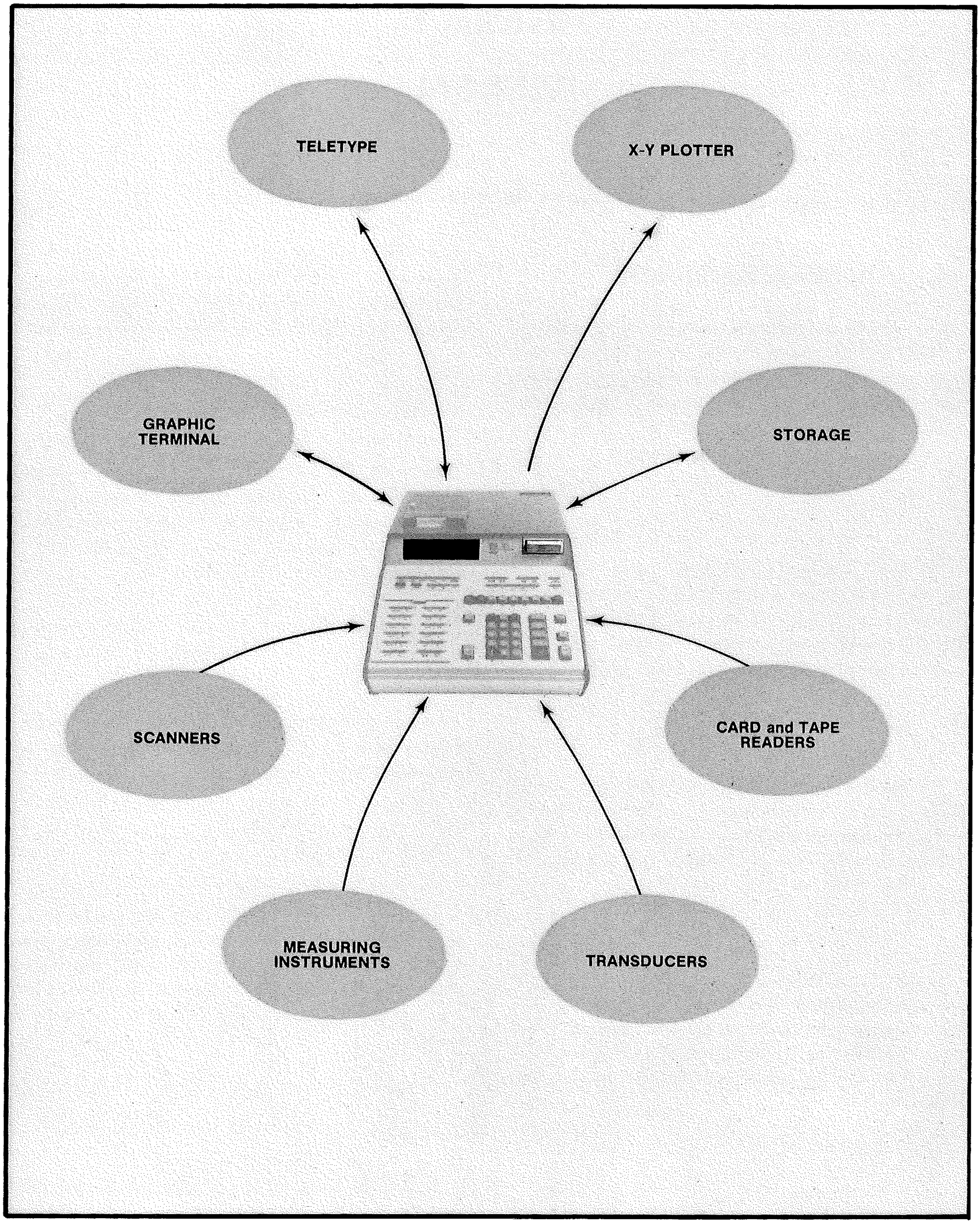


Fig. 1-1. A Calculator System.

INTRODUCTION

Tektronix calculators can be connected to any number of external devices to form a system capable of complete data acquisition, processing, and display. Refer to Fig. 1-1. One or more devices can be used to provide the calculator with raw digital data for processing and analysis. This includes scanners, measuring instruments, transducers, card readers, and tape readers. The calculator, acting as the central processor, processes the data and immediately sends the results to be displayed or stored for later use. The information can be displayed on graphic terminals, teletypes, and digital x-y plotters in the form of numbers, written words, or graphs.

The purpose of interface circuitry is to match the data-handling-capability of the external device to the calculator. The interface circuits insure that the information is properly formatted and travels at the proper rate.

The calculator initiates all transfers with an external device through remote control. The REMOTE Key on the keyboard gives the operator complete freedom to address and carry on a data exchange with any device in the system. Remote commands can also be issued under program control. An external device may operate asynchronous to the calculator, however, each transfer must be synchronized to the calculator timing system.

Refer to Fig. 1-2. Information exchange with the calculator is carried on over two groups of wires, called busses. The Data Bus consists of four wires; the External Bus consists of eight wires. These busses and associated Input/Output (I/O) signal lines are made available through a 50 pin Input/Output (I/O) connector located on the rear panel of the calculator.

The Data Bus is used to transfer data to and from the calculator's data registers. Data contained in these registers is displayed on the calculator display board. Data transferred on the Data Bus is formated into 16 time frames and transferred in BCD bit-parallel, digit-serial format. Details on the Data Bus are found in section 4, DATA BUS.

The External Bus is used primarily to transfer ASCII code which represents operating instructions or alphanumeric characters. Seven lines of the bus are used to transfer each code in parallel format. The activation of the eighth signal line on the bus tells the calculator that the code represents an alphanumeric character. The non-activation of the eighth signal line tells the calculator that the code represents an operating instruction.

The TEKTRONIX 31 Calculator offers a feature which allows R-register data and program steps to be transferred in byte-parallel format over the External Bus. This feature is called Direct Memory Access (DMA). Other devices in the system can exchange large quantities of stored information with the internal memory at a high speed. Program steps are transferred in one byte and R-register data in eight bytes. The rate of transfer on the External Bus is variable up to a maximum rate of 125k bytes/sec. Details on the External Bus and DMA are found in section 5, EXTERNAL BUS.

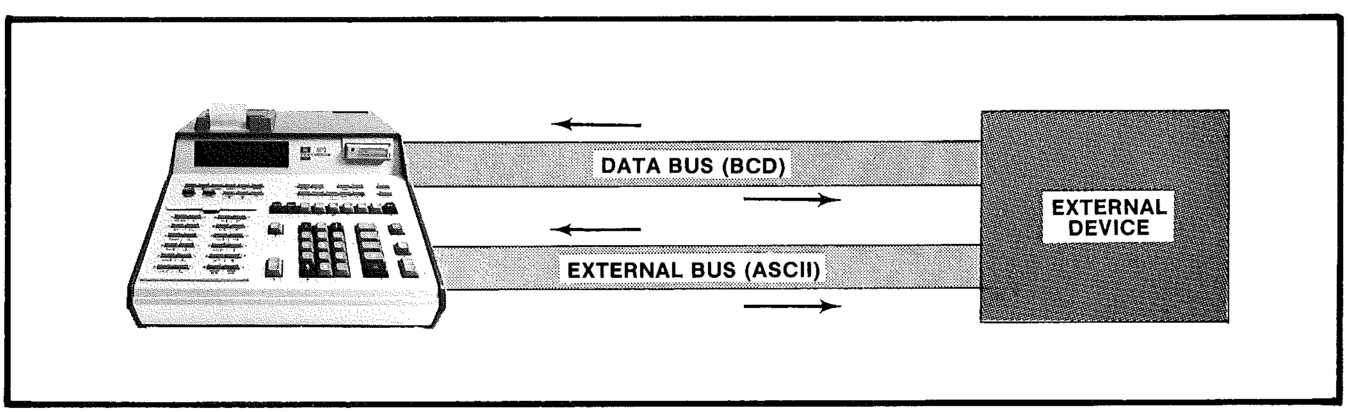


Fig. 1-2. Information exchange with an external device.

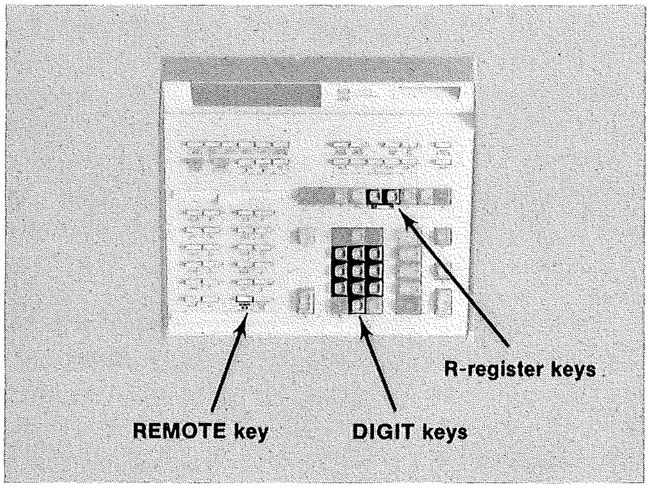


Fig. 1-3. The REMOTE Addressing Keys.

REMOTE ADDRESSING

General

The calculator addresses a peripheral device and communicates with the device through the remote address signal lines. The device responds by activating input signal lines on the I/O connector. Remote commands are used to enable and disable different functions within the device, set up data transfers, set and reset counters, and ask predefined questions during interrogation procedures.

Refer to Fig. 1-3. To send a command to an external device, the REMOTE key on the calculator keyboard is depressed, followed by two digits. (Or the entries can be made under program control.) The two digits represent a code unique to only one device in the system. This code identifies the device and the function it is to perform.

A remote address can also be indirectly called from an R-register in the TEKTRONIX 31 Calculator or a K-register is the TEKTRONIX 21 Calculator. Depressing the REMOTE key, the R▪▪▪key, or R▪▪key (K▪key on the 21 Calculator), and the appropriate digit(s) for the register location, puts the remote address on the lines. The last two digits of the mantissa stored in the register are considered the remote address.

If the entry following the REMOTE key is not a digit or an address for a register, the TEKTRONIX 31 Calculator responds with an E-3 error message which indicates a digit is required. The error message goes away when the required digits are entered.

The two digit address is sent in BCD format to be decoded by each device simultaneously. The designated peripheral responds when circuits on its interface are activated by the code.

Using a Remote Command to Control One Function.

It is common for a designer to let the TENS digit in the address designate the peripheral device and the UNITS digit designate the particular function it is to perform. For example:

> REMOTE 42 – "External device number 4, reset your counter to 0000".

In this example, the designer has used the TENS digit (4) to designate the device. The UNITS digit (2) designates the function "reset your counter to 0000". The combination of both numbers triggers the necessary circuits on the interface to reset the counter to 0000.

A remote address can also be used by the calculator for interrogation purposes. For example:

> REMOTE 30 – "External device number 3, is your memory full? If so, assert the $\overline{\text{ERST}}$ signal line to let me know."

In this case, the TENS digit (3) designates the device and the UNITS digit (0) triggers circuitry on the interface which responds to the predefined question. If the condition exists, the external device signals the calculator by activating the $\overline{\text{ERST}}$ signal line which causes the calculator display to flash. The calculator, operating under program control, recognizes the flashing display and takes further action as defined in the program.

The assignment of each remote address is completely arbitrary and it is up to the interface designer to assign an address to each function performed by his peripheral. (Addresses 00-09 will not conflict with Tektronix standard peripherals.)

Using a Remote Command to Control Many Functions

It might appear from the above examples that if the TENS digit is used to identify the peripheral device, then the number of definable functions for each device is limited to 10 (UNITS digit 0 through 9). However, the number of functions controlled by one remote address is, for all practical purposes, unlimited.

Assume that an interface designer wants to control a device which has 200 different functions. One remote address will do. This is accomplished by using the address to set up a data exchange over the Data Bus. The data value in the calculator display is used to represent the function in the device.

Consider this keystroke sequence:

CLEAR 123 = REMOTE 48

Depressing the CLEAR key clears the display (if it is not already clear). Depressing the 1 key, 2 key, and 3 key puts the number 123 in the display. Depressing the equal key places the decimal point in the proper location. The remote command 48 triggers circuits on the interface which interact with the calculator and complete the data transfer to the device. The external device then decodes the data value and interprets the number 123 as "Execute the 123rd function." If the number in the display were 122, then the 122nd function would be executed. It can be seen that by using this method, the number of functions controlled by one remote address is limited only by the size of the number that can be entered into the display.

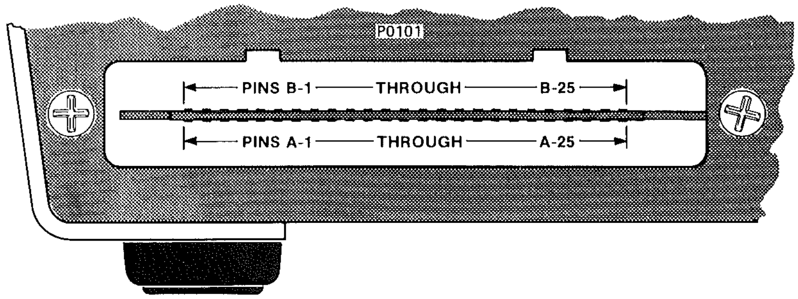
Detailed information on the remote address signal lines is found in section 3, SYSTEM TIMING AND CONTROL.

INPUT/OUTPUT (I/O) CONNECTOR

Signal Definitions

Refer to Fig. 1-4(A). All input/output signals are made available through a 50 pin connector located on the rear panel of the calculator. The following is a list of signal definitions and signal mnemonic names along with their pin assignments on the I/O connector. The complete signal name is given along with the mnemonic name and pin assignment in Fig. 1-4(B). All I/O signals with a bar over the mnemonic name ($\overline{\quad}$) are made TRUE (asserted) in the low state (near 0 VDC). All other signals, specifically UD1 – UD8 and TD1 – TD8, are made TRUE (asserted) in the high state (near +3.6 VDC).

GND (A1)	Chassis ground on the calculator.
$\overline{\text{STOP}}$ (A2)	Asserted by the calculator or an external device to stop a program in progress. Must be activated until the calculator releases the $\overline{\text{BUSY}}$ signal.
$\overline{\text{BUSY}}$ (A3)	Asserted by the calculator or an external device to inform the system that it is busy acquiring or processing data.
$\overline{\text{DISP}}$ (A4)	Used by an external device to send formatted messages directly to the calculator display.
$\overline{\text{ESYNC}}$ (A5)	A free running clock pulse issued every 50 μs. Defines a calculator word. (Some cycles may be less than 50 μs during a calculation.)
$\overline{\text{ECP2}}$ (A6)	A free running 500 kHz pulse used as a Master Clock. Defines a 2 μs time window for placing data on a bus.
$\overline{\text{ECP1}}$ (A7)	A free running 500 kHz pulse used as a Master Clock. Used to "strobe" data on and off a bus.
$\overline{\text{EXT32}}$ (A8)	The 6th line on the External Bus.
$\overline{\text{CALL}}$ (A9)	Used by an external device to start the execution of a subroutine located in the calculator's memory. Must be asserted along with an octal code from 101 to 130 which identifies the subroutine.
$\overline{\text{DCLOW}}$ (A10)	Used as a master reset to initialize logic circuits at power up. Asserted by the calculator when the power switch is turned on.
$\overline{\text{ERST}}$ (A11)	Asserted by an external device to set the calculator display into a FLASHING mode. Can be used as a signal to notify the calculator that a given condition exists or has occured within the external device.
$\overline{\text{EROT}}$ (A12)	Asserted by the calculator to let external devices know that the display is in the FLASHING mode.
+5 VOLTS (A13)	Voltage used to terminate all I/O signal lines at the far end. Cannot be used to supply power to interface circuits.
$\overline{\text{PRRT}}$ (A14)	A pulse issued by the calculator each time the CLEAR key is depressed. Can be issued under program control.
$\overline{\text{EXT128}}$ (A15)	The 8th line on the External Bus. Determines if the ASCII code on the other seven lines represents an operating instruction or an alphanumeric character.



(A) I/O Connector

A Lower Side: Function	Signal	Pin	Pin	Signal	B Upper Side: Function
CHASSIS GROUND	GND	1	1	GND	CHASSIS GROUND
STOP	STOP	2	2	DIO4	DATA INPUT/OUTPUT BIT 4
BUSY	BUSY	3	3	DIO8	DATA INPUT/OUTPUT BIT 8
DISPLAY	DISP	4	4	DIO2	DATA INPUT/OUTPUT BIT 2
EXTERNAL SYNC	ESYNC	5	5	DIO1	DATA INPUT/OUTPUT BIT 1
EXTERNAL CLOCK PHASE 2	ECP2	6	6	EXT 16	EXTERNAL BUS BIT 16
EXTERNAL CLOCK PHASE 1	ECP1	7	7	INHOUT	INHIBIT OUTPUT
EXTERNAL BUS BIT 32	EXT32	8	8	STBE	STROBE
CALL	CALL	9	9	EXT8	EXTERNAL BUS BIT 8
DC LOW	DCLOW	10	10	EXT 4	EXTERNAL BUS BIT 4
ERROR SET	ERST	11	11	EXT 64	EXTERNAL BUS BIT 64
ERROR FLASH	EROT	12	12	CRCV	CALCULATOR RECEIVE
TERMINATING VOLTAGE	+5 VOLTS	13	13	+5 VOLTS	TERMINATION VOLTAGE
PROGRAMMABLE RESET	PRRT	14	14	EXT 1	EXTERNAL BUS BIT 1
EXTERNAL BUS BIT 128	EXT128	15	15	EXT 2	EXTERNAL BUS BIT 2
ADDRESS VALID STROBE	AVS	16	16	EXTPTR	EXTERNAL PRINTER PRESENT
DIRECT MEMORY ACCESS WRITE	DMAWR	17	17	FLTP	FLOATING POINT DATA
DIRECT MEMORY ACCESS READ	DMARD	18	18	AC	ADDRESS COMPARE
FIXED POINT DATA	FXTP	19	19	IHIN	INHIBIT INPUT
UNITS DIGIT BIT 1	UD1	20	20	DMARY	DIRECT MEMORY ACCESS READY
TENS DIGIT BIT 1	TD1	21	21	UD2	UNITS DIGIT BIT 2
DATA WORD	DW	22	22	TD2	TENS DIGIT BIT 2
TENS DIGIT BIT 4	TD4	23	23	UD8	UNITS DIGIT BIT 8
TENS DIGIT BIT 8	TD8	24	24	UD4	UNITS DIGIT BIT 4
CHASSIS GROUND	GND	25	25	GND	CHASSIS GROUND

(B) I/O pin assignments

Fig. 1-4. I/O Connector with pin assignments.

$\overline{\text{AVS}}$ (A16) — Indicates that the address on the remote addressing lines is valid.

$\overline{\text{DMAWR}}$ (A17) — Asserted by an external device to indicate that it will be sending data to the calculator's internal memory during a Direct Memory Access transfer.

$\overline{\text{DMARD}}$ (A18) — Asserted by an external device to indicate that it will be receiving data from the calculator's internal memory during a Direct Memory Access transfer.

$\overline{\text{FXTP}}$ (A19) — Asserted by an external device to indicate to the calculator that a numeric value in scientific notation is to be sent over the Data Bus.

UD1 (A20) — One of eight remote address lines. Carries the least significant BCD bit of the Units digit in the remote address. Asserted in the high state.

TD1 (A21) — One of eight remote address signal lines. Carries the least significant BCD bit of the Tens digit in the remote address. Asserted in the high state.

$\overline{\text{DW}}$ (A22) — A 32 μs pulse issued by the calculator during a Data Bus transfer. Lets the external device know when the data transfer is taking place.

TD4 (A23) — One of the eight remote address signal lines. Carries the second most significant BCD bit of the Tens digit in the remote address.

TD8 (A24) — One of eight remote address signal lines. Carries the most significant BCD bit of the Tens digit in the remote address.

GND (A25) — Chassis ground on the calculator.

GND (B1) — Chassis ground on the calculator.

$\overline{\text{DIO4}}$ (B2) — One of four data lines on the Data Bus. Carries the second most significant bit.

$\overline{\text{DIO8}}$ (B3) — One of four data lines on the Data Bus. Carries the most significant bit.

$\overline{\text{DIO2}}$ (B4) — One of four data lines on the Data Bus. Carries the second least significant bit.

$\overline{\text{DIO1}}$ (B5) — One four data lines on the Data Bus. Carries the least significant bit.

$\overline{\text{EXT16}}$ (B6) — The 5th line on the External Bus.

$\overline{\text{INHOUT}}$ (B7) — Asserted by the calculator to prevent external devices from placing information on the External Bus.

$\overline{\text{STBE}}$ (B8) — A strobe pulse issued with every byte of information placed on the External Bus. Indicates that the information is valid and can be "read" by the receiving device.

$\overline{\text{EXT8}}$ (B9) — The 4th line on the External Bus.

$\overline{\text{EXT4}}$ (B10) — The 3rd line on the External Bus.

$\overline{\mathrm{EXT64}}$ (B11)	The 7th line on the External Bus.
$\overline{\mathrm{CRCV}}$ (B12)	Asserted by an external device during a "handshake" to tell the calculator it wants to send data over the Data Bus.
+5 VOLTS (B13)	Voltage used to terminate all I/O signal lines at the far end. Cannot be used to supply power to interface circuits.
$\overline{\mathrm{EXT1}}$ (B14)	The 1st line on the External Bus.
$\overline{\mathrm{EXT2}}$ (B15)	The 2nd line on the External Bus.
$\overline{\mathrm{EXTPTR}}$ (B16)	Asserted by an external printer to inform the calculator of its presence.
$\overline{\mathrm{FLTP}}$ (B17)	Asserted by an external device to indicate to the calculator that it desires the numeric value as it appears in the display.
$\overline{\mathrm{AC}}$ (B18)	Asserted by an external device when its remote address is issued by the calculator. Indicates to the calculator that the device is present.
$\overline{\mathrm{IHIN}}$ (B19)	Asserted by the calculator to prevent external devices from receiving information placed on the External Bus.
$\overline{\mathrm{DMARY}}$ (B20)	Indicates to an external device that the calculator is ready to begin a Direct Memory Access transfer.
UD2 (B21)	One of eight remote address lines. Carries the second least significant BCD bit of the Units digit in the remote address. Asserted in the high state.
TD2 (B22)	One of eight remote address lines. Carries the second least significant BCD bit of the Tens digit in the remote address. Asserted in the high state.
UD8 (B23)	One of eight remote address lines. Carries the most significant BCD bit in the Units digit in the remote address. Asserted in the high state.
UD4 (B24)	One of eight remote address lines. Carries the second most significant BCD bit of the Units digit in the remote address. Asserted in the high state.
GND (B25)	Chassis ground on the calculator.

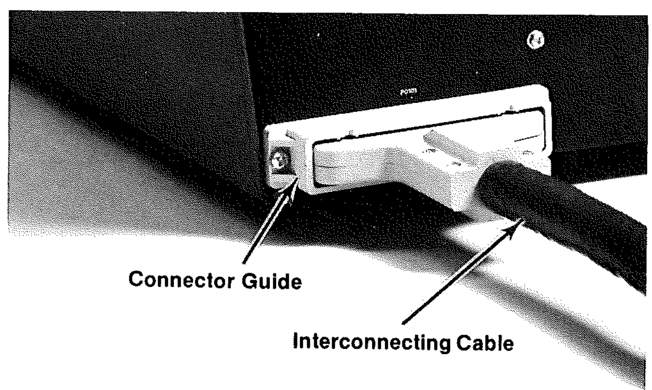


Fig. 1-5. Interconnecting Cable Installed.

Interconnecting Cable

Refer to Fig. 1-5. External devices are connected to the calculator via an Interconnecting Cable as shown in Fig. 1-5. A cable guide must be installed on the calculator to support the weight of the cable and to prevent the connector from being inserted upside down.

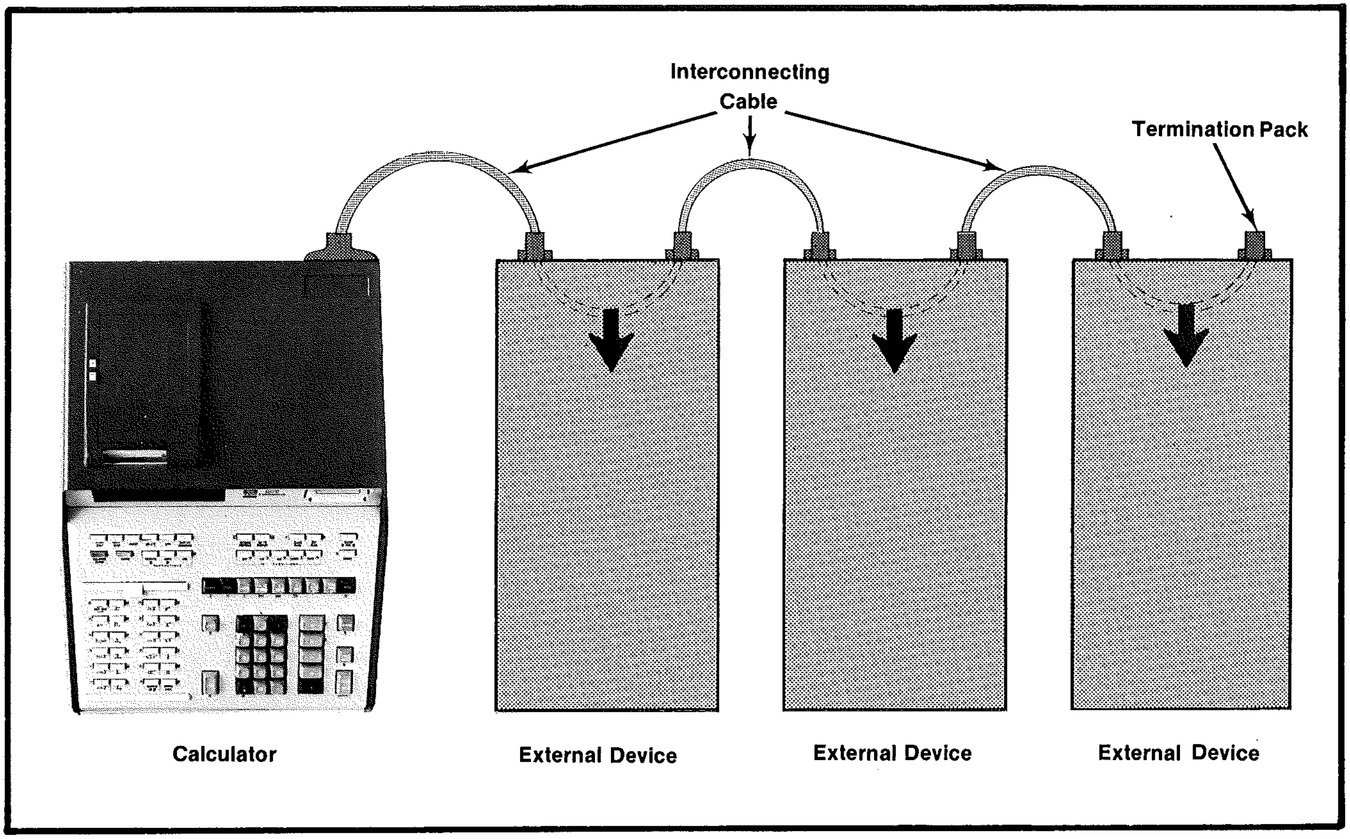


Fig. 1-6. External Devices connected in parallel.

Refer to Fig. 1-6. Any number of devices can be connected to the calculator as long as the total cable length does not exceed 100 feet. Connecting them as illustrated places the units electrically in parallel with each other; the last device provides signal line termination. A variety of cables are available from Tektronix and can be ordered through any Tektronix field office. These cables are listed in Fig. 1-7.

Signal Line Termination

All signal lines from the I/O connector are transmission lines driven by open-collector devices. They must be terminated at each end. One end of each line is already terminated within the calculator. The farthest end must be terminated with a 120Ω-5% resistor connected in series with two diodes and the +5V termination voltage from pins A13 and B13. (The diodes drop the voltage down to +3.6 VDC). Fig. 1-8 shows a common design for a termination resistor "pack." All signal lines from the I/O connector must be terminated regardless of whether they are used or not.

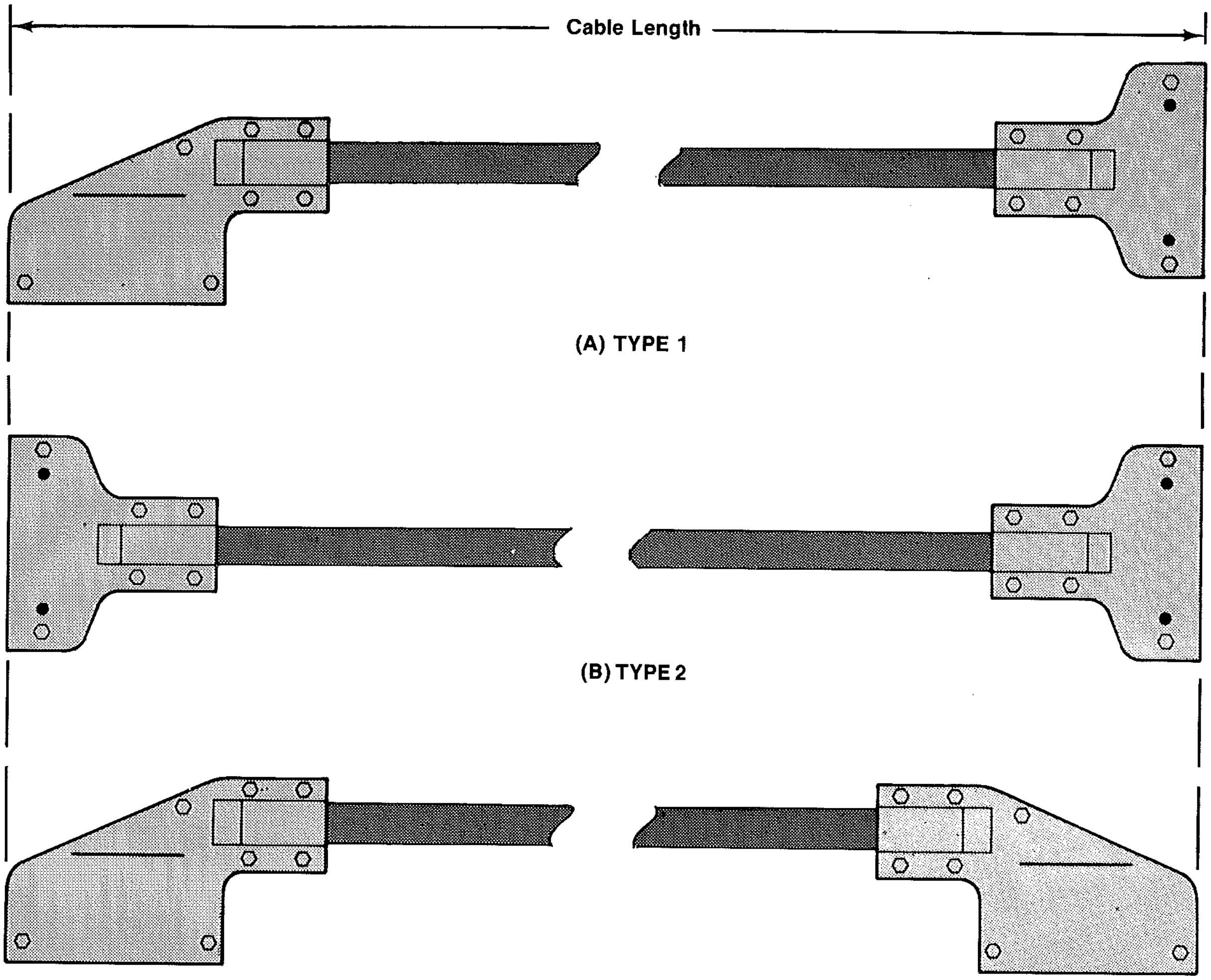
Interface circuits can be enclosed in a self-contained unit or nested within the main frame of the peripheral. Regardless of location, the interface must draw its main power from a source other than the calculator. Under no circumstances should the termination voltage (A13 and B13) be used to supply power to the interface.

Signal Line Receivers

Refer to Fig. 1-8. Each I/O signal line must be "buffered" with a National Semi-conductor Unified Bus Receiver or its equivalent. These devices are designed for use in bus-organized data transmission systems. They provide high noise immunity (2 Volts typical). In addition, these devices do not draw current from the bus if they have no power applied. There are currently two such receivers in common usage:

DM 8836 Quad Inverting Receiver

DM 8837 Hex Inverting Receiver



Cable Length	TYPE 1	TYPE 2	TYPE 3
30" ±10%	012-0498-00	012-0499-00	012-0500-00
7 ft. ±10%	012-0498-01	012-0499-01	012-0500-01

Fig. 1-7. Cables available from Tektronix.

Their electrical specifications include:

Input current (input high)	⩽ 50 μA
Turn on Delay	⩽ 30 ns
Turn off Delay	⩽ 30 ns

Signal Line Drivers

Refer to Fig. 1-8. Signal line drivers must be TTL open-collector SN7438N devices or their equivalent and have the following characteristics:

High level output leakage current	⩽ 250 μA
Low level output voltage	⩽ 0.8 Volts, 60 mA sink current
Turn on Delay	⩽ 18 ns
Turn off Delay	⩽ 22 ns

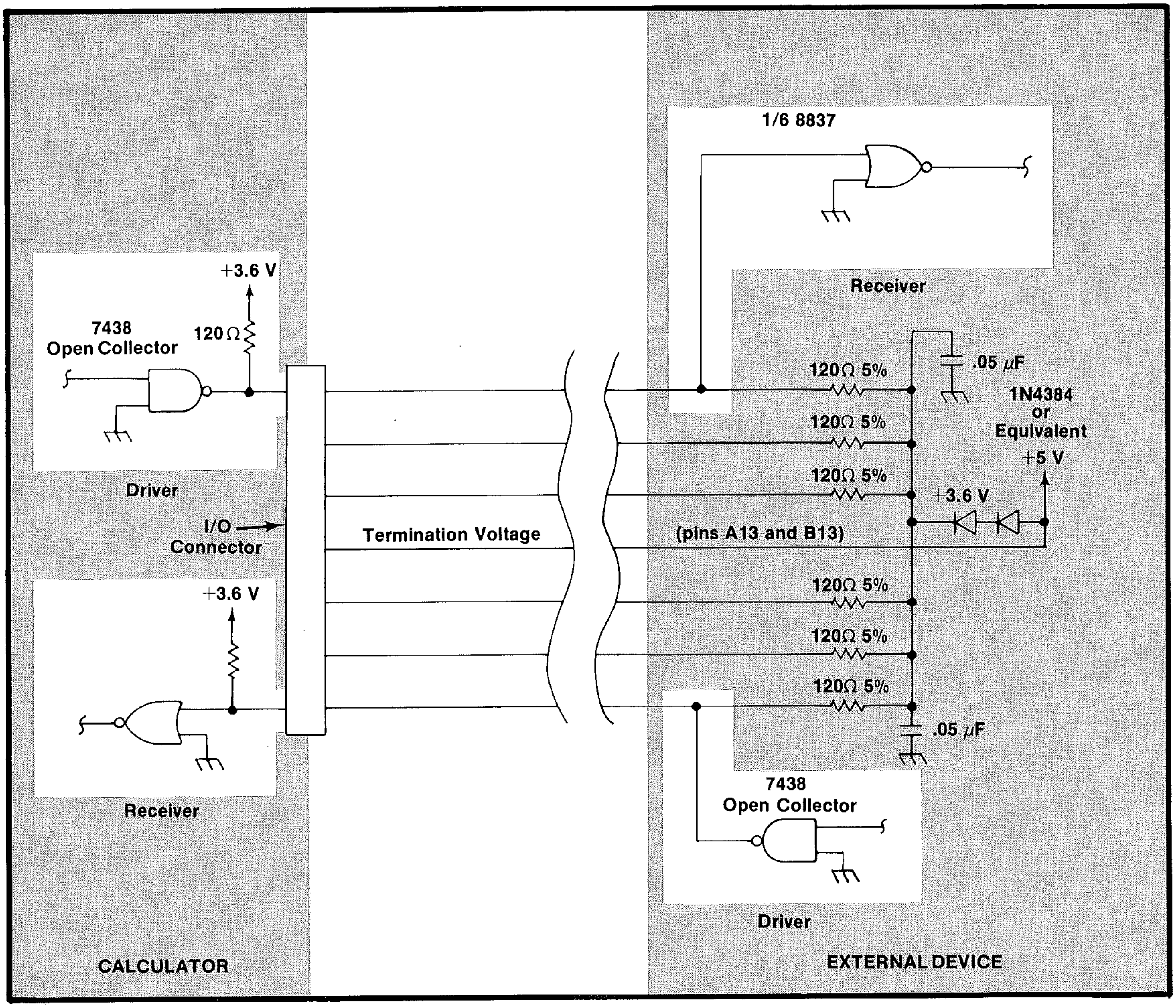


Fig. 1-8. Terminating I/O signal lines at the far end.

Section 2

SYSTEM DESCRIPTION

INTRODUCTION

This section describes the TEKTRONIX 31 and 21 System Block Diagrams found on pullout sheets in section 6 DIAGRAMS, (Fig. 6-1 and Fig. 6-2). It is recommended that the reader keep the appropriate diagram extended for use as a reference while reading this section and the rest of the manual. Following the TEKTRONIX 21 System Block Diagram description is an illustration and description of the I/O signal organization.

TEKTRONIX 31 CALCULATOR SYSTEM DESCRIPTION

General

The diagram on pullout sheet Fig. 6-1 represents the TEKTRONIX 31 Calculator connected to two external devices. The eight blocks on the left represent the calculator; the two on the right represent two external devices. It can be seen that the Interconnecting Cable is just an extension of the internal data lines within the calculator. Devices connected to the I/O signal lines are connected in parallel with each other.

Bus Lines

There are two major busses within the calculator, the External Bus and the Data Bus. The External Bus is used to transfer operational keycodes and alphanumeric codes in bit parallel format. The Data Bus is used to transfer numeric values in BCD bit-parallel, digit-serial format.

Keyboard

The keyboard makes it possible for an operator to manually enter data and give instructions to the calculator. Each time a key is depressed, a seven bit ASCII code is placed on the External Bus. This ASCII code can represent an operating instruction, an alphanumeric character, or a digit for data entry. The ASCII code represents an alphanumeric character when the operator depresses the HOLD FOR ALPHA key. A listing of TEKTRONIX 31 Calculator keycodes is found in Table 5-1, section 5.

Calculator Board

The calculator board contains logic circuits which perform mathematical operations. The board receives its instructions from the External Bus and uses the Data Bus to manipulate data during a calculation. Data entering the calculator is placed in data registers located on the calculator board. When the board executes an instruction, it works with the contents of these registers.

Display Board

The display is used to present information to the operator. This information can be entry data, computed results, error messages, storage register contents, or program steps. Information in the display is a duplication of data held elsewhere in the calculator and is erased each time the calculator updates the display on the trailing edge of the $\overline{\text{BUSY}}$ signal.

Programmer Board

The programmer board contains circuitry required to carry out the programming functions of the calculator. These circuits handle non-mathematical instructions such as LIST, LABEL, and PRINT DSPLY. The programmer board also coordinates data flow within the calculator. Each keycode placed on the External Bus is examined by the programmer board, which sends it to the proper location for processing.

Memory Board

The memory board contains registers used to store data and program steps. Part of the storage capacity is set aside for program steps and the other part for R-register data. The number of program steps and R-registers for a given machine depends on the memory option selected when the machine was purchased.

Magnetic Tape

The magnetic tape provides a way of making a permanent record of R-register data and program steps stored on the memory board. Large amounts of data and/or program steps can be transferred to and from the magnetic tape at a fast rate of speed.

Printer

The printer is used to make a readable record of data and program steps. Information sent to the printer is first kept in temporary storage on the programmer board. The characters are printed when the total number reaches 16, or when the HOLD FOR ALPHA key is released and an operational keycode is executed.

System Timing and Control

The system timing and control block contains circuits which generate clock pulses and make control signals available to external devices. This block includes the remote address signal lines.

External Device Number 1 and External Device Number 2

External Devices number 1 and 2 are examples of peripherals connected to the 31 Calculator. Each device is connected in parallel, with the last device providing signal line termination. Information placed on the External Bus or the Data Bus is available to each device simultaneously; however, a calculator command through remote addressing is required before a device can take information from the lines or place information on the lines.

System Operation

The calculator receives its operating instructions via the External Bus. If the calculator is not busy, it responds to a seven bit ASCII code when it is placed on the External Bus and execution command $\overline{\text{STBE}}$ (STROBE) is given. $\overline{\text{STBE}}$ is issued by the device placing the keycode on the bus. The ASCII code can come from the calculator keyboard or an external keyboard.

The programmer board continually monitors the External Bus, examines each code coming into the calculator, and sends it to the proper location for processing. If the code represents a mathematical function, the programmer board re-issues the code to the calculator board via the External Bus. The calculator board reads the code, enters the BUSY mode, and executes the instruction.

If an alphanumeric code is placed on the External Bus, the programmer board captures the code and temporarily stores it. Stored codes are printed as soon as the total number reaches 16, or when the HOLD FOR ALPHA key is released and an operational instruction is executed.

Data entries from the keyboard are also placed on the External Bus. These digits are passed from the External Bus to the Data Bus via the calculator board. They are put in the calculator board data registers and appear in the display.

The calculator board uses the Data Bus for data processing during a calculation. Data exchange with an external device is also carried over the Data Bus. The calculator exercises strict control over the use of the Data Bus and transfers data to an external device only during selected time periods.

If the calculator is in the LEARN mode, the programmer board asserts $\overline{\text{IHIN}}$ to inhibit the input to every device on the External Bus. This is to prevent the keycodes from being executed. The programmer board has an automatic sequencer which takes each code from the External Bus and stores it on the memory board in the proper location. If the START keycode is placed on the External Bus, the calculator enters the program RUN mode. START can be issued from the calculator keyboard or from an external device acting as a keyboard. When the START keycode is recognized by the programmer board, it asserts $\overline{\text{INHOUT}}$ to inhibit the output of every device on the External Bus. (This is to prevent outside interference.) The automatic sequencer then retrieves the desired program steps from the memory board and places them on the External Bus, one at a time, to be executed. A strobe is issued with each keycode. The program runs to completion unless it is interrupted with a $\overline{\text{STOP}}$ command. The $\overline{\text{STOP}}$ command is non-destructive; that is, the program continues where it left off as soon as the CONTINUE key is struck.

External devices can gain direct access to the memory board via the External Bus. This is advantageous if large amounts of data and/or program steps must be transferred. The information is transferred over the External Bus in byte-parallel format. For details on how this is accomplished, refer to the text titled DIRECT MEMORY ACCESS in section 5, EXTERNAL BUS.

TEKTRONIX 21 CALCULATOR SYSTEM DESCRIPTION

General

The diagram on pullout sheet Fig. 6-2 represents the TEKTRONIX 21 Calculator connected to two external devices. The seven blocks on the left represent the calculator; the two on the right represent two external devices. It can be seen that the Interconnecting Cable is just an extension of the internal data lines within the calculator. Devices connected to the I/O signal lines are connected in parallel with each other.

Bus Lines

There are two major busses in the calculator, the External Bus and the Data Bus. The External Bus is used to transfer operational keycodes in bit-parallel format. The Data Bus is used to transfer numeric values in BCD bit-parallel, digit-serial format.

Keyboard

The keyboard makes it possible for an operator to manually give instructions to the calculator. Each time a key is depressed, a keycode is placed on the External Bus. This keycode is a seven bit ASCII code which represents an operating instruction or a digit for data entry. A listing of TEKTRONIX 21 Calculator keycodes is found in Table 5-2, section 5.

Calculator Board

The calculator board contains logic circuits which perform mathematical operations. The board receives its instructions from the External Bus and uses the Data Bus to manipulate data during a calculation. Data entering the calculator is placed in data registers located on the calculator board. When the board executes an instruction, it works with the contents of these registers.

Display Board

The display board is used to present information to the operator. This information can be entry data, computed results, storage register contents, or an error message. Information in the display is a duplication of information held elsewhere in the calculator and is erased each time the calculator updates the board on the trailing edge of $\overline{\text{BUSY}}$.

f(X) Board

The f(X) board contains circuitry required to carry out the programming functions of the calculator. These circuits handle the execution of non-mathematical instructions such as LEARN and STEP.

Magnetic Card

The magnetic card provides a way to make a permanent record of program steps stored on the f(X) board. Up to 256 program steps can be recorded on a magnetic card, which is a strip of plastic coated with a magnetic material.

Printer

The printer is used to make a readable record of data and program steps stored within the calculator. When the PRINT DSPLY key is depressed, the numeric value in the display is sent to the printer via the Data Bus. When the LIST key is depressed, the program steps stored on the f(X) board are sent to the printer via the External Bus.

System Timing and Control

The system timing and control block contains circuits which generate clock pulses and make control signals available to external devices. This block includes the remote address signal lines.

External Device Number 1 and External Device Number 2

External devices number 1 and 2 are examples of peripherals connected to the 21 Calculator. Each device is connected in parallel, with the last device providing signal line termination. Information placed on a Bus is available to each device simultaneously; however, a calculator command through remote addressing is required before a device can take information from the lines or place information on the lines.

System Operation

The calculator receives its operating instructions via the External Bus. If the calculator is not busy, it responds when a seven bit ASCII code is placed on the External Bus and execution command $\overline{\text{STBE}}$ (STROBE) is given. $\overline{\text{STBE}}$ is issued by the device placing the keycode on the bus. The ASCII code can come from the calculator keyboard or an external keyboard.

Data entries from the keyboard are placed on the External Bus. These digits are passed from the External Bus to the Data Bus via the calculator board. They are put in the calculator board data registers and appear in the display.

If the calculator enters the LEARN mode, the f(X) board asserts $\overline{\text{IHIN}}$ to inhibit the input to every device on the External Bus. This is to prevent the keycodes from being executed. The f(X) board has an automatic sequencer which takes each code from the External Bus and stores it on the f(X) board in the proper location.

The calculator enters the program RUN mode when f(X) keycode is placed on the External Bus (X=desired number). This can be issued from the keyboard or from an external device acting as a keyboard. When f(X) is recognized by the f(X) board, the f(X) board asserts $\overline{\text{INHOUT}}$ to inhibit the output of every device on the External Bus. (This is to prevent outside interference.) The automatic sequencer then places the program steps on the External Bus one at a time to be executed. A strobe ($\overline{\text{STBE}}$) is issued with each keycode. The program is executed to completion unless the STOP key is depressed. The STOP key is non-destructive; that is, the program continues where it left off as soon as the CONT (continue) key is struck.

If the PRINT DSPLY key is depressed, the contents of the display are sent to the printer as a 16 character string of digits over the Data Bus. All leading zeros are suppressed. If the LIST key is depressed, the automatic sequencer sends the program steps stored on the f(X) board to the printer. They are sent one at a time via the External Bus.

I/O SIGNAL ORGANIZATION

Introduction

This text describes Fig. 2-1. The I/O signals are organized into groups according to the function they perform. Although the I/O signal lines are not physically arranged within the cable as shown in the figure, this illustration is provided to help the reader develop a mental picture of the I/O signal organization.

General

Fig. 2-1 is a cut-away view of an Interconnecting Cable as it would appear if the I/O signals were grouped according to their function. The +5 V termination voltage (pins A13 and B13) and chassis ground (pins A1 and A25; B1 and B25) are not included. There are three major groups of signals; the Data Bus and Associated Signals, System Timing and Control, and the External Bus and Associated Signals.

Compare Fig. 2-1 to the three major signal paths on the 21 and 31 Calculator System Block Diagrams. External devices connected to the I/O signal lines are connected in parallel with each other. Any device in the system can activate a signal line "by pulling it low", however, a device must receive a remote command from the calculator before it is allowed to do so.

Data Bus and Associated Signals

The Data Bus is the calculator's primary path for data exchange. Four bit BCD codes representing digits in a data value are placed on lines $\overline{\text{DIO1}}$-$\overline{\text{DIO8}}$ one digit at a time. The lines are labeled according to the significance of the bit they carry. Before the transfer, a data exchange agreement is made with the external device via handshake signals $\overline{\text{AC}}$ (ADDRESS COMPARE), $\overline{\text{CRCV}}$ (CALCULATOR RECEIVE), $\overline{\text{FXTP}}$ (FIXED POINT), and $\overline{\text{FLTP}}$ (FLOATING POINT). A special function $\overline{\text{DISP}}$ (DISPLAY) allows an external device to send BCD digits directly to the display board. These digits can be formatted into a numeric code which informs the operator that a given condition exists or has occurred within the device. Details on the Data Bus and Associated Signals are found in section 4, DATA BUS.

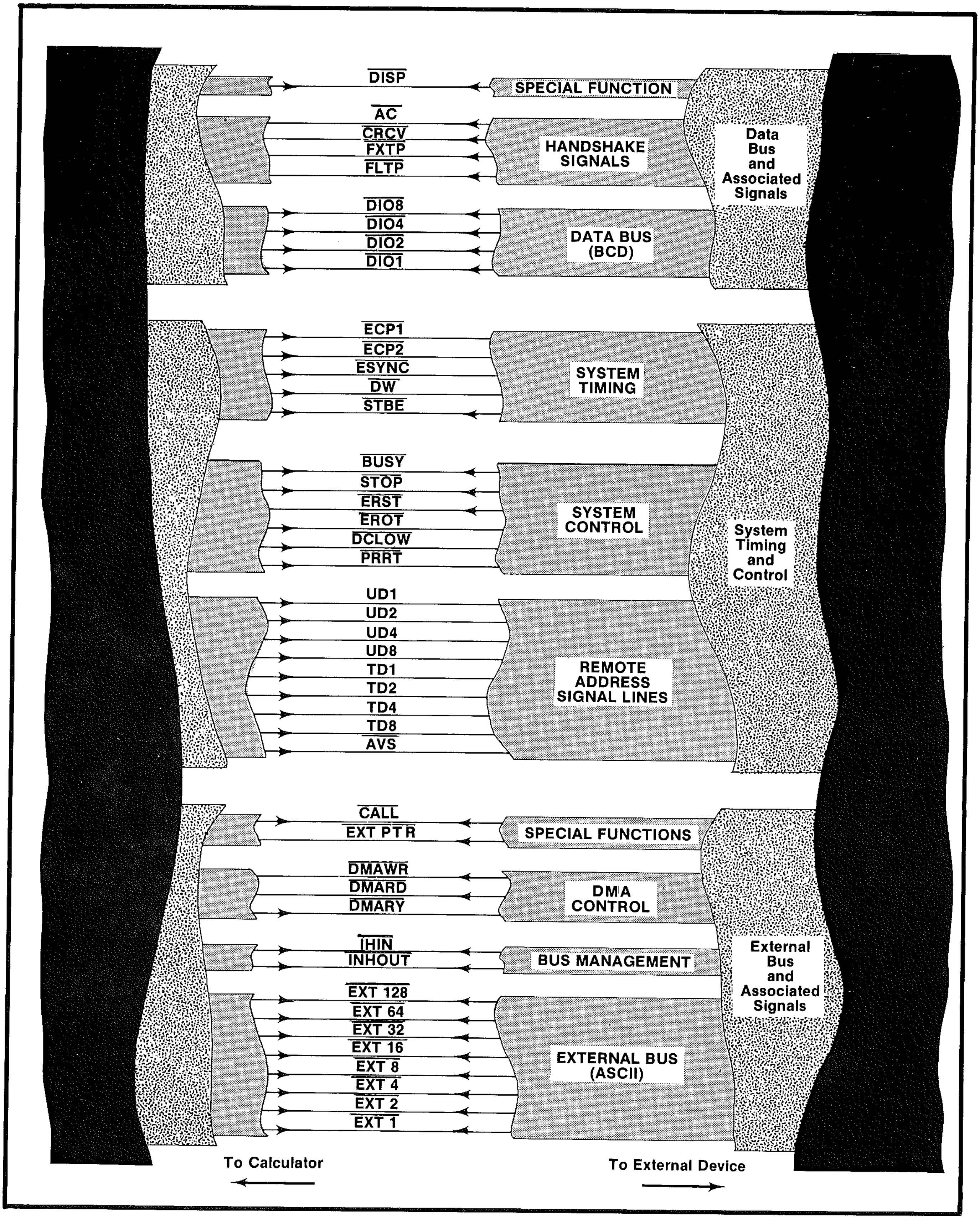


Fig. 2-1. I/O Signal Organization.

System Timing and Control

The center group of signals, System Timing and Control, is divided into three sub-groups. System Timing includes $\overline{\text{ECP1}}$ (EXTERNAL CLOCK PHASE ONE) and $\overline{\text{ECP2}}$ (EXTERNAL CLOCK PHASE TWO), which are master clocks for the system. $\overline{\text{ESYNC}}$ (EXTERNAL SYNC) marks the beginning of a calculator word. $\overline{\text{DW}}$ (DATA WORD) provides a "time window" for each Data Bus transfer. $\overline{\text{STBE}}$ (STROBE) transfers a byte of information on the External Bus.

System Control signals include the following: $\overline{\text{BUSY}}$, which informs the system that the calculator and/or peripheral is busy; $\overline{\text{STOP}}$, which stops a program being executed by the calculator; $\overline{\text{ERST}}$ (ERROR SET), which causes the display to flash; $\overline{\text{EROT}}$ (ERROR FLASH), which informs the system that the calculator display is flashing; $\overline{\text{DCLOW}}$ (DC LOW), which is the master reset for the system on power up; and $\overline{\text{PRRT}}$ (PROGRAMMABLE RESET), which is activated by the CLEAR key on the calculator keyboard.

Remote Address Signal Lines TD1-TD8, UD1-UD8, and $\overline{\text{AVS}}$ (ADDRESS VALID STROBE) deliver remote commands to an external device. Details on the System Timing and Control signals are found in section 3, SYSTEM TIMING AND CONTROL.

Details on System Timing and Control are found in section 3, SYSTEM TIMING AND CONTROL.

External Bus and Associated Signals

The External Bus ($\overline{\text{EXT1-EXT128}}$) is used to transfer ASCII code, R-register data, and program steps to and from an external device. $\overline{\text{IHIN}}$ (INHIBIT INPUT) and $\overline{\text{INHOUT}}$ (INHIBIT OUTPUT) are non-selective bus management signals which block the inputs and outputs of all devices connected to the External Bus.

DMA Control signals $\overline{\text{DMARD}}$ (DMA READ), $\overline{\text{DMAWR}}$ (DMA WRITE), and $\overline{\text{DMARY}}$ (DMA READY) set up a direct transfer between the external device and the calculator's internal memory.

The special function $\overline{\text{EXTPTR}}$ (EXTERNAL PRINTER) informs the calculator that an external printer is present. The special function $\overline{\text{CALL}}$ allows an external device to call a subroutine from the calculator's memory for immediate execution.

Details on the External Bus and Associated Signals are found in section 5, EXTERNAL BUS.

Section 3

SYSTEM TIMING AND CONTROL

INTRODUCTION

This section describes the I/O signals used for System Timing and Control. Refer to Fig. 3-1. The following signals are included: System Timing $\overline{\text{ECP1}}$, $\overline{\text{ECP2}}$, $\overline{\text{ESYNC}}$, $\overline{\text{DW}}$, and $\overline{\text{STBE}}$. System Control $\overline{\text{BUSY}}$, $\overline{\text{STOP}}$, $\overline{\text{ERST}}$, $\overline{\text{EROT}}$, $\overline{\text{DCLOW}}$, and $\overline{\text{PRRT}}$. Remote Address lines TD1 – TD8, UD1 – UD8, and $\overline{\text{AVS}}$.

SYSTEM TIMING

General

Refer to Fig. 3-2. Calculator operations are synchronized from two master clocks. These two clocks, $\overline{\text{ECP1}}$ and $\overline{\text{ECP2}}$, originate from a 5 MHz master oscillator and are out of phase so that the negative going pulses do not overlap. An additional pulse, $\overline{\text{ESYNC}}$, is generated every 50μs. This defines a 25-bit calculator word period.

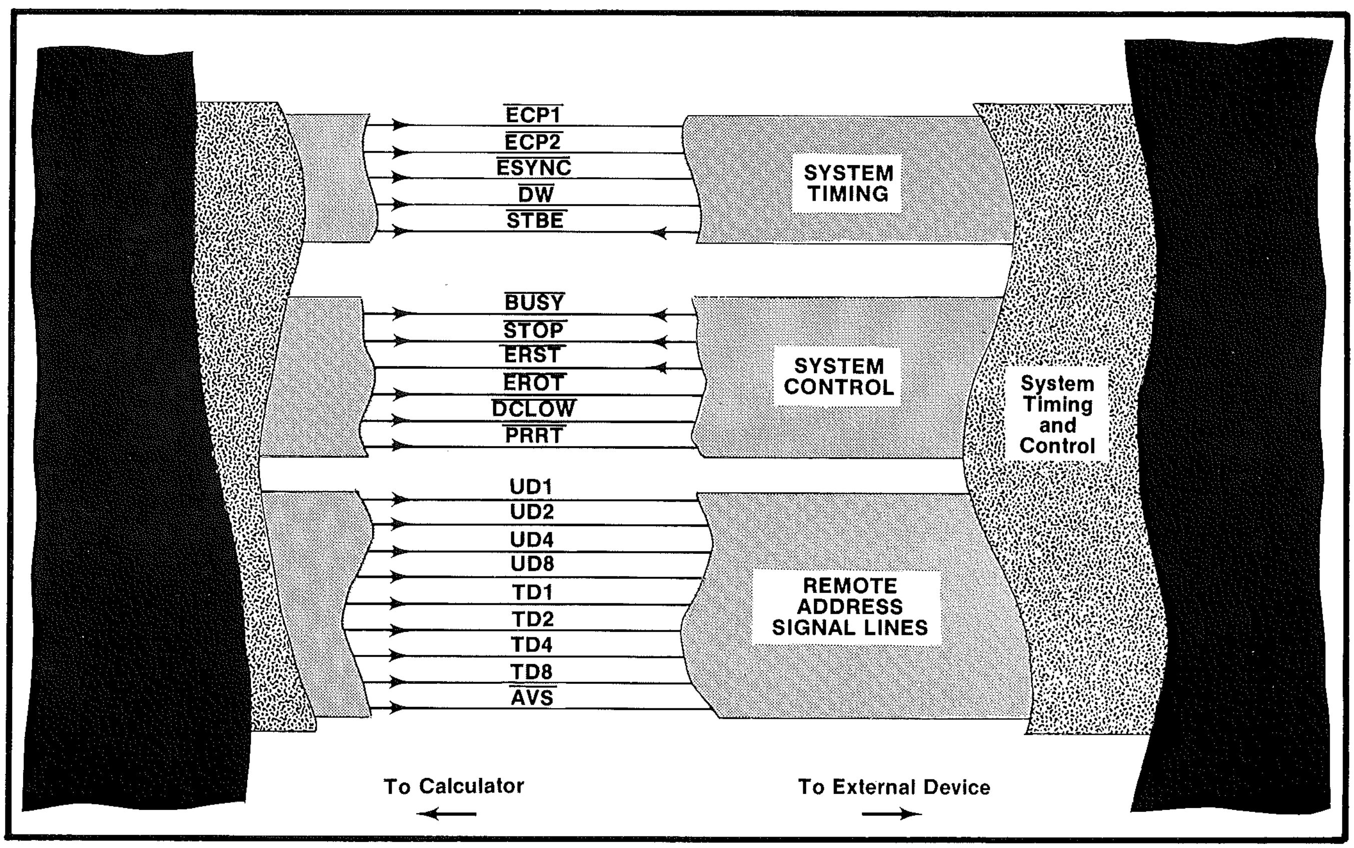


Fig. 3-1. System Timing and Control.

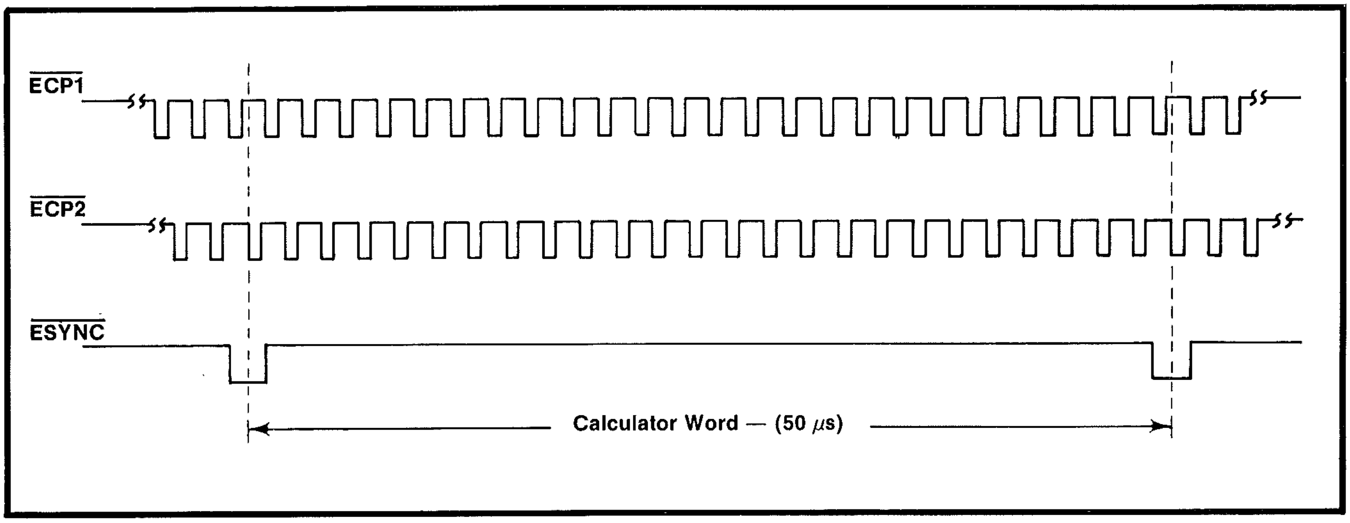


Fig. 3-2. Master Timing Signals

EXTERNAL CLOCK PHASE ONE ($\overline{ECP1}$)

Refer to Fig. 3-3. $\overline{ECP1}$ is a constant flow of negative going pulses which occur every 2 µs (leading edge to leading edge), and last for 800 nanoseconds. Within the calculator, $\overline{ECP1}$ is used as a time reference for data manipulation on the individual boards. It is also used to generate the $\overline{ESYNC}$ (EXTERNAL SYNC) pulse.

EXTERNAL CLOCK PHASE TWO ($\overline{ECP2}$)

Refer again to Fig. 3-3. $\overline{ECP2}$ is a constant flow of negative going pulses. These pulses also occur every 2 µs (leading edge to leading edge), but last for 600 nanoseconds. $\overline{ECP2}$ is used as the time reference for information transfer on the Data Bus and the External Bus.

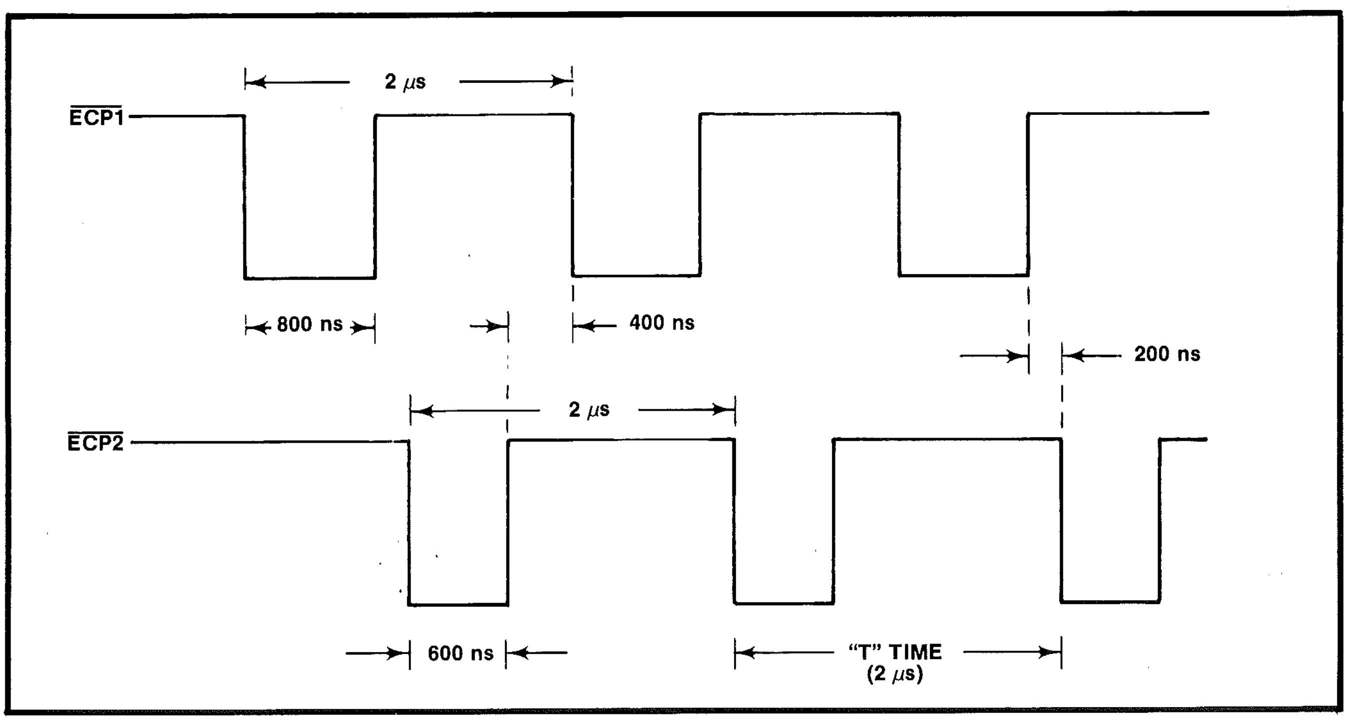


Fig. 3-3. Two Phase Master Clock.

Ⓐ

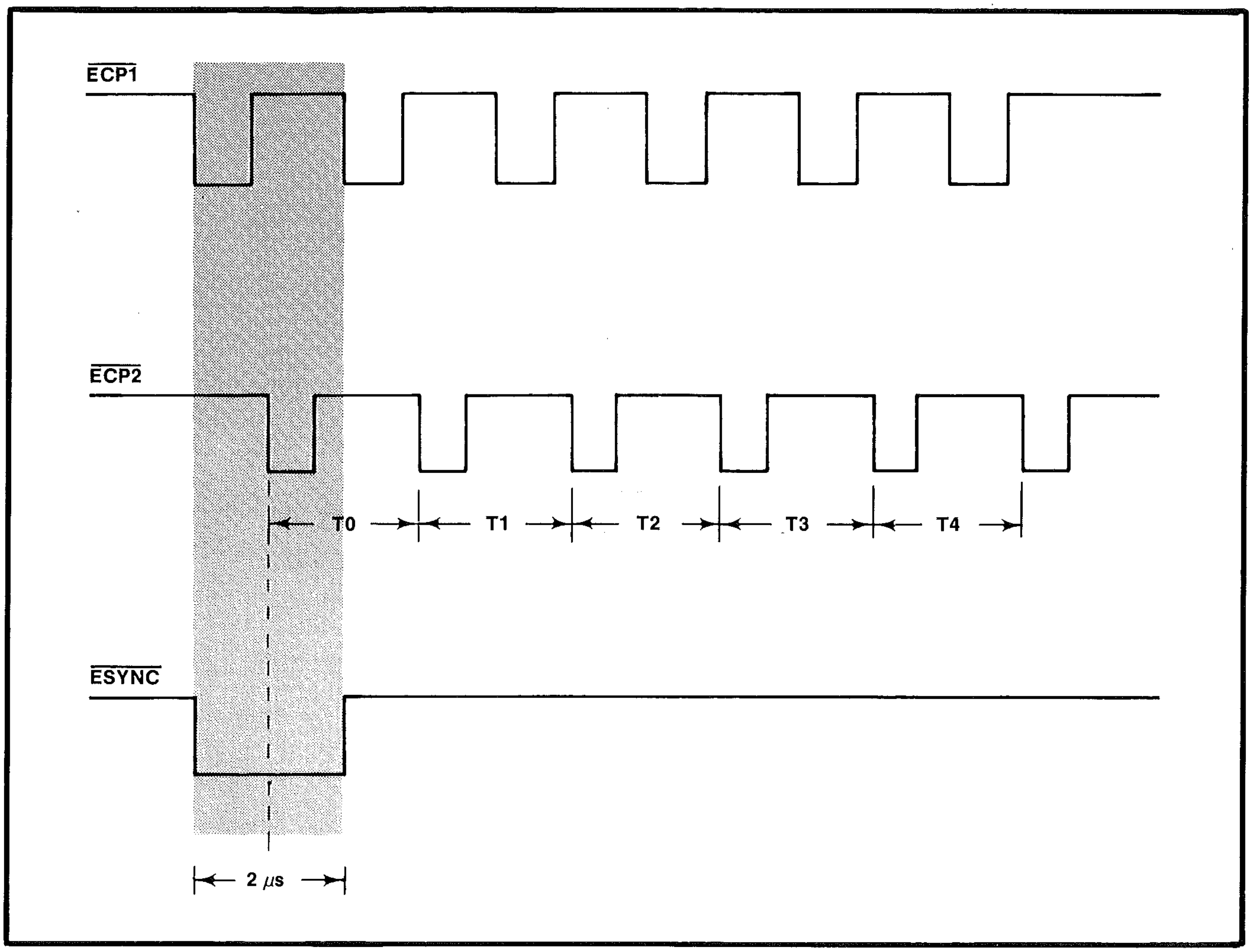


Fig. 3-4. $\overline{\text{ESYNC}}$ establishes T0 for a Data Bus transfer.

EXTERNAL SYNC ($\overline{\text{ESYNC}}$)

Refer to Fig. 3-4. The $\overline{\text{ESYNC}}$ pulse is generated every 50 μs. The middle of the 2 μs $\overline{\text{ESYNC}}$ pulse marks the beginning of a 25 bit calculator word.

T Times

Refer to Fig. 3-4. The cycle time of $\overline{\text{ECP2}}$ is used as a time frame for transferring a byte of information on the Data Bus and the External Bus. This time frame is called a T Time. A T Time starts at the leading edge of $\overline{\text{ECP2}}$ and lasts for the duration of the cycle, exactly 2μs.

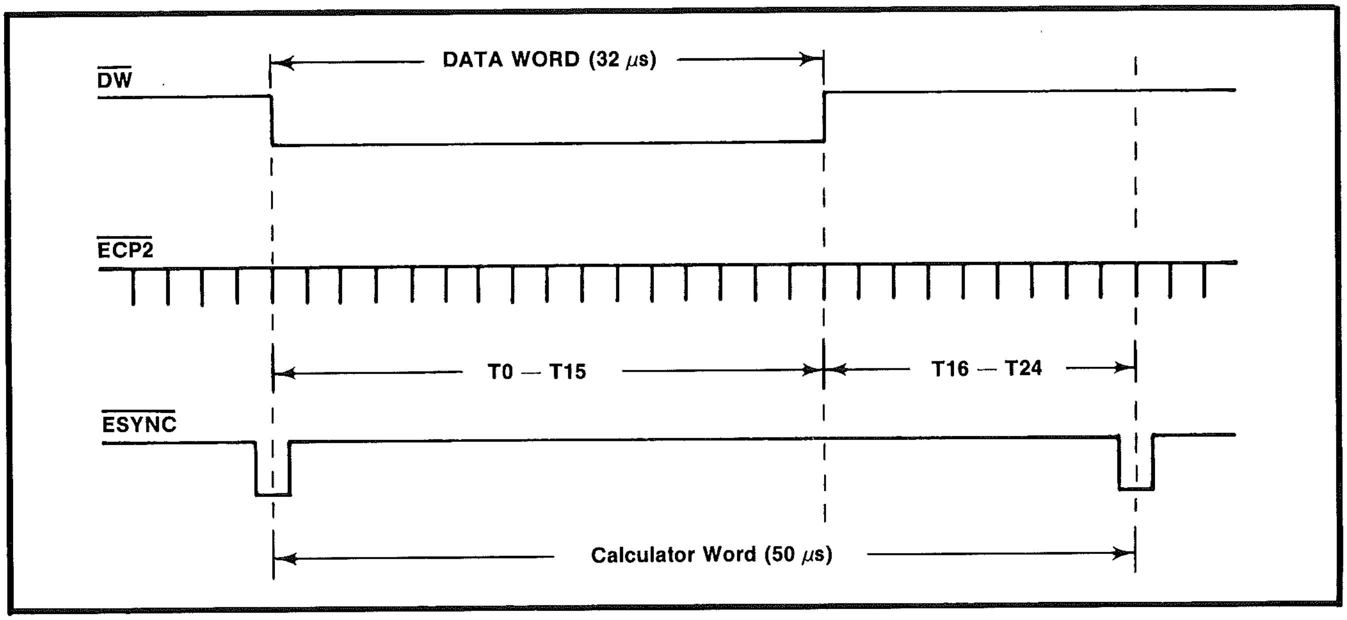


Fig. 3-5. DATA WORD issued from T0-T15.

Timing Requirements for the Data Bus

Refer to Fig. 3-5. It takes the first 16 T Times of a calculator word (T0 – T15) to transfer a complete numeric value over the Data Bus. (Period T16 – T24 is used by the calculator for internal data manipulation.) The data transmission takes place when the calculator issues a 32 μs "time window" called DATA WORD ($\overline{\text{DW}}$). If the calculator is sending data to an external device, it starts the transmission at the beginning of $\overline{\text{DW}}$. If an external device is sending data to the calculator, it places the data on the Data Bus when the calculator issues $\overline{\text{DW}}$. $\overline{\text{DW}}$ is the only time when a transmission can take place over the Data Bus.

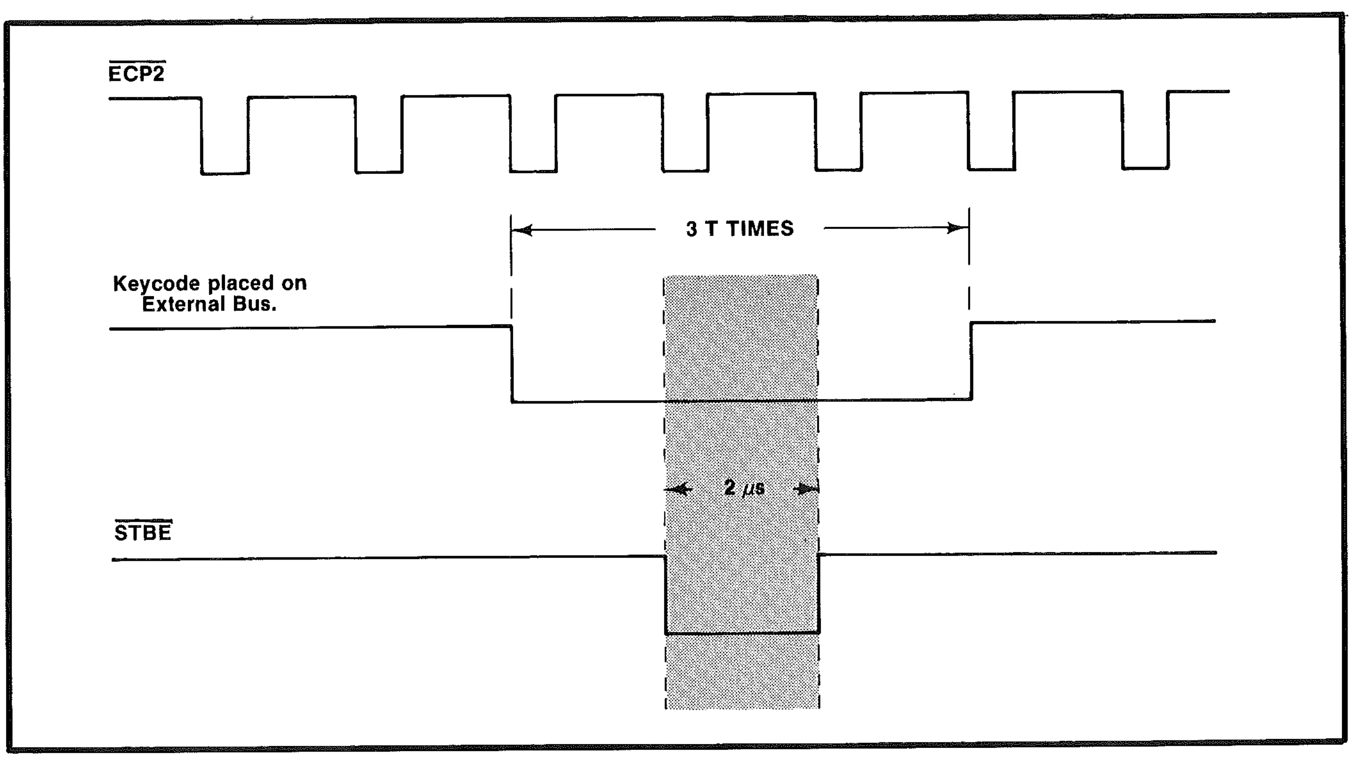


Fig. 3-6. Keycode transfer on the External Bus.

Timing Requirements for the External Bus

Refer to Fig. 3-6. It takes three T Times (6 μs) to transfer a byte of information over the External Bus. The selection of these three T Times is arbitrary; that is, any three $\overline{ECP2}$ cycles can be used. The information is placed on the bus during the first T Time, "strobed" during the second T Time, and must remain on the lines for a third T Time. The 2 μs $\overline{STBE}$ pulse lets the receiving device know that the information is valid and may be taken off the External lines. The strobing rate is determined by the time it takes the calculator to process the individual bytes of information. A new byte of information can be put on the lines and "strobed" after the calculator releases the $\overline{BUSY}$ signal (unless $\overline{INHOUT}$ is asserted). The maximum rate of transfer is 125k bytes/sec. This can occur during a Direct Memory Access transfer (see Fig. 3-7).

SYSTEM CONTROL

General

The following is a list of general purpose signals which contribute to the smooth operation of the system.

$\overline{BUSY}$

The $\overline{BUSY}$ signal indicates that some member of the system is busy. If the calculator is asserting $\overline{BUSY}$, an external device must wait until $\overline{BUSY}$ is released before additional interaction can take place. If an external device is asserting $\overline{BUSY}$, the calculator will wait until $\overline{BUSY}$ is released before continuing its operations. $\overline{BUSY}$ is found on pin A3.

$\overline{STOP}$

Any device in the system can assert $\overline{STOP}$ to stop a program being executed by the calculator. $\overline{STOP}$ must be asserted until the calculator releases $\overline{BUSY}$. $\overline{STOP}$ is found on pin A2.

$\overline{ERST}$ (ERROR SET)

Asserting $\overline{ERST}$ for at least one T Time will cause the calculator display to enter a FLASHING mode. $\overline{ERST}$ can be used by an external device as a signal to inform the calculator that a given condition exists or has occured. $\overline{ERST}$ is found on pin A11.

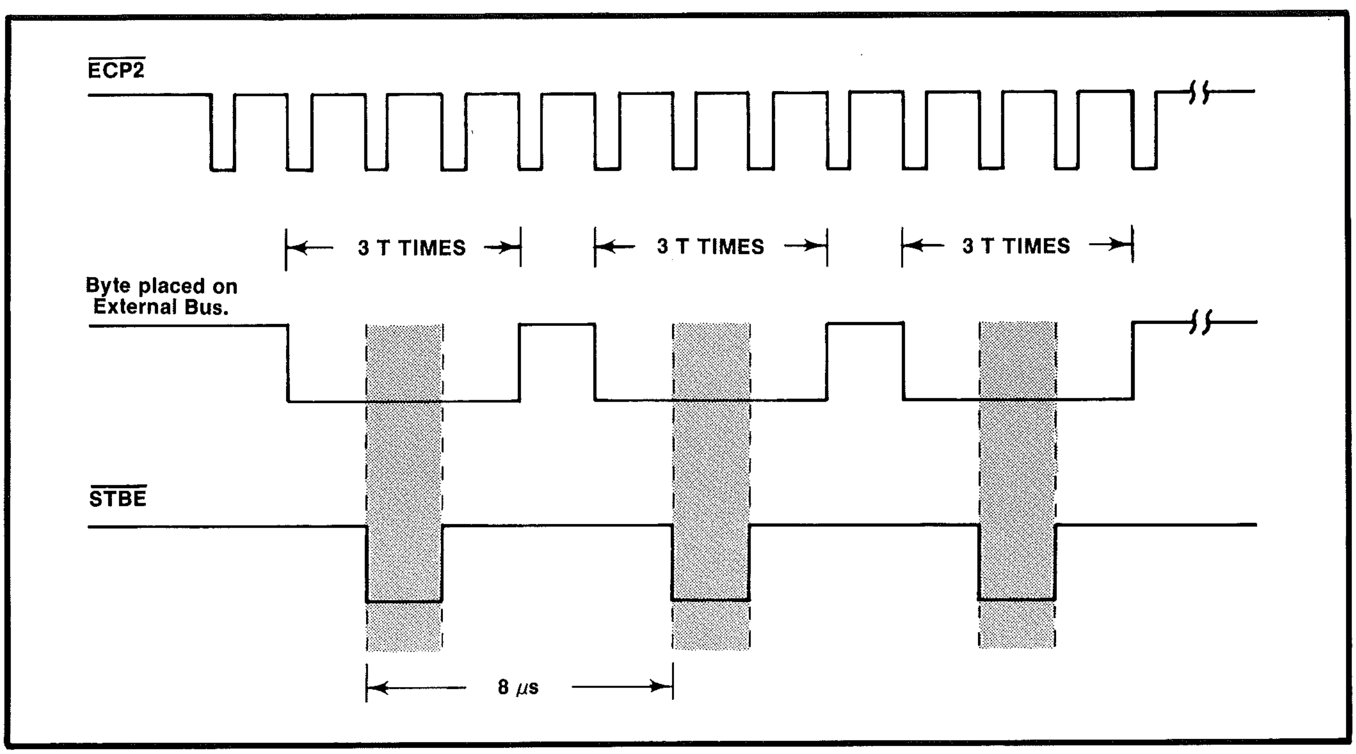


Fig. 3-7. Maximum rate of transfer on the External Bus.

$\overline{\text{EROT}}$ (ERROR FLASH)

$\overline{\text{EROT}}$ goes active (low) when the calculator display starts flashing. It is used to inform other members in the system that the display is flashing. $\overline{\text{EROT}}$ is found on pin A12.

$\overline{\text{DCLOW}}$ (DC LOW)

$\overline{\text{DCLOW}}$ is the master reset for the system on power up. It is used to reset logic circuits in an external device when the calculator power switch is turned on or when a power interruption occurs. $\overline{\text{DCLOW}}$ is found on pin A10.

$\overline{\text{PRRT}}$ (PROGRAMMABLE RESET)

$\overline{\text{PRRT}}$ is a negative-going pulse which is generated when the CLEAR key is depressed on the calculator keyboard or when CLEAR is generated under program control. $\overline{\text{PRRT}}$ is automatically made true when $\overline{\text{DCLOW}}$ is made true (near 0 VDC) on power up. $\overline{\text{PRRT}}$ is found on pin A14.

REMOTE ADDRESS SIGNAL LINES

General

As was explained in section 1, the calculator operates each external device in the system through remote control. The following text describes the remote address signal lines UD1 thru UD8, TD1 thru TD8, and $\overline{\text{AVS}}$ (ADDRESS VALID STROBE).

Depressing the REMOTE key and two digit keys on the keyboard sends a remote address to an external device. The first digit is referred to as the TENS digit; the second digit is referred to as the UNITS digit. The TENS digit is stored until the UNITS digit is entered, then they are both put on the remote addressing lines together.

Eight lines are used to carry the remote address to the external device. Four lines, TD1-TD8, carry the BCD code for the TENS digit. The other four lines, UD1-UD8, carry the BCD code for the UNITS digit. All of these signal lines are held in the high state (+3.6 VDC) when not being used. When the remote address is put on the lines, the lines representing a binary one remain high; the lines representing a binary zero go low (near 0 VDC).

Refer to Fig. 3-8. Fig. 3-8 is a timing diagram for the command "REMOTE 48". As soon as the address is entered into the calculator, the BCD code (0100) for the TENS digit (4) is placed on TD8 thru TD1, respectively. Notice that the TD4 line remains high and the other three lines go low. A short time later, the BCD code (1000) for the UNITS digit (8) is placed on the lines UD8-UD1, respectively; UD8 remains high and the other lines go low. Along with the UNITS digit, the calculator issues $\overline{\text{AVS}}$ (ADDRESS VALID STROBE). This signal indicates that the address is valid and should be decoded.

Decoding the Remote Address

Fig. 3-9 shows a typical remote address decoder. Each calculator signal line is buffered by an inverting bus receiver. The UD and TD signals are again inverted to a positive polarity before they are fed to the BCD to Decimal Decoders. The output of each decoder is equipped with a movable strap which allows the address code to be changed in different applications. The TENS digit (4) and the UNITS digit (8) are ANDED together with $\overline{\text{AVS}}$ (ADDRESS VALID STROBE) to trigger a pre-defined function.

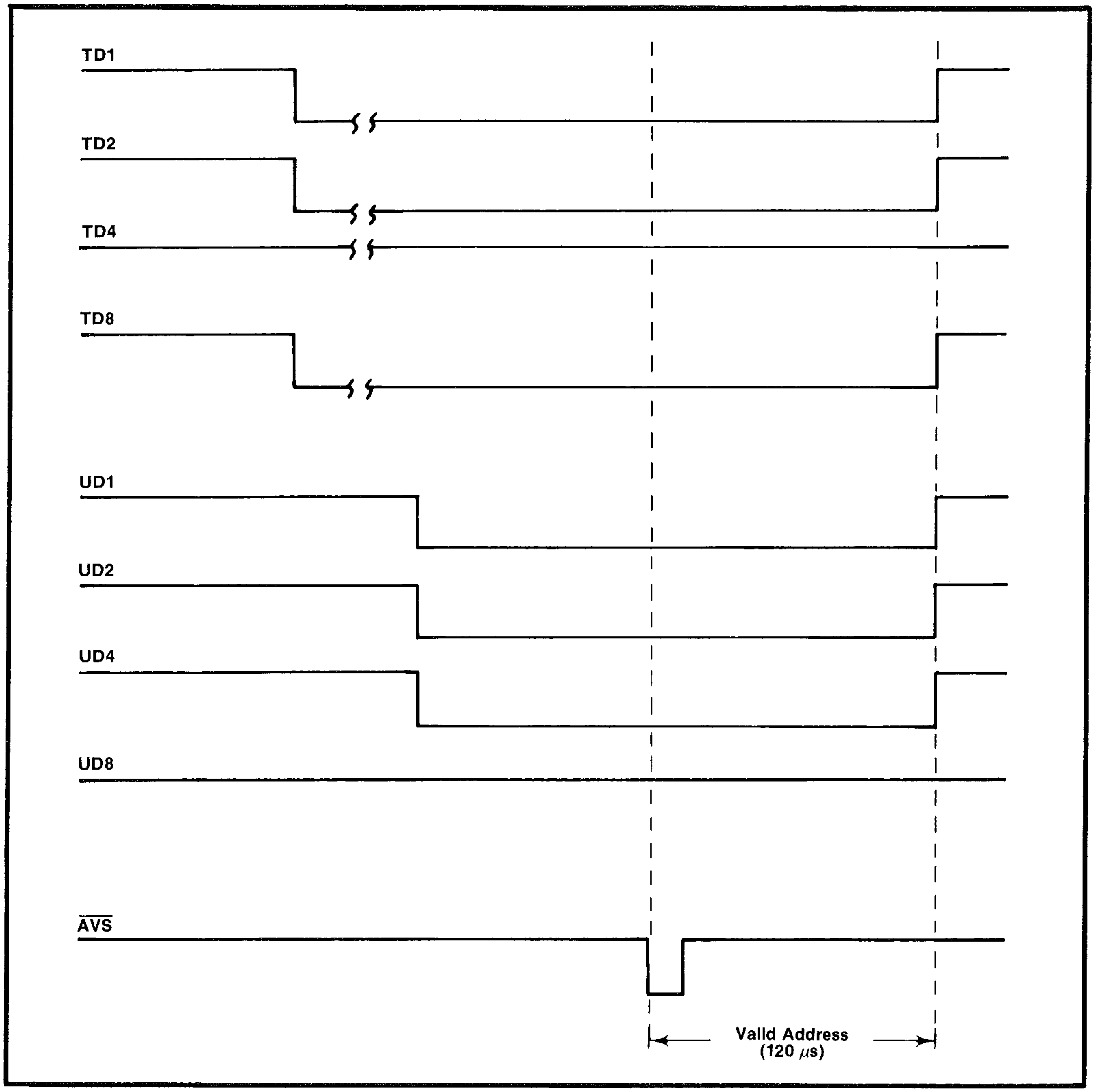


Fig. 3-8. Remote Address signal lines: "REMOTE 48".

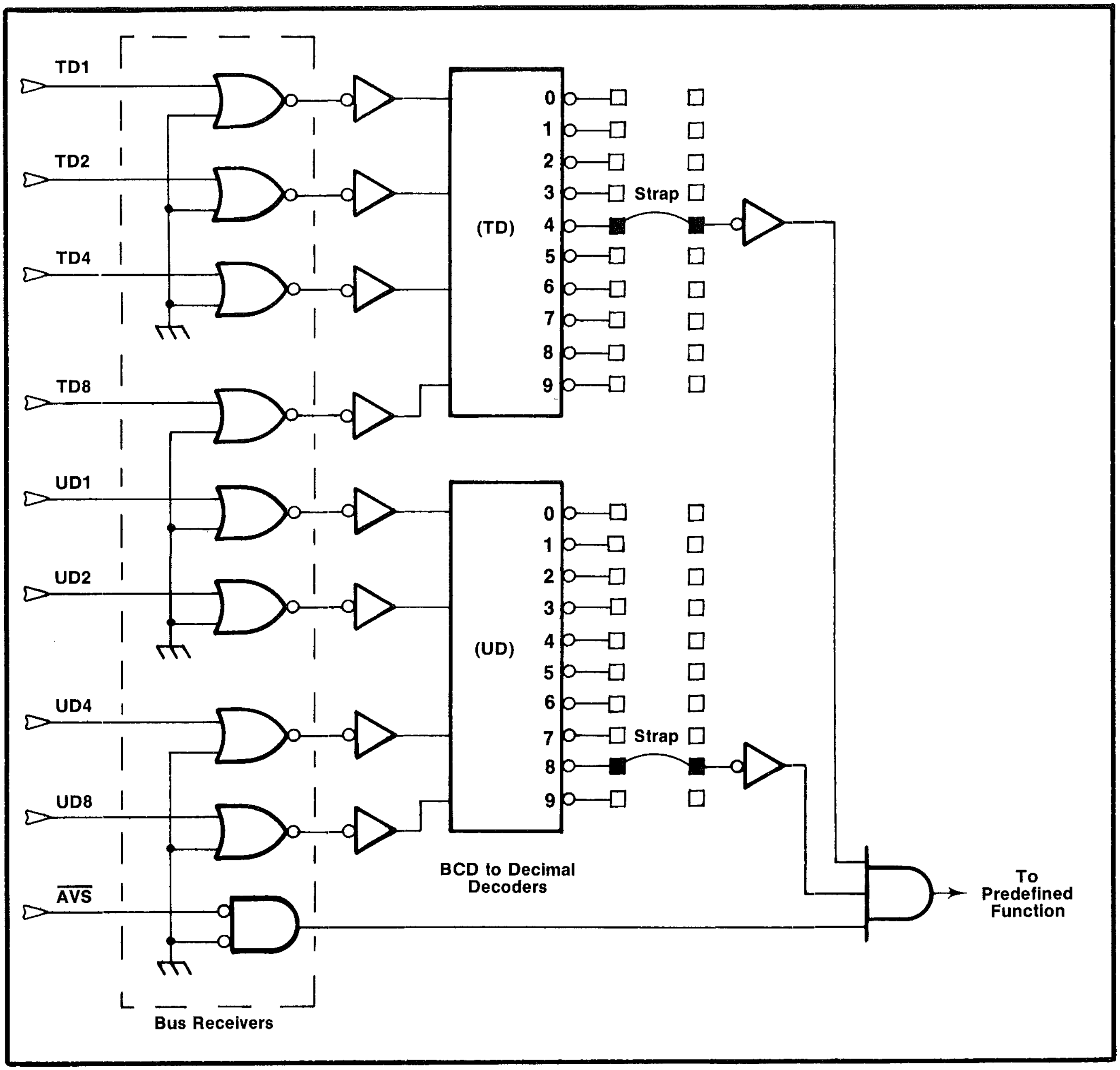


Fig. 3-9. Decoding the Remote Address.

Section 4

DATA BUS

INTRODUCTION

The Data Bus is a set of four parallel conductors used to transfer data to and from the calculator's data registers. Each transfer is preceded by an exchange agreement called a "handshake". This section describes in detail how data is transfered over the Data Bus. The following signals will be discussed: data lines $\overline{\text{DIO1}}$, $\overline{\text{DIO2}}$, $\overline{\text{DIO4}}$, $\overline{\text{DIO8}}$; handshake signals $\overline{\text{AC}}$, $\overline{\text{CRCV}}$, $\overline{\text{FXTP}}$, $\overline{\text{FLTP}}$; and a special function, $\overline{\text{DISP}}$ (see Fig. 4-1).

DATA FORMAT

Refer to Fig. 4-2. As explained previously, when an external device makes a request for data, the numeric value in the calculator display is formatted into 16 time frames and transferred in BCD bit-parallel, digit-serial format. It can be transferred in either fixed point notation or floating point notation. (Fixed point notation refers to scientific notation. Floating point notation refers to any numeric value not in scientific notation.) Each digit is represented in BCD. Status information is represented by four-bit binary numbers. Fig. 4-2 shows how the display is divided into 16 T times in preparation for the transfer. The number 6.28×10^{18} is used as an example. Status bits transferred during T0 and T1 in this example form a binary "10" when the individual bits are placed on the Data Bus.

DATA REGISTERS

Refer to Fig. 4-3. Fig. 4-3 is a diagram of the data registers located on the calculator board. All data entering and leaving the calculator is placed in these registers. The Floating Point Register (often referred to as the Display Register) can carry each data value in floating point notation or fixed point notation. The data in this register is displayed on the display board after each data entry and after each calculation. The Fixed Point Register carries the data value in scientific notation. When the calculator board processes the data, it works with the contents of this register.

Data entering the calculator from the keyboard is placed in the Fixed Point Register and the Floating Point Register as shown in Fig. 4-4(A). The number 1729 is used as an example. After the number is keyed into the calculator, it should be "normalized" before it is sent to the external device. Hitting the = key or the) key normalizes the data. When this is done, the calculator converts the value in the Fixed Point Register into scientific notation, and places a decimal point behind the T6 digit in the Floating Point Register, as shown in Fig. 4-4(B).

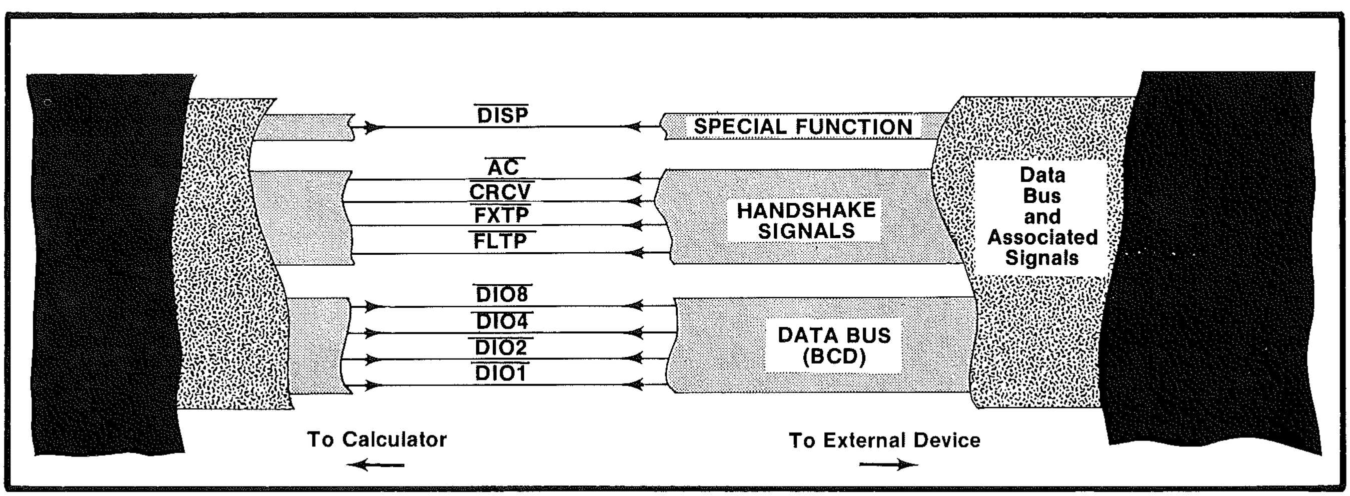


Fig. 4-1. Data Bus and associated signals.

T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
6.	2	8	0	0	0	0	0	0	0	0	0	1	8	9	10

Displayed Number (T15–T6)

Guard Digits; not Displayed (T5–T4)

Exponent; Displayed (T3–T2)

Status Bits; not Displayed (T1–T0)

Complete Mantissa (T15–T4)

DISPLAY

T0 — Status Bits

DIO1 = 0 (high) = POSITIVE EXPONENT
DIO1 = 1 (low) = NEGATIVE EXPONENT
DIO2 = 0 (high) = ORDINARY NOTATION
DIO2 = 1 (low) = SCIENTIFIC NOTATION
DIO4 = 0 (high) = POSITIVE MANTISSA
DIO4 = 1 (low) = NEGATIVE MANTISSA
DIO8 = 0 (high) = DECIMAL POINT NOT DISPLAYED
DIO8 = 1 (low) = DECIMAL POINT DISPLAYED

T1 — Identifies the decimal point location in the display, from 0 through 10. Position zero is the right-most position of the display (right side of T6 digit). Position 10 is the left-most position of the display (left side of T15 digit).

T2 — Least Significant Digit (LSD) of exponent in Binary Coded Decimal format (BCD).

T3 — Most Significant Digit (MSD) of exponent in BCD.

T4 — LSD of mantissa in BCD (not displayed).

T5 — 2nd LSD of mantissa in BCD (not displayed).

T6 — LSD of displayed number in BCD.

T7 through T14 are BCD digits of displayed number.

T15 — MSD of mantissa in BCD.

Fig. 4-2. Data Format on the Data Bus.

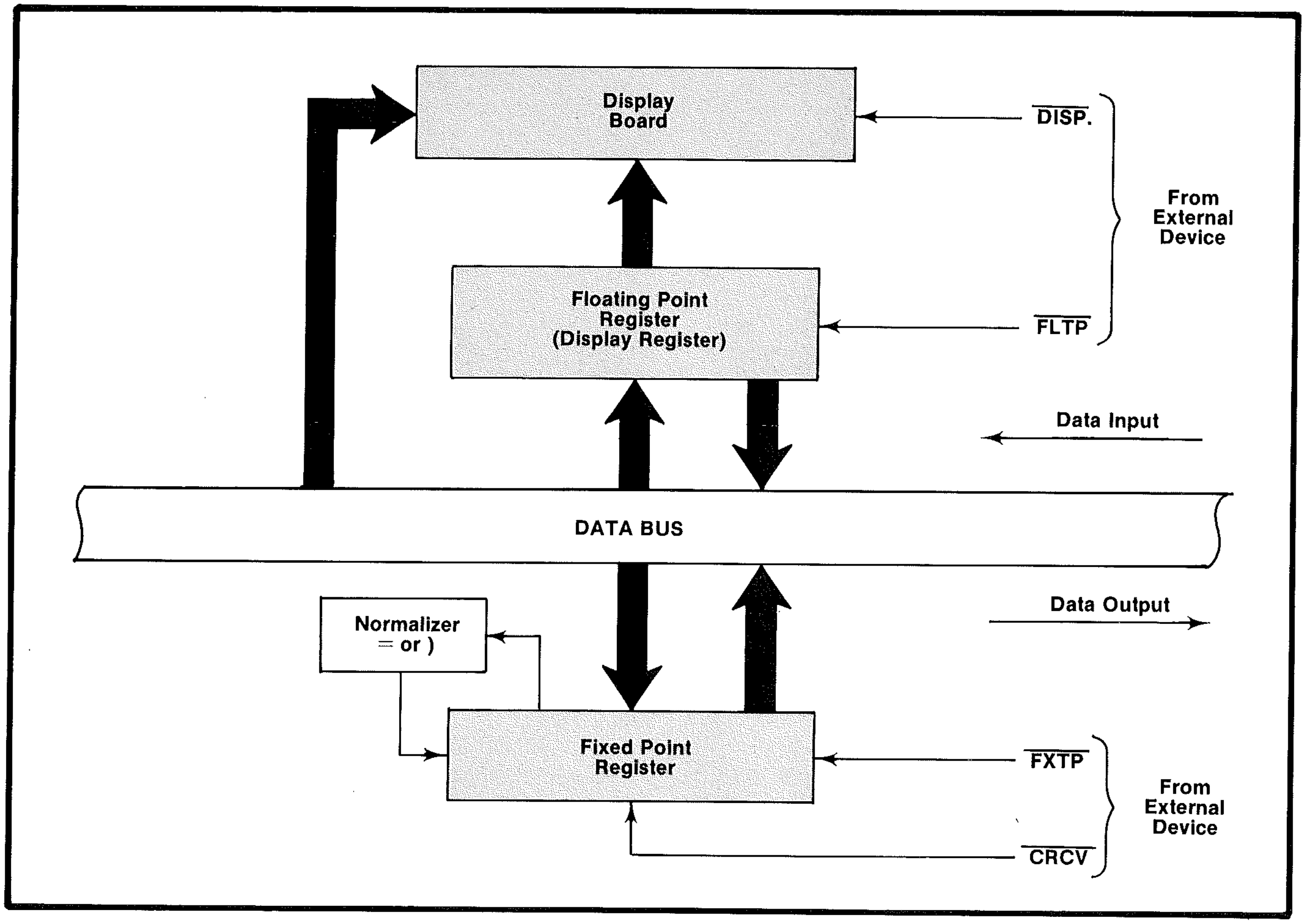


Fig. 4-3. Data Registers on the Calculator Board.

T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	
0	0	0	0	0	0	1	7	2	9	0	0	0	0	0	0	$\overline{FLTP}$ REGISTER

T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	
0	0	0	0	0	0	1	7	2	9	0	0	0	0	0	0	$\overline{FXTP}$ REGISTER

(A) Data not normalized.

T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	
0	0	0	0	0	0	1	7	2	9.	0	0	0	0	0	0	$\overline{FLTP}$ REGISTER

T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	
1.	7	2	9	0	0	0	0	0	0	0	0	0	3	9	10	$\overline{FXTP}$ REGISTER

(B) Data normalized by depressing the = key or the) key.

Fig. 4-4. Data entered in floating point notation.

	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
FLTP REGISTER	0	0	0	0	0	0	1.	7	2	9	0	0	0	3	3	10
FXTP REGISTER	0	0	0	0	0	0	1.	7	2	9	0	0	0	3	3	10

(A) Data not normalized.

	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
FLTP REGISTER	1.	7	2	9	0	0	0	0	0	0	0	0	0	3	9	10
FXTP REGISTER	1.	7	2	9	0	0	0	0	0	0	0	0	0	3	9	10

(B) Data normalized by depressing the = key or the) key.

Fig. 4-5. Data entered in scientific notation.

Refer to Fig. 4-5(A). If the data is keyed into the calculator in scientific notation, then both registers will contain the value in scientific notation as shown in Fig. 4-5(A). When the data is normalized by hitting the = key or) key, the mantissa is shifted to the left as shown in Fig. 4-5(B).

If data is entered into the calculator in scientific notation, it will be sent from the Floating Point Register in scientific notation (Fig. 4-5). It cannot be transferred in Floating Point unless it is first converted to Floating Point. (Hitting the DEG/RAD key twice will change a number in scientific notation back to floating point, provided the number is within the range of the Floating Point Register.)

CHOOSING A DATA FORMAT

Data transferred to an external device can come from either the Fixed Point Register or the Floating Point Register. The external device must indicate its choice during the handshake before the transfer begins. Refer to Fig. 4-3. If data is desired in scientific notation, then the receiving device asserts FXTP during the handshake and the contents of the Fixed Point Register are sent. If the receiving device wants the data as it appears in the display, then it asserts FLTP during the handshake and the contents of the Floating Point Register are sent. It is a question as to whether one must keep track of an exponent (fixed point) or a decimal point (floating point).

It is up to the operator to insure that the data is entered into the data registers in the proper format. If the data in the Fixed Point Register is not normalized by hitting the = key or the) key, it cannot be transferred in scientific notation; it will be transferred from the Fixed Point Register in the same format as it was keyed in, no matter what the format may be. See Fig. 4-4(A).

Refer to Fig. 4-3. Data entering the calculator from an external device is put into both the Fixed Point Register and the Floating Point Register. The external device asserts CRCV and FXTP during the handshake. Asserting FXTP does not mean that the data has to be formatted in scientific notation; it can be formatted in any manner. The data is automatically normalized once calculations resume.

It might be noted that if during a calculation the range of the Floating Point Register is exceeded, the contents will automatically shift to scientific notation. If an operation is done which causes a number in floating point notation to shift to fixed point notation, and this is followed by an operation which causes the number to come back within the range of the Floating Point Register, the number will return to floating point notation.

SENDING MESSAGES DIRECTLY TO THE DISPLAY BOARD

Refer to Fig. 4-3. The signal line $\overline{DISP}$ (DISPLAY) is used by an external device to send numeric coded messages directly to the display board. When $\overline{DISP}$ is activated during the handshake, the BCD digits by-pass the data registers and go directly to the display board. The external device must send the BCD digits over the Data Bus one at a time, starting at T0 and ending at T15. Any digit equal to a binary 15 (1111) appears as a blank. $\overline{DISP}$ can be activated anytime before T0 and must be held until after T15. If the external device is asserting $\overline{BUSY}$ while sending the message, it must release $\overline{BUSY}$ before it releases $\overline{DISP}$. Otherwise, the calculator overwrites the information with the contents of the Floating Point Register on the trailing edge of $\overline{BUSY}$.

THE HANDSHAKE

General

An exchange agreement called a handshake is required before each data transfer with an external device. The handshake determines which device is going to send data and what the data format will be. The handshake is initiated by the calculator through a remote command. The remote address triggers circuits on the interface which activate signal lines on the I/O connector. These I/O signals determine the direction of the transfer and the data format.

Data transferred in different formats must be assigned different remote addresses. For example, the designer could assign the address REMOTE 62 to receive data from the calculator in fixed point, REMOTE 63 to receive data from the calculator in floating point, and REMOTE 64 to send data to the calculator.

Receiving Data from the Calculator

Fig. 4-6(A) is a graphic illustration of the handshake required for a calculator peripheral to receive data from the calculator. The handshake begins when the calculator issues a pre-defined remote address along with $\overline{AVS}$ (ADDRESS VALID STROBE). The peripheral interface responds with $\overline{AC}$ (ADDRESS COMPARE), and either $\overline{FXTP}$ (FIXED POINT) or $\overline{FLTP}$ (FLOATING POINT). The calculator then issues $\overline{DW}$ (DATA WORD) and puts the data on the Data Bus during the period T0 to T15. (See Fig. 3-4 for timing details.) $\overline{DW}$ lets the receiving device know when the requested data is being placed on the Data Bus.

If the peripheral device is busy on another task when the remote command is given, it can assert $\overline{AC}$ then finish the task before asserting $\overline{FXTP}$ or $\overline{FLTP}$. Asserting $\overline{AC}$ lets the calculator know that the device is present. The calculator then holds in a busy condition until $\overline{FXTP}$ or $\overline{FLTP}$ is asserted. If $\overline{AC}$ is not asserted immediately after $\overline{AVS}$ (within 50 μs), the calculator assumes that the device is not present and does not issue $\overline{DW}$.

If $\overline{FLTP}$ is asserted by the calculator peripheral during the handshake, the calculator sends the contents of the Floating Point Register only once. At the end of $\overline{DW}$ the remote address is reset by the calculator, and the device must release $\overline{AC}$ and $\overline{FLTP}$. If, however, $\overline{FXTP}$ is asserted during the handshake, the calculator keeps sending the contents of the Fixed Point Register on every calculator word and its accompanying $\overline{DW}$ until $\overline{FXTP}$ is released. If $\overline{FXTP}$ is released in the middle of a transmission, the calculator finishes that transmission before going on to another task.

Sending Data to the Calculator

Refer to Fig. 4-6(B). A data transfer to the calculator must be initiated by the calculator (usually under program control). The calculator starts the handshake by putting a pre-defined remote address on the lines and issues $\overline{AVS}$. The peripheral device responds with $\overline{AC}$, $\overline{CRCV}$, and $\overline{FXTP}$. $\overline{CRCV}$ and $\overline{AC}$ must be asserted no later than 50 μs after $\overline{AVS}$. $\overline{FXTP}$ can be delayed if the device is busy on another task. Although $\overline{FXTP}$ must be asserted when sending data to the calculator, the data can be formatted in any manner. After $\overline{FXTP}$ is asserted, the calculator issues $\overline{DW}$. The peripheral device puts the data on the Data Bus during this time period (T0 through T15). $\overline{FXTP}$ must then be released before T15 or the calculator will re-issue $\overline{DW}$.

GENERATING T TIME PULSES ON THE INTERFACE

If data is sent to an external device, it is up to that device to take the information off the lines in the right time frames. T Time enable pulses can be generated on the interface by using $\overline{ESYNC}$ and $\overline{ECP2}$ to clock a shift register as shown in Fig. 4-7. The combination of $\overline{ESYNC}$ and $\overline{ECP2}$ sets the T0 output high. The combination of the high T0 output and the next $\overline{ECP2}$ pulse sets the T1 output high, and so on. This ripple effect continues through T15. The process starts over on the next calculator word when $\overline{ESYNC}$ and $\overline{ECP2}$ again clock the T0 output high. As each T Time pulse is generated, it enables receiver circuits to take the information off the bus in that particular time frame. The T Time generator runs continually and generates a set of T Time pulses for every calculator word.

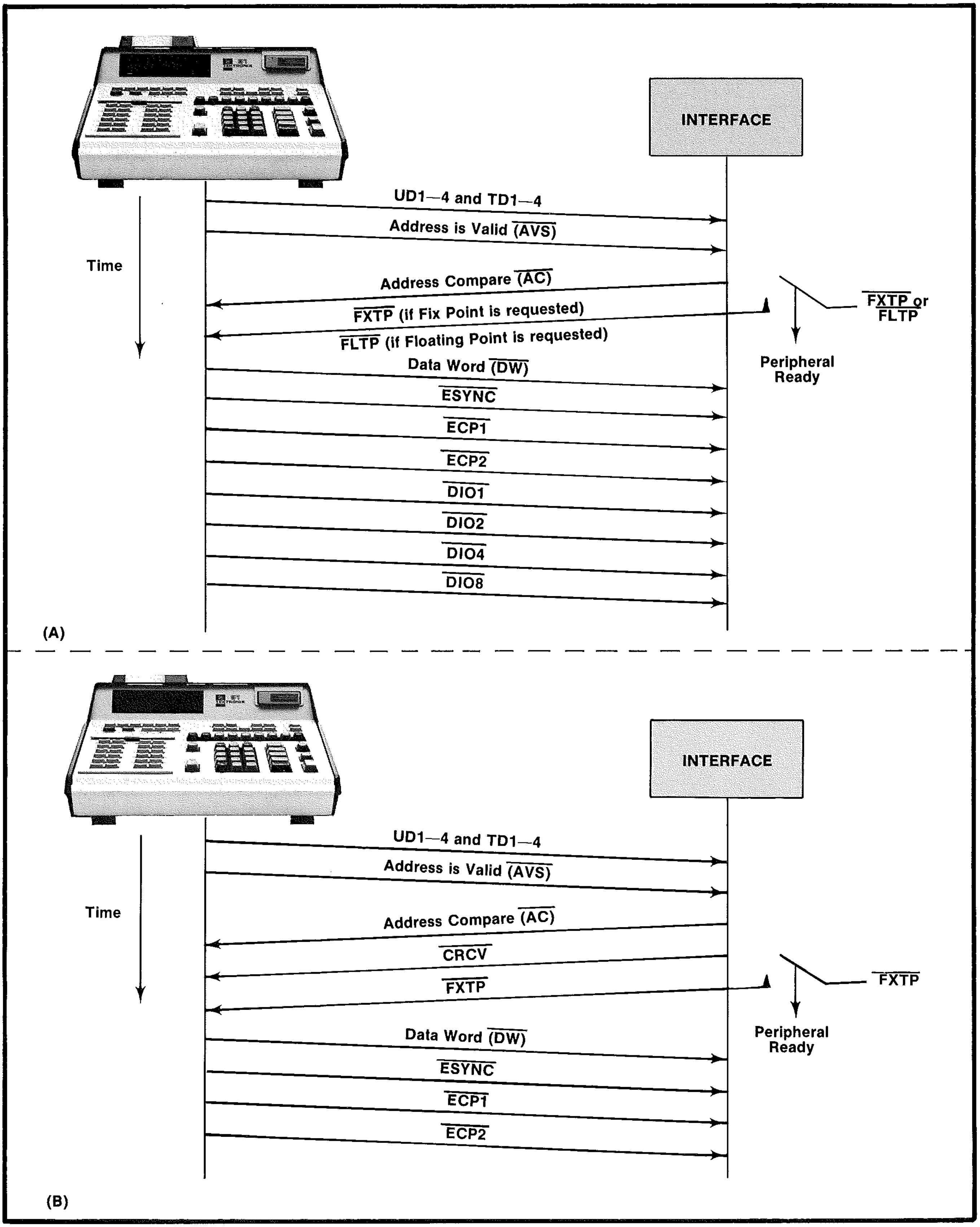


Fig. 4-6. Handshake Sequence.

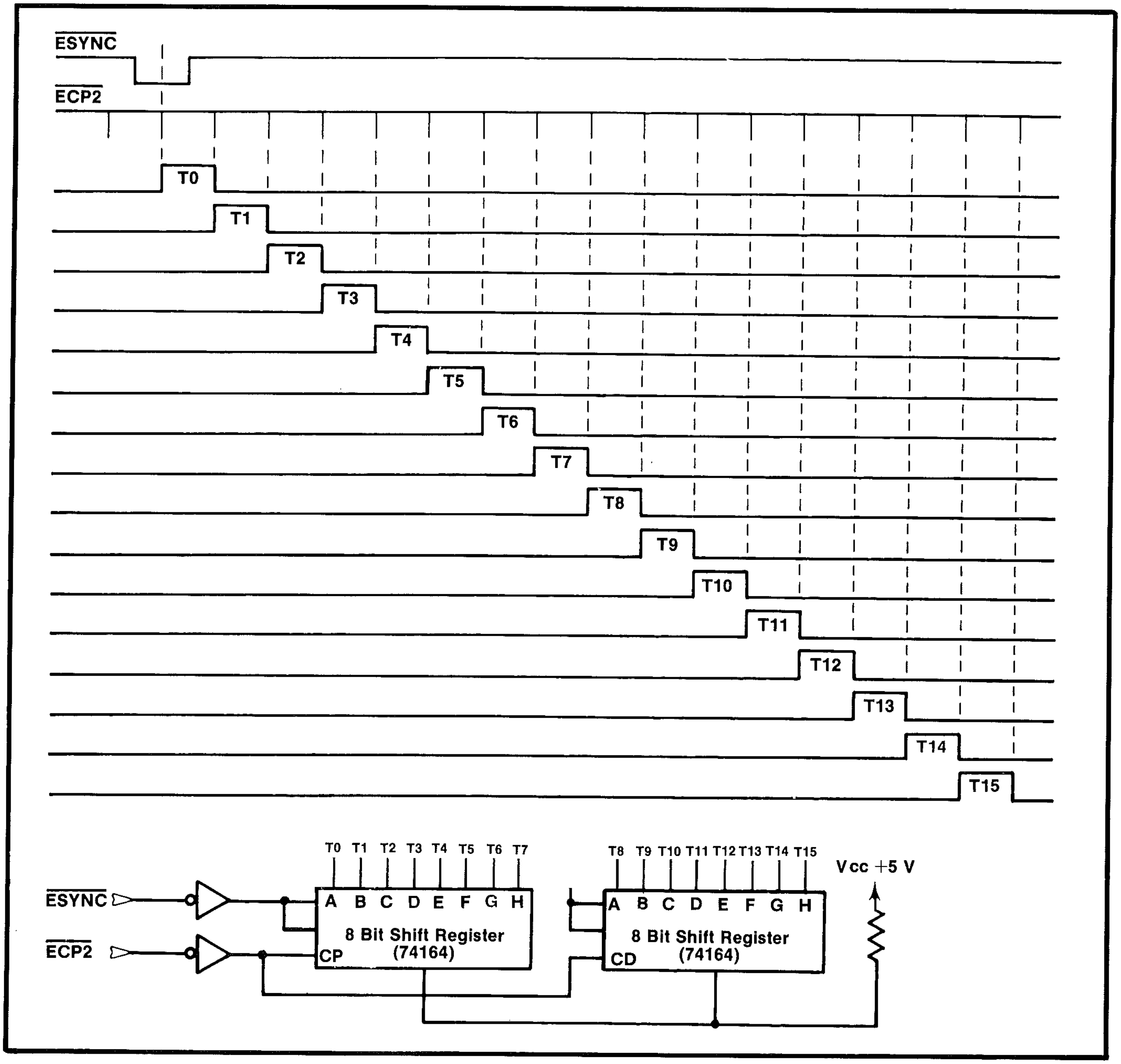


Fig. 4-7. T Time Generator.

TYPICAL INTERFACE CIRCUITS FOR DATA TRANSFER

Calculator Send Data

General. Refer to pullout sheet Fig. 6-3 CALCULATOR SEND DATA in section 6, and the timing diagram Fig. 4-8. The Calculator Send Data schematic and timing diagram show how a number can be entered into the calculator display and transferred to an interface to preset a down counter. The schematic diagram is divided into three sections: Handshake Circuitry, T Time Generator, and BCD Down Counter.

Handshake Circuitry. The handshake circuitry includes the bus receivers U3B-C, U4A-C, and U5A-D; inverters U6A-D and U7A-D, BCD to decimal decoders U8 and U9, D-type flip-flop U10, and bus drivers U11A and U11B.

When the calculator issues a remote command, the remote address signal lines are activated. The BCD code for the UNITS digit is received by bus receivers U3B-C, U4A-B, inverted by inverters U6A-U6D, and decoded by the BCD-to-decimal decoder U8. In this case, the output "6" has been selected for a function code. In a similar manner, the TENS digit is received by bus receivers U4C, U5A-C, inverted by inverters U7A-U7D, and decoded by BCD-to-decimal decoder U9. The TENS digit "1" has been selected as the device code.

If the calculator addresses this function of the interface, it puts the code "16" on the lines. Output 1 of the TENS digit decoder is inverted by U7C and applied as a high to flip-flop U10 (point①). Shortly thereafter, the calculator issues $\overline{\text{AVS}}$ (ADDRESS VALID STROBE). $\overline{\text{AVS}}$ clocks flip-flop U10 (point②), causing the Q output to go high and remain high. The high Q output is inverted by U11A which asserts $\overline{\text{AC}}$ (ADDRESS COMPARE) (point③). $\overline{\text{FLTP}}$ (point⑤) is also asserted via U11B unless the external device is asserting BUSY ($\overline{\text{BZY}}$, point④). If $\overline{\text{BZY}}$ is active, $\overline{\text{FLTP}}$ is asserted as soon as $\overline{\text{BZY}}$ is released. Shortly after $\overline{\text{FLTP}}$ is asserted, the calculator issues a 32 μs time window $\overline{\text{DW}}$ (DATA WORD) and sends the contents of the Floating Point Register (displayed value). This transfer is discussed following the discussion on the T Time Generator and the BCD Down Counter.

T Time Generator. The T Time Generator consists of two shift registers, U14 and U15. The inputs $\overline{\text{ESYNC}}$ and $\overline{\text{ECP2}}$ come from the calculator via bus receivers U2A and U2B. As was explained earlier, the T Time Generator generates a set of 16 T Time pulses during each calculator word. Only the T6, T7, T8 and T9 pulses are used in this example. The 2 μs pulses enable NAND gates U16A, U16B, U16C and U16D during the proper T time.

BCD Down Counter. The BCD Down Counter consists of U17, U18, U19, and U20 connected in cascade. When a 4-bit BCD code is placed on the input to one of these counters, and the LOAD (L) pin goes low, the output of the counter is set to the same value (in BCD). The $\overline{\text{COUNT DOWN}}$ input from the external device (top of U17) clocks the counter down to zero. When the counter reaches zero, the $\overline{\text{COUNT = 0}}$ line (BOR, bottom of U20) goes low, which signals the external device that the count is zero. (In other applications, these devices can be used to capture and hold BCD digits from the the calculator until the external device is ready to receive them.)

The Transfer. The transfer starts after $\overline{\text{FLTP}}$ (FLOATING POINT) is asserted by the external device and the calculator issues $\overline{\text{DW}}$ (DATA WORD). $\overline{\text{DW}}$ is ANDED at U13A with the combined address from U12 to form a 32 μs positive going pulse called UNIT WORD (point⑥). This design insures that $\overline{\text{DW}}$ activates only the receiving circuits being addressed by the calculator. UNIT WORD enables U13B which allows the $\overline{\text{ECP1}}$ clock pulses to pass through and go to the four gates, U16A-D (point⑦).

$\overline{\text{ECP1}}$ clocks the LOAD (L) inputs on each BCD down counter. Notice in the timing diagram (waveform⑦) that the rising edge of $\overline{\text{ECP1}}$ occurs in the middle of each T Time.

At the beginning of UNIT WORD (point⑥) the calculator starts placing display status information on the Data Bus. At T6 the BCD digit "6" is placed on the Data Bus. The T6 pulse from the T Time Generator (U14) enables gate U16A and the leading edge of $\overline{\text{ECP1}}$ clocks U17, which "captures" the BCD digit. The output of U17 is set to this value (in BCD). At T7, the calculator puts the BCD "9" on the lines. The T7 pulse from U14 enables U16B and the leading edge of $\overline{\text{ECP1}}$ (point⑦) clocks U18 which captures the BCD "9". This sequence repeats until the counter output is set to "1496".

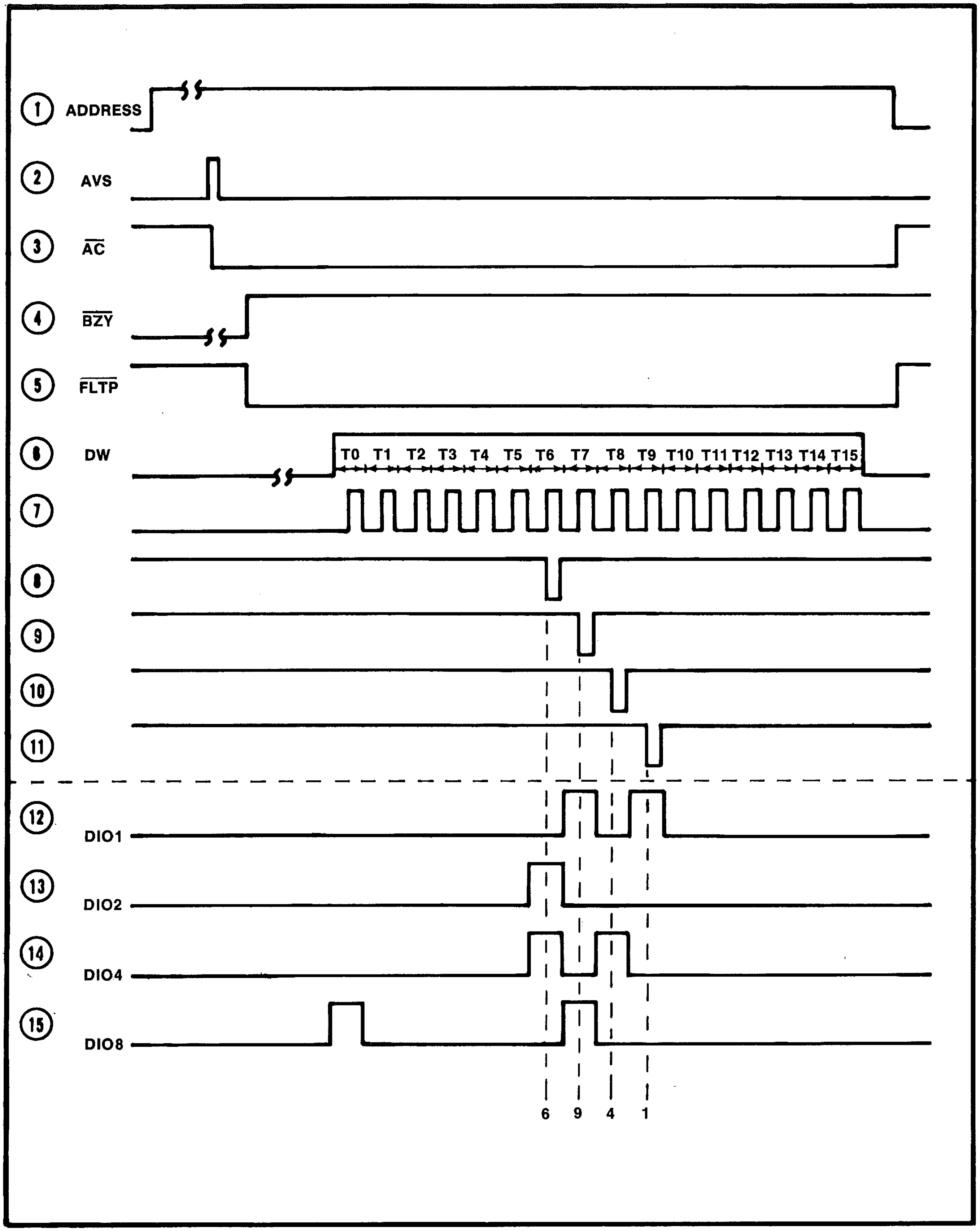


Fig. 4-8. Calculator Send Data. Keystroke Sequence: CLEAR, 1496=REMOTE 16.

Calculator Receive Data

General Refer to pullout sheet Fig. 6-4 CALCULATOR RECEIVE DATA and timing diagram Fig. 4-9. The Calculator Receive Data schematic and timing diagram show how an external device can enter data into the calculator's data registers via the Data Bus. Data put in these registers is displayed on the display board. The schematic is divided into three sections: Handshake Circuitry, T Time Generator, and Data Bus Drivers.

Handshake Circuitry. The handshake begins when the calculator issues the remote address "17" to the interface. The UNITS digit (BCD "7") is decoded by U6; output 7 goes low. The TENS digit (BCD "1") is decoded by U7; output 1 goes low. The output of U7 is inverted by U5C and applied to the D inputs of U8A and U8B (point①). Shortly thereafter, the $\overline{AVS}$ (ADDRESS VALID STROBE) from the calculator (point②) clocks both flip-flops and their Q outputs go high and remain high.

The high output of U8A is applied to bus drivers U9A and U9B to assert $\overline{AC}$ (ADDRESS COMPARE point③), and $\overline{CRCV}$ (CALCULATOR RECEIVE point④). The high Q output of U8B causes the output of bus driver U9C to go low unless $\overline{BZY}$ (point⑤) is asserted by the external device. If $\overline{BZY}$ is active, as shown in the timing diagram, $\overline{FXTP}$ is asserted as soon as $\overline{BZY}$ is released.

After $\overline{FXTP}$ is asserted by the external device, the calculator issues $\overline{DW}$ (DATA WORD) and the transfer begins. The transfer is discussed following the discussion on the T Time Generator and the Data Bus Drivers.

T Time Generator. The T Time Generator is the same configuration as shown in Fig. 4-7. Input signals $\overline{ESYNC}$ and $\overline{ECP2}$ cause shift registers U12 and U13 to generate a series of 16-2 μs pulses from T0 to T15. Only pulses T6, T7, T8 and T9 are used in this example.

Data Bus Drivers. The Data Bus Drivers U15A-D, U16A-D, U17A-D and U18A-D place BCD digits on the Data Bus. A set of four drivers are enabled by a pulse from the T Time Generator.

The Transfer. As mentioned previously, $\overline{DW}$ is asserted by the calculator for 32 μs after the external device asserts $\overline{FXTP}$. $\overline{DW}$ is used for two purposes in this diagram. First, $\overline{DW}$ is ANDED at U11 with the combined address from U10 to form a positive going pulse called UNIT WORD (point⑦). This design insures that $\overline{DW}$ activates only the circuits which are addressed by the calculator. UNIT WORD enables U14A-D to pass the T Time pulses to the Data Bus Drivers. Secondly, $\overline{DW}$ is inverted by U5D and applied to the CLEAR (CL) input on flip-flop U8B. The leading edge of $\overline{DW}$ clears U8B, which causes the Q output to go low. This releases the $\overline{FXTP}$ signal line so $\overline{DW}$ won't be re-issued on the next calculator word.

The external device must have the BCD digits ready for transfer before $\overline{FXTP}$ is asserted. The digits are placed on the lines in the upper left corner of the schematic. Digits 5872 are used as an example.

With gates U14A-D enabled by UNIT WORD, the T6 pulse is applied to Data Bus Drivers U15A-D for 2 μs. This puts the BCD digit "2" on the Data Bus during T Time 6. The T7 pulse follows, which enables Data Bus Drivers U16A-D to put the BCD digit "7" on the Data Bus during T Time 7. This sequence continues for T8 and T9. When the transfer is complete, the digits appear in the calculator display as shown in the illustration on the schematic.

With $\overline{FXTP}$ released, the calculator takes the remote address off the lines and flip-flop U8A is cleared, which releases $\overline{AC}$ and $\overline{CRCV}$.

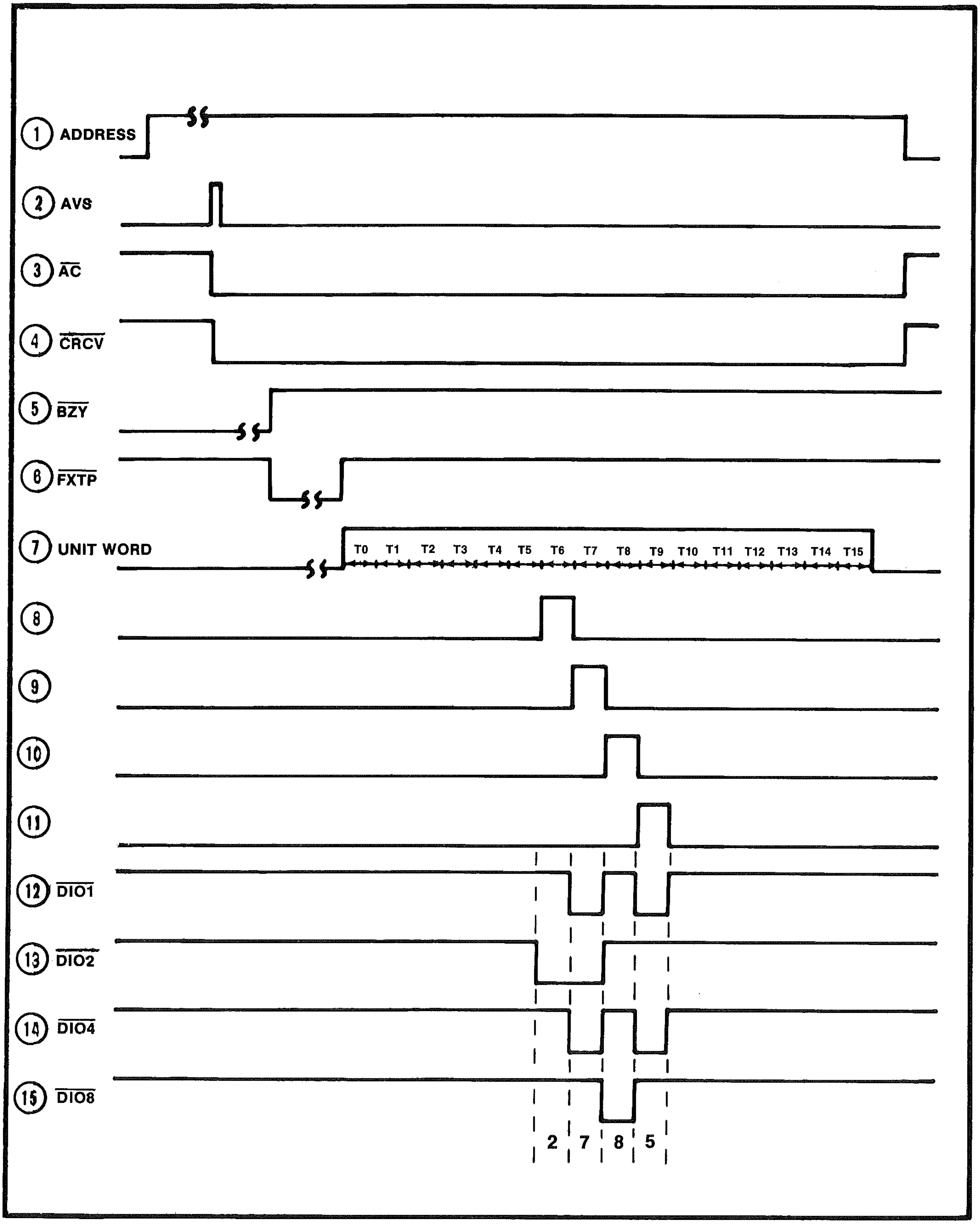


Fig. 4-9. Calculator Receive Data. Keystone Sequence: CLEAR, REMOTE 17.

Section 5
EXTERNAL BUS

INTRODUCTION

This section describes the External Bus and Associated Signal Lines. The External Bus is used to transfer operational keycodes, alphanumeric keycodes, program steps, and R-register data in parallel format. Refer to Fig. 5-1. The following signals will be discussed: External lines $\overline{\text{EXT1}}$-$\overline{\text{EXT128}}$, External Bus Management signals $\overline{\text{IHIN}}$ and $\overline{\text{INHOUT}}$, Direct Memory Access Control Signals $\overline{\text{DMAWR}}$, $\overline{\text{DMARD}}$, and $\overline{\text{DMARY}}$, and two Special Functions, $\overline{\text{CALL}}$ and $\overline{\text{EXTPTR}}$.

OPERATIONAL KEYCODES

Any device in the system which is used as an external keyboard, such as a display terminal or a card reader, is connected to the calculator via the External Bus. Keystrokes are represented by seven bit ASCII codes which are placed on lines $\overline{\text{EXT1}}$-$\overline{\text{EXT64}}$. The external device asserts $\overline{\text{STBE}}$ for 2 μs after the keycode is placed on the Bus. This causes the calculator to read the code, enter the Busy mode, and execute the instruction.

Operational keycodes for the calculator are the same as ASCII character codes, but have been re-defined for use within the calculator. A listing of the codes can be found in Tables 5-1 and 5-2. Each binary code is also represented by an octal number (octal coded binary). Refer to Fig. 5-2 for translating the octal code into its binary equivalent.

ALPHANUMERIC KEYCODES

Refer to Fig. 5-2. The TEKTRONIX 31 Calculator interprets a seven bit ASCII code as an alphanumeric character if $\overline{\text{EXT128}}$ is asserted (made TRUE). $\overline{\text{EXT128}}$ can be asserted by the external device or by depressing the HOLD FOR ALPHA key on the calculator keyboard. Notice that when the eighth bit is made true, the octal code increases by a value of "200". The only difference between an operational code and its alphanumeric counterpart is the addition of the $\overline{\text{EXT128}}$ bit.

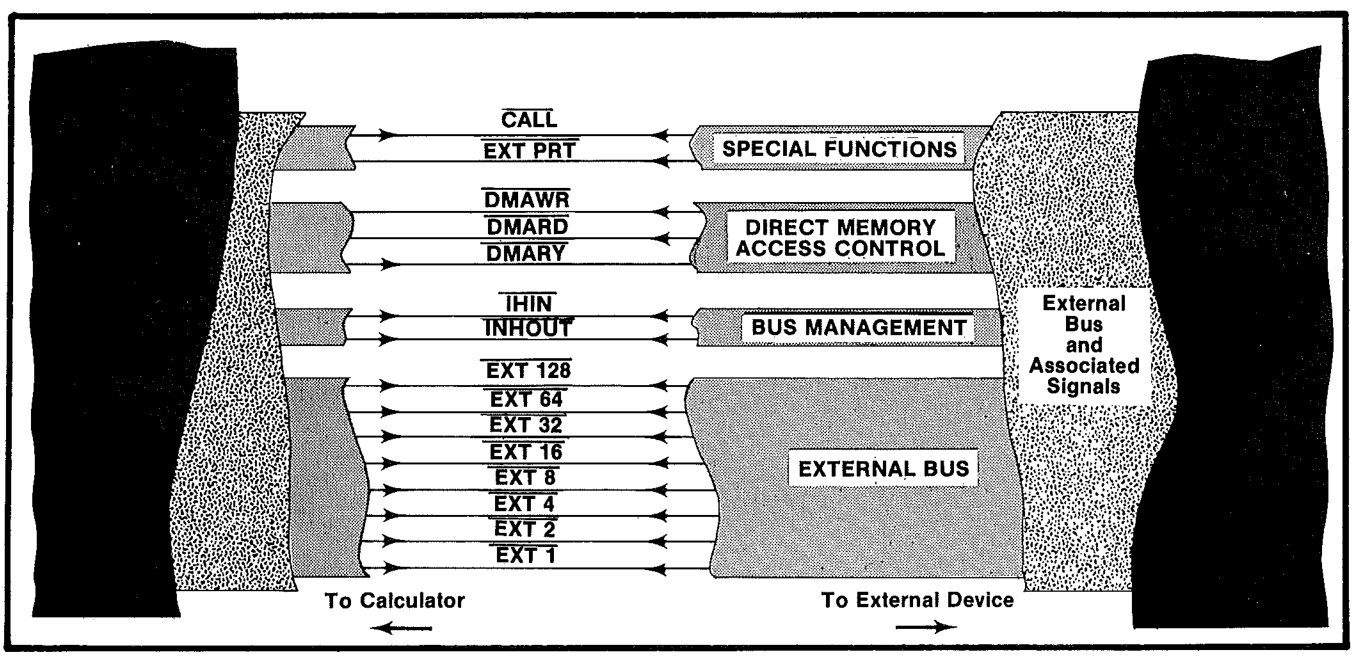


Fig. 5-1. External Bus and associated signals.

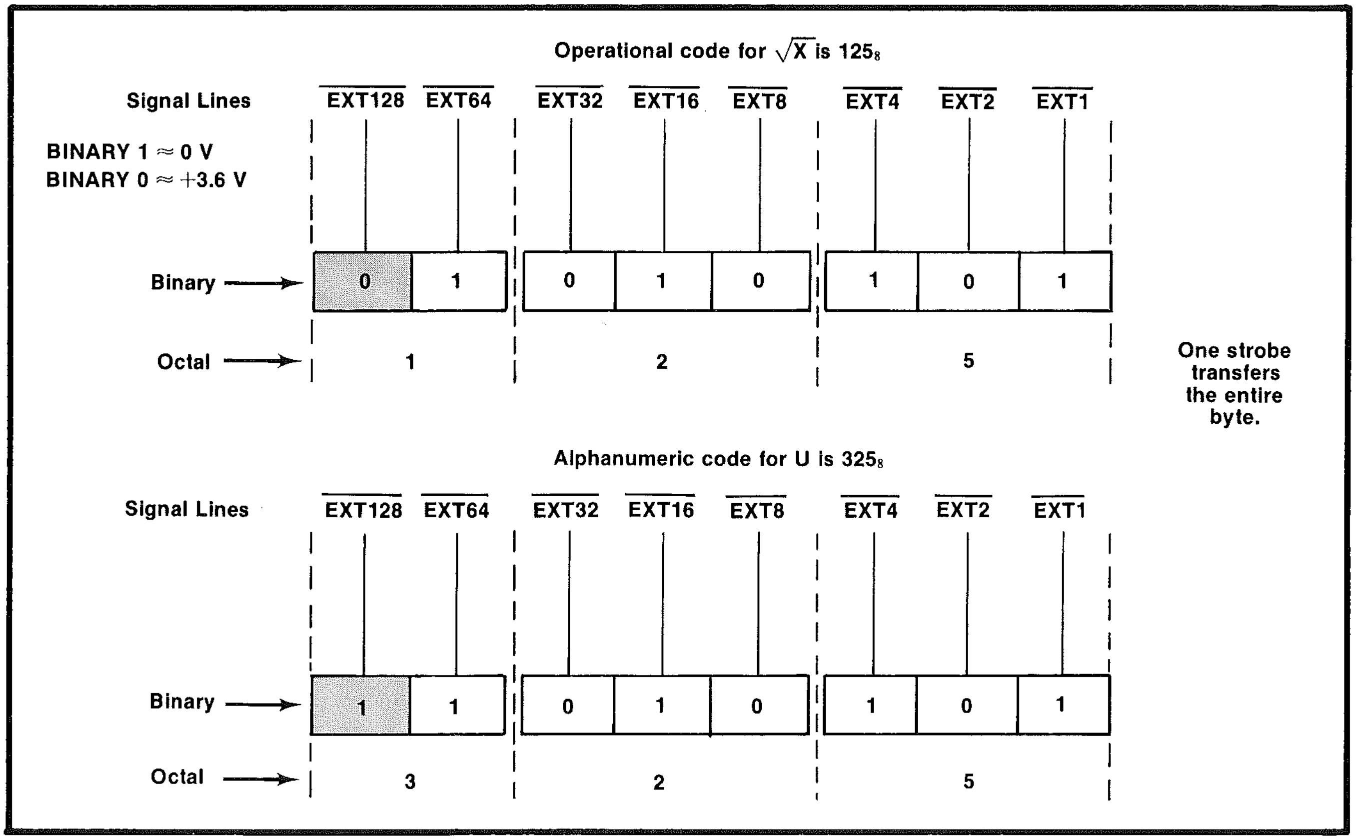


Fig. 5-2. Octal code translated into binary.

Refer to Table 5-1. Keystrokes with octal codes from 000-037 are ASCII control characters and are not assigned an alphanumeric symbol. The calculator does not print these ASCII control character symbols when the HOLD FOR ALPHA key is depressed, however, the octal codes 200-237 are placed on the External Bus. In addition, these codes (200-237) can be stored in memory and issued to an external device under program control.

Refer to Fig. 6-1 TEKTRONIX 31 Calculator System Block Diagram. When $\overline{\text{EXT128}}$ is activated, the input to the calculator board is inhibited. If the calculator is in the LEARN mode when the HOLD FOR ALPHA key is depressed, the programmer board takes each keycode entry off the External Bus and places it in memory. If the calculator is not in the LEARN mode, the programmer board places each keycode in temporary storage until 16 are gathered, then sends all 16 to the printer at once. ASCII control character codes (octal 200-237) are ignored.

TIMING

A remote keyboard can send an instruction to the calculator if the calculator is not busy. It takes three T Times (6 μs) to issue the instruction. The keycode is placed on the bus during the first T Time, "strobed" during the second T Time, and must remain on the bus for a third T Time. The selection of these T Times is arbitrary; that is, any three cycles of $\overline{\text{ECP2}}$ can be used.

New codes can be issued by an external device as soon as the calculator finishes executing the present instruction and releases $\overline{\text{BUSY}}$ and $\overline{\text{INHOUT}}$. The maximum rate for "strobing" the bus is one strobe every 8 μs. For timing diagrams, refer to Fig. 3-6 and 3-7 in section 3.

TABLE 5-1

TEKTRONIX 31 CALCULATOR KEYCODES

KEY	HOLD FOR ALPHA	OPERATIONAL CODE									ALPHANUMERIC CODE									ASCII SYMBOL
		OCTAL	BINARY								OCTAL	BINARY								
			128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1	
LABEL ■		001	0	0	0	0	0	0	0	1	201	1	0	0	0	0	0	0	1	SOH
FROM TAPE ■		002	0	0	0	0	0	0	1	0	202	1	0	0	0	0	0	1	1	STX
TO TAPE ■		003	0	0	0	0	0	0	1	1	203	1	0	0	0	0	0	1	1	ETX
EXECUTE ■		004	0	0	0	0	0	1	0	0	204	1	0	0	0	0	1	0	0	EOT
CLEAR R FILE ■		005	0	0	0	0	0	1	0	1	205	1	0	0	0	0	1	0	1	ENQ
GO TO ■■■■	BELL	007	0	0	0	0	0	1	1	1	207	1	0	0	0	0	1	1	1	BEL
R ■■■	SPACE	010	0	0	0	0	1	0	0	0	210	1	0	0	0	1	0	0	0	BS
R ■■	TAB	011	0	0	0	0	1	0	0	1	211	1	0	0	0	1	0	0	1	HT
* ← STEP		020	0	0	0	1	0	0	0	0	220	1	0	0	1	0	0	0	0	DLE
* INSERT		021	0	0	0	1	0	0	0	1	221	1	0	0	1	0	0	0	1	DC1
* DELETE		022	0	0	0	1	0	0	1	0	222	1	0	0	1	0	0	1	0	DC2
* → STEP		023	0	0	0	1	0	0	1	1	223	1	0	0	1	0	0	1	1	DC3
* LIST		024	0	0	0	1	0	1	0	0	224	1	0	0	1	0	1	0	0	DC4
*DISPLAY PROGRAM		025	0	0	0	1	0	1	0	1	225	1	0	0	1	0	1	0	1	NAK
* LEARN		026	0	0	0	1	0	1	1	0	226	1	0	0	1	0	1	1	0	SYN
CLEAR DSPLY	SPACE	040	0	0	1	0	0	0	0	0	240	1	0	1	0	0	0	0	0	SP
FLASH	!	041	0	0	1	0	0	0	0	1	241	1	0	1	0	0	0	0	1	!
SET FLAG	"	042	0	0	1	0	0	0	1	0	242	1	0	1	0	0	0	1	0	"
STOP	#	043	0	0	1	0	0	0	1	1	243	1	0	1	0	0	0	1	1	#
PRINT DSPLY	$	044	0	0	1	0	0	1	0	0	244	1	0	1	0	0	1	0	0	$
CLEAR	%	045	0	0	1	0	0	1	0	1	245	1	0	1	0	0	1	0	1	%
PAUSE	&	046	0	0	1	0	0	1	1	0	246	1	0	1	0	0	1	1	0	&
CLEAR FLAG	/	047	0	0	1	0	0	1	1	1	247	1	0	1	0	0	1	1	1	/
(	(	050	0	0	1	0	1	0	0	0	250	1	0	1	0	1	0	0	0	(
)	)	051	0	0	1	0	1	0	0	1	251	1	0	1	0	1	0	0	1	)
X	*	052	0	0	1	0	1	0	1	0	252	1	0	1	0	1	0	1	0	*
+	+	053	0	0	1	0	1	0	1	1	253	1	0	1	0	1	0	1	1	+
RESET	,	054	0	0	1	0	1	1	0	0	254	1	0	1	0	1	1	0	0	,

***Operational Code Not Programmable.**

TABLE 5-1 (cont)

TEKTRONIX 31 CALCULATOR KEYCODES

KEY	HOLD FOR ALPHA	OPERATIONAL CODE OCTAL	BINARY 128	64	32	16	8	4	2	1	ALPHANUMERIC CODE OCTAL	BINARY 128	64	32	16	8	4	2	1	ASCII SYMBOL
—	—	055	0	0	1	0	1	1	0	1	255	1	0	1	0	1	1	0	1	—
·	·	056	0	0	1	0	1	1	1	0	256	1	0	1	0	1	1	1	0	·
÷	/	057	0	0	1	0	1	1	1	1	257	1	0	1	0	1	1	1	1	/
0	0	060	0	0	1	1	0	0	0	0	260	1	0	1	1	0	0	0	0	0
1	1	061	0	0	1	1	0	0	0	1	261	1	0	1	1	0	0	0	1	1
2	2	062	0	0	1	1	0	0	1	0	262	1	0	1	1	0	0	1	0	2
3	3	063	0	0	1	1	0	0	1	1	263	1	0	1	1	0	0	1	1	3
4	4	064	0	0	1	1	0	1	0	0	264	1	0	1	1	0	1	0	0	4
5	5	065	0	0	1	1	0	1	0	1	265	1	0	1	1	0	1	0	1	5
6	6	066	0	0	1	1	0	1	1	0	266	1	0	1	1	0	1	1	0	6
7	7	067	0	0	1	1	0	1	1	1	267	1	0	1	1	0	1	1	1	7
8	8	070	0	0	1	1	1	0	0	0	270	1	0	1	1	1	0	0	0	8
9	9	071	0	0	1	1	1	0	0	1	271	1	0	1	1	1	0	0	1	9
ADDRS	:	072	0	0	1	1	1	0	1	0	272	1	0	1	1	1	0	1	0	:
START	;	073	0	0	1	1	1	0	1	1	273	1	0	1	1	1	0	1	1	;
<Ø	<	074	0	0	1	1	1	1	0	0	274	1	0	1	1	1	1	0	0	<
=	=	075	0	0	1	1	1	1	0	1	275	1	0	1	1	1	1	0	1	=
≥Ø	>	076	0	0	1	1	1	1	1	0	276	1	0	1	1	1	1	1	0	>
=Ø	?	077	0	0	1	1	1	1	1	1	277	1	0	1	1	1	1	1	1	?
K	@	100	0	1	0	0	0	0	0	0	300	1	1	0	0	0	0	0	0	@
DEG RAD	A	101	0	1	0	0	0	0	0	1	301	1	1	0	0	0	0	0	1	A
arc	B	102	0	1	0	0	0	0	1	0	302	1	1	0	0	0	0	1	0	B
hyper	C	103	0	1	0	0	0	0	1	1	303	1	1	0	0	0	0	1	1	C
*tan*X	D	104	0	1	0	0	0	1	0	0	304	1	1	0	0	0	1	0	0	D
*cos*X	E	105	0	1	0	0	0	1	0	1	305	1	1	0	0	0	1	0	1	E
*sin*X	F	106	0	1	0	0	0	1	1	0	306	1	1	0	0	0	1	1	0	F
X!	G	107	0	1	0	0	0	1	1	1	307	1	1	0	0	0	1	1	1	G
Π_4	H	110	0	1	0	0	1	0	0	0	310	1	1	0	0	1	0	0	0	H
Δ_3	I	111	0	1	0	0	1	0	0	1	311	1	1	0	0	1	0	0	1	I
${}_3\Sigma_2$	J	112	0	1	0	0	1	0	1	0	312	1	1	0	0	1	0	1	0	J
Σ_1	K	113	0	1	0	0	1	0	1	1	313	1	1	0	0	1	0	1	1	K
Σ_0	L	114	0	1	0	0	1	1	0	0	314	1	1	0	0	1	1	0	0	L

TABLE 5-1 (cont)

TEKTRONIX 31 CALCULATOR KEYCODES

KEY	HOLD FOR ALPHA	OPERATIONAL CODE									ALPHANUMERIC CODE									ASCII SYMBOL
		OCTAL	BINARY								OCTAL	BINARY								
			128	64	32	16	8	4	2	1		128	64	32	16	8	4	2	1	
*ln*X	M	115	0	1	0	0	1	1	0	1	315	1	1	0	0	1	1	0	1	M
*log*X	N	116	0	1	0	0	1	1	1	0	316	1	1	0	0	1	1	1	0	N
*int*X	O	117	0	1	0	0	1	1	1	1	317	1	1	0	0	1	1	1	1	O
$\sqrt{\Sigma x^2}$	P	120	0	1	0	1	0	0	0	0	320	1	1	0	1	0	0	0	0	P
$\|x\|^a$	Q	121	0	1	0	1	0	0	0	1	321	1	1	0	1	0	0	0	1	Q
REMOTE ■ ■	R	122	0	1	0	1	0	0	1	0	322	1	1	0	1	0	0	1	0	R
e^x	S	123	0	1	0	1	0	0	1	1	323	1	1	0	1	0	0	1	1	S
X^2	T	124	0	1	0	1	0	1	0	0	324	1	1	0	1	0	1	0	0	T
$\sqrt{X}$	U	125	0	1	0	1	0	1	0	1	325	1	1	0	1	0	1	0	1	U
$\frac{1}{x}$	V	126	0	1	0	1	0	1	1	0	326	1	1	0	1	0	1	1	0	V
Π	W	127	0	1	0	1	0	1	1	1	327	1	1	0	1	0	1	1	1	W
PAPER FEED	X	130	0	1	0	1	1	0	0	0	330	1	1	0	1	1	0	0	0	X
$x10^{00}$	Y	131	0	1	0	1	1	0	0	1	331	1	1	0	1	1	0	0	1	Y
CONT	Z	132	0	1	0	1	1	0	1	0	332	1	1	0	1	1	0	1	0	Z
RETURN ADDRESS	[	133	0	1	0	1	1	0	1	1	333	1	1	0	1	1	0	1	1	[
+ − (with arrows)	/	134	0	1	0	1	1	1	0	0	334	1	1	0	1	1	1	0	0	/
GO TO DISPLAY	]	135	0	1	0	1	1	1	0	1	335	1	1	0	1	1	1	0	1	]
FLAG	↑	136	0	1	0	1	1	1	1	0	336	1	1	0	1	1	1	1	0	↑
END	—	137	0	1	0	1	1	1	1	1	337	1	1	0	1	1	1	1	1	—

TABLE 5-2

TEKTRONIX 21 CALCULATOR KEYCODES

KEY	OCTAL	BINARY								KEY	OCTAL	BINARY							
		128	64	32	16	8	4	2	1			128	64	32	16	8	4	2	1
IF ≤Ø GO TO	001	0	0	0	0	0	0	0	1	*f*2	072	0	0	1	1	1	0	1	0
*f*7	002	0	0	0	0	0	0	1	0	*f*1	073	0	0	1	1	1	0	1	1
*f*6	003	0	0	0	0	0	0	1	1	=	075	0	0	1	1	1	1	0	1
END	004	0	0	0	0	0	1	0	0	K ■	100	0	1	0	0	0	0	0	0
*f*3	007	0	0	0	0	0	1	1	1	DEG RAD	101	0	1	0	0	0	0	0	1
*f*4	010	0	0	0	0	1	0	0	0	*arc*	102	0	1	0	0	0	0	1	0
*f*5	011	0	0	0	0	1	0	0	1	*hyper*	103	0	1	0	0	0	0	1	1
*STEP	023	0	0	0	1	0	0	1	1	*tan*X	104	0	1	0	0	0	1	0	0
*LEARN *f*(X)	024	0	0	0	1	0	1	0	0	*cos*X	105	0	1	0	0	0	1	0	1
*LIST	026	0	0	0	1	0	1	1	0	*sin*X	106	0	1	0	0	0	1	1	0
CLEAR DSPLY	040	0	0	1	0	0	0	0	0	X!	107	0	1	0	0	0	1	1	1
STOP	043	0	0	1	0	0	0	1	1	Π_4	110	0	1	0	0	1	0	0	0
PRINT DSPLY	044	0	0	1	0	0	1	0	0	Δ_3	111	0	1	0	0	1	0	0	1
CLEAR	045	0	0	1	0	0	1	0	1	${}_3\Sigma_2$	112	0	1	0	0	1	0	1	0
(	050	0	0	1	0	1	0	0	0	Σ_1	113	0	1	0	0	1	0	1	1
)	051	0	0	1	0	1	0	0	1	Σ_0	114	0	1	0	0	1	1	0	0
X	052	0	0	1	0	1	0	1	0	*ln*X	115	0	1	0	0	1	1	0	1
+	053	0	0	1	0	1	0	1	1	*log*X	116	0	1	0	0	1	1	1	0
*f*0	054	0	0	1	0	1	1	0	0	*int*X	117	0	1	0	0	1	1	1	1
−	055	0	0	1	0	1	1	0	1	$\sqrt{\Sigma x^2}$	120	0	1	0	1	0	0	0	0
·	056	0	0	1	0	1	1	1	0	$\lvert x \rvert^a$	121	0	1	0	1	0	0	0	1
÷	057	0	0	1	0	1	1	1	1	REMOTE ■ ■	122	0	1	0	1	0	0	1	0
0	060	0	0	1	1	0	0	0	0	e^x	123	0	1	0	1	0	0	1	1
1	061	0	0	1	1	0	0	0	1	X^2	124	0	1	0	1	0	1	0	0
2	062	0	0	1	1	0	0	1	0	$\sqrt{X}$	125	0	1	0	1	0	1	0	1
3	063	0	0	1	1	0	0	1	1	$\frac{1}{X}$	126	0	1	0	1	0	1	1	0
4	064	0	0	1	1	0	1	0	0	Π	127	0	1	0	1	0	1	1	1
5	065	0	0	1	1	0	1	0	1	PAPER FEED	130	0	1	0	1	1	0	0	0
6	066	0	0	1	1	0	1	1	0	$x10^{00}$	131	0	1	0	1	1	0	0	1
7	067	0	0	1	1	0	1	1	1	CONT	132	0	1	0	1	1	0	1	0
8	070	0	0	1	1	1	0	0	0	+ −	134	0	1	0	1	1	1	0	0
9	071	0	0	1	1	1	0	0	1	GO TO	137	0	1	0	1	1	1	1	1

***Not Programmable.**

Ⓐ

BUS MANAGEMENT

General

The External Bus is just as extension of the calculator's internal data lines and Bus Management signals are necessary to keep external devices from interfering with calculator operations. Although these signals can be asserted by any device in the system, they are normally used by the calculator when learning or executing program instructions.

INHIBIT INPUT ($\overline{\text{IHIN}}$)

Refer to Fig. 6-1 TEKTRONIX 31 Calculator System Block Diagram. When the calculator is idle, the programmer board asserts $\overline{\text{IHIN}}$. This prevents the calculator board and other functions within the calculator from "seeing" a keycode when it is placed on the External Bus by the keyboard or an external device. When a keycode is placed on the External Bus and $\overline{\text{STBE}}$ is asserted, the programmer board examines the code. When it is ready, the programmer board releases $\overline{\text{IHIN}}$ and re-issues the code to the proper location via the External Bus. (An "echo" suppressor may be needed to prevent the interface from seeing the re-issued code.)

If the calculator is put in the LEARN mode, the programmer board asserts $\overline{\text{IHIN}}$ continually to prevent the keycodes from being executed. It then stores every keycode placed on the External Bus.

INHIBIT OUTPUT ($\overline{\text{INHOUT}}$)

Refer to Fig. 6-1 TEKTRONIX 31 Calculator System Block Diagram. When the calculator is executing a program, the programmer board issues operational codes to the calculator board over the External Bus. At this time, the calculator asserts $\overline{\text{INHOUT}}$. $\overline{\text{INHOUT}}$ is non-selective and prevents every external device from placing information on the External Bus.

Using the $\overline{\text{CALL}}$ Line to Execute a Subroutine

Refer to Fig. 6-1 TEKTRONIX 31 Calculator System Block Diagram. $\overline{\text{CALL}}$ performs a call to a subroutine stored in the calculator's memory. When $\overline{\text{CALL}}$ is asserted by an external device along with a keycode from octal 101 to 130, the calculator does not execute the normal keycode instruction. Instead, the programmer board starts at memory location 0000 and searches through its memory until it finds a subroutine labeled by that keycode name. It then executes the subroutine. If the label is not found, an E-5 error message appears in the display.

Asserting the $\overline{\text{CALL}}$ line is the same as using the User Definable Overlay on the calculator keyboard. Placing the Overlay into the clip on the keyboard depresses a switch which activates the $\overline{\text{CALL}}$ line.

EXTERNAL PRINTER PRESENT ($\overline{\text{EXTPTR}}$)

Any external device with printing capability informs the calculator of its presence by activating the $\overline{\text{EXTPTR}}$ line. If $\overline{\text{EXTPTR}}$ is asserted when the PRINT DSPLY key is depressed, the calculator not only prints the display on its printer, but it sends the display contents to the external printer via the External Bus as a string of 16 alphanumeric characters. ($\overline{\text{EXT128}}$ is asserted by the calculator). The alphanumeric characters are placed on the External Bus one at a time, starting with the character on the left hand side of the display and ending with the character on the right hand side of the display.

If the PAPER FEED key is depressed with $\overline{\text{EXTPTR}}$ asserted, the calculator places ASCII code 130 on the External Bus to advance the paper on the calculator printer. It then puts the ASCII code for LF (LINE FEED, octal 212), and ASCII code for CR (CARRIAGE RETURN, octal 215), on the External Bus to advance the paper and return the carriage on the external printer.

DIRECT MEMORY ACCESS (DMA)

General

The TEKTRONIX 31 Calculator has a direct memory access capability which allows R-register data and program steps to be transferred in byte parallel format over the External Bus. DMA is bi-directional. An external device can write program steps and R-register data into the calculator's memory and extract program steps and R-register data from the calculator's memory. Program steps and R-register data cannot be transferred at the same time.

Before DMA begins, the calculator's sequencer must be set to the desired memory location. The calculator then issues a remote command to the external device to start the transfer. The length of the transfer is controlled by the external device which terminates the operation when the desired memory locations are acquired.

The signals associated with DMA are as follows:

$\overline{\text{DMAWR}}$ (DMA WRITE)	Tells the calculator that the external device is going to write information into the calculator's memory.
$\overline{\text{DMARD}}$ (DMA READ)	Tells the calculator that the external device is going to read information stored in the calculator's memory.
$\overline{\text{DMARY}}$ (DMA READY)	Tells the external device that the calculator is ready to begin the DMA transfer.
$\overline{\text{STBE}}$ (STROBE)	Issued by the external device to transfer a byte of information over the External Bus.
$\overline{\text{BUSY}}$	Asserted by the external device to prevent the calculator from operating on the information being transferred.
$\overline{\text{IHIN}}$ (INHIBIT INPUT)	Activated by the calculator during DMA to prevent other devices from receiving the information being placed on the External Bus.

The signals $\overline{\text{DMAWR}}$, $\overline{\text{DMARD}}$, $\overline{\text{STBE}}$, and $\overline{\text{BUSY}}$ are asserted by the external device. The signals $\overline{\text{DMARY}}$ and $\overline{\text{IHIN}}$ are asserted by the calculator.

DMA Data Formatting

Program Steps Operational keycodes and alphanumeric codes which are written into program step locations are formatted as shown in Fig. 5-2. Numeric digits (0-9) in the program are represented by octal codes 060-071. Alphanumeric digits (0-9) are represented by octal codes 260-271 as shown in Table 5-1. One strobe transfers one octal code in or out of a program step location.

R-register Data. Data stored in an R-register enters the register serially from the Data Bus as a string of BCD digits and binary numbers (see Fig. 4-2). One R-register holds one numeric value. In order to transfer the contents of an R-register over the External Bus, the lines are divided into two groups. $\overline{EXT1}$-$\overline{EXT8}$ carry a BCD digit and $\overline{EXT16}$-$\overline{EXT128}$ carry a BCD digit. Two digits are transferred as one byte. It takes eight bytes to transfer the entire R-register.

Refer to Fig. 5-3. Fig. 5-3 shows how the BCD digits are placed on the External Bus. The digits "5" and "2" are used as an example. The "2" is placed on lines $\overline{EXT1}$-$\overline{EXT8}$ with $\overline{EXT8}$ representing the most significant bit. The "5" is placed on lines $\overline{EXT16}$-$\overline{EXT128}$ with EXT128 representing the most significant bit. It is important to remember that a binary 1 is denoted by an external line which is pulled low (0 VDC) and a binary 0 is denoted by an external line which is left high (+3.6 VDC).

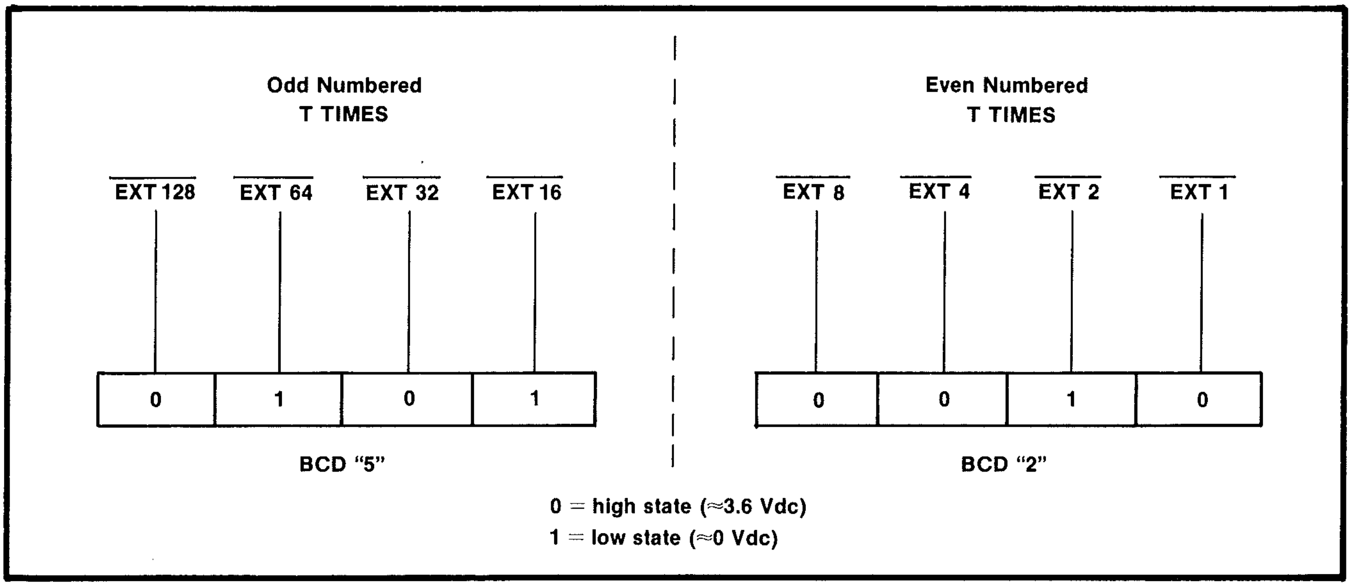


Fig. 5-3. BCD digits placed on the External bus.

Fig. 5-4 is a timing diagram which shows how the number 18749 is transferred to R-register 035 over the External Bus. Each BCD digit is labeled according to the time frame it occupies on a Data Bus transfer. All even T Time digits are transferred on $\overline{\text{EXT1}}$-$\overline{\text{EXT8}}$. All odd T Time digits are transferred on $\overline{\text{EXT16}}$-$\overline{\text{EXT128}}$. T0 and T1 are transferred as one byte followed by T2 and T3 and so on until all eight bytes are transferred. (One strobe transfers one byte.)

A Word of Caution. It is important to make sure that information is transferred in the right format and placed in the proper location. If BCD digits are written into program step locations, the calculator will see the data as an octal code and perform the indicated operation if put under program control. The same holds true for octal codes stored in R-registers. The octal codes are seen as BCD digits, which means invalid data.

Writing Information Into the Calculator's Memory ($\overline{\text{DMAWR}}$)

As stated previously, DMA transfers are initiated by the calculator through a remote command. A remote address (assigned by the designer) tells the external device to set up for DMA. The calculator's automatic sequencer must be placed in the desired memory location before the remote command is given.

To write program steps into memory, the following keys are struck, with X indicating desired numbers:

ADDRESS GOTO XXXX REMOTE XX

To write data into R-registers, the following keys are struck:

ADDRESS R▪▪▪ XXX REMOTE XX

The ADDRESS GOTO XXXX sequence tells the calculator that program steps are going to be transferred and gives the starting location. The ADDRESS R▪▪▪ XXX sequence tells the calculator that R-register data is going to be transferred and gives the starting location. The remote command in each case triggers circuits on the interface which causes the interface to assert $\overline{\text{DMAWR}}$ (DMA WRITE) and $\overline{\text{BUSY}}$. The calculator responds with $\overline{\text{DMARY}}$ (DMA READY) and $\overline{\text{IHIN}}$ (INHIBIT INPUT).

After the calculator asserts DMA READY, the external device places the first byte of information on the External Bus and issues strobe ($\overline{\text{STBE}}$). The information must be on the lines for at least 2 μs before the leading edge of $\overline{\text{STBE}}$ and must remain on the lines at least 2 μs after the trailing edge of $\overline{\text{STBE}}$. Each strobe transfers one byte of information into the memory. For example:

1 strobe = 1 program step

5 strobes = 5 program steps

8 strobes = 1 R-register

40 strobes = 5 R-registers

The strobes can be issued at any asynchronous rate equal to or less than 125k strobes/sec.

When the external device sends the last byte of information, it signals the calculator by releasing $\overline{\text{DMAWR}}$ and $\overline{\text{BUSY}}$. $\overline{\text{DMAWR}}$ and $\overline{\text{BUSY}}$ cannot be released until 8 μs after the last strobe, or the last byte of information will be lost. After they are released, the calculator responds by releasing $\overline{\text{DMARY}}$ and $\overline{\text{IHIN}}$, then continues program execution if operating under program control.

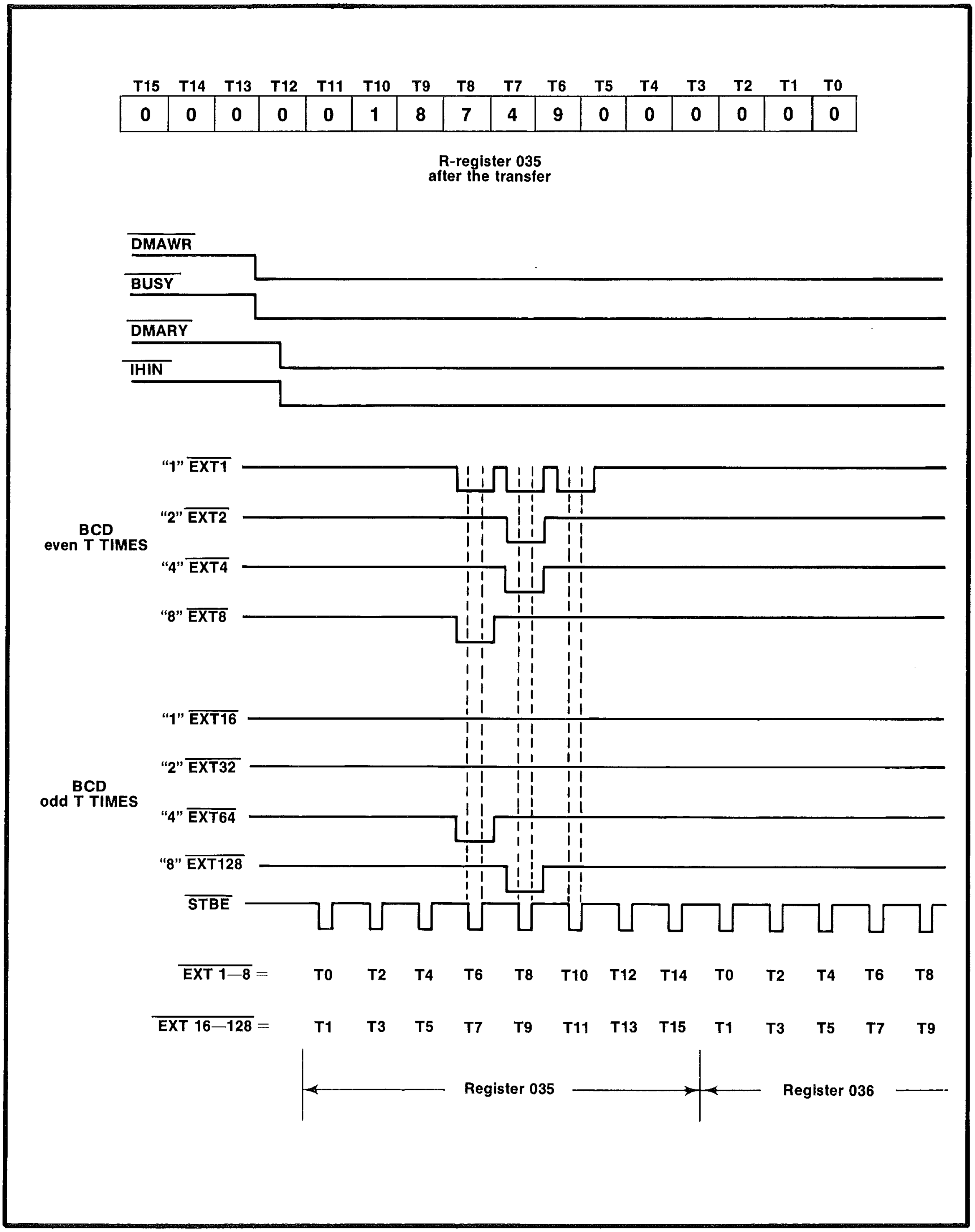


Fig. 5-4. Transferring 18749 into R-register 035.

Reading the Calculator's Memory ($\overline{\text{DMARD}}$)

The procedure for transferring program steps and R-register data from the calculator's memory is almost identical to the procedure for transferring information to the calculator's memory. The calculator's automatic sequencer is placed in the desired memory location and a remote command is given to start the transfer.

To read program steps, the following keys are struck, with X indicating desired numbers:

ADDRESS GOTO XXXX REMOTE XX

To read R-register data, the following keys are struck, with X again indicating desired numbers:

ADDRESS R▪▪▪ XXX REMOTE XX

The REMOTE XX in each case causes the interface to assert $\overline{\text{DMARD}}$ (DMA READ) and $\overline{\text{BUSY}}$. The calculator responds with $\overline{\text{DMARY}}$ (DMA READY), $\overline{\text{IHIN}}$ (INHIBIT INPUT) and places the first byte of information on the External Bus. The external device receives the information by issuing a strobe ($\overline{\text{STBE}}$). $\overline{\text{STBE}}$ tells the calculator that the information has been taken off the lines and tells it to put the next byte of information on the lines.

If R-register data is read, an initialization strobe is required to start the transfer. If program steps are read, an initialization strobe is not required. For example:

To read program steps:

1 program step = 1 strobe

5 program steps = 5 strobes

To read R-register data:

1 R-register = 1 initialization strobe
+ 8 byte strobes

5 R-registers = 1 initialization strobe
+ 40 byte strobes

Fig. 5-5 shows how the value π (3.141592653) is read from R-register 047. Notice that an initialization strobe is required before the first byte of data is taken off the lines. After the first register is transferred, the next eight strobes will transfer the next register, and so on.

The strobe can be issued synchronous to the calculator's clocking system or issued at any asynchronous rate equal to or less than 125k strobes/sec.

After the external device has read the desired amount of information, it signals the calculator by releasing $\overline{\text{DMARD}}$ and $\overline{\text{BUSY}}$. The calculator responds by releasing $\overline{\text{DMARY}}$ and $\overline{\text{IHIN}}$, then continues program execution if operating under program control.

Controlling the Length of the Transfer

The length of the transfer can be controlled by a down counter located on the interface. The counter is set by using the Data Bus and a remote command (as discussed in section 4). The desired number of steps are entered into the calculator's display as a data value and transferred over the Data Bus to set the counter. During the DMA operation, the counter is counted down as each byte of information is transferred. When the counter reaches 0, it causes the external device to release $\overline{\text{DMAWR}}$, $\overline{\text{BUSY}}$ and to inactivate the $\overline{\text{STBE}}$ signal line.

Software Features

If DMA is executed under the program control and program steps are written into the calculator's memory, the calculator makes a check at the end of the transfer. If the instructions which set up the transfer are contained in the same memory page, the calculator performs an unconditional branch to step 0000 and re-starts the program. (A page contains 1000 steps; 0-999 for example.) If the instructions which set up the transfer are not contained in the same memory page, the calculator branches back to the program step following the last executed instruction and resumes program execution. When writing data into R-registers, program execution resumes at the location following the last executed instruction regardless of the memory file used during the transfer.

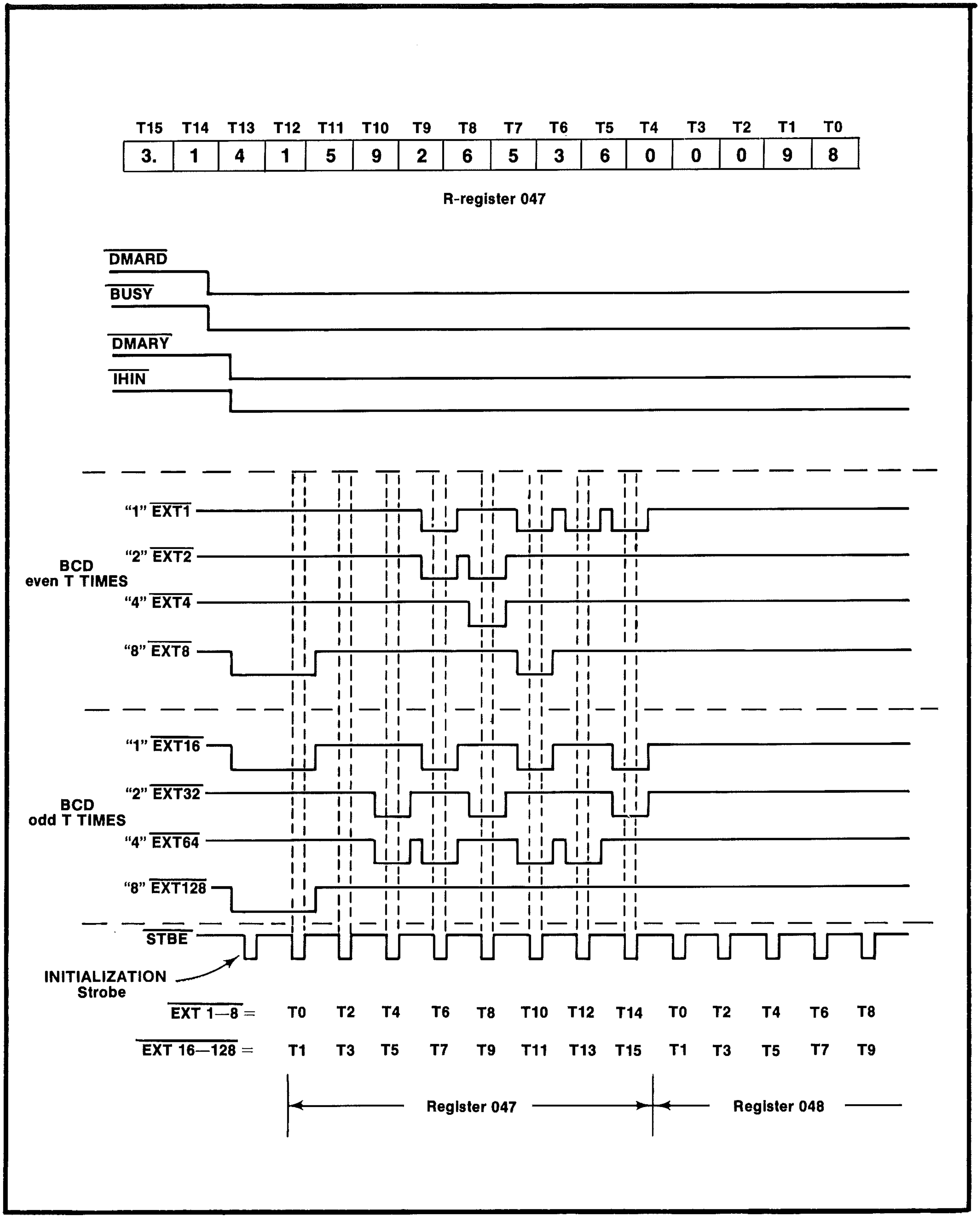


Fig. 5-5. Transferring π (3.1415926536) from R-register 047.

TYPICAL INTERFACE CIRCUITS FOR DMA TRANSFER

DMA WRITE

General. Refer to pullout sheet Fig. 6-5 DMA WRITE in section 6 and timing diagram, Fig. 5-6. The DMA WRITE schematic and timing diagram show how program steps are written into the calculator's memory from an external device. The same circuits can be used to write R-register data. The transfer of 5 keycodes – 9 (octal 071), 2 (octal 062), $\sqrt{X}$ (octal 125), ln X (octal 115), and Σ (octal 114) are illustrated in the timing diagram.

Keeping Track of the Program Steps Transferred. It was previously mentioned that an external device is responsible for keeping track of the program steps transferred during DMA. Before the transfer begins, the calculator must be set to the desired memory location and a down counter set to the number of steps to be transferred. The down counter in Fig. 6-3, Calculator Send Data is an example of a counter that can be used for this purpose.

Refer to pullout sheet Fig. 6-3, Calculator Send Data. The number of steps to be transferred is entered into the calculator display and transferred to the interface to set the counter. (This can be done under program control.) Another remote command is then given to start the DMA transfer. The output of the counter is used to address memory locations in the external device and the counter is counted down as each byte of information is transferred. In this configuration, the clock pulse from U14B on the DMA WRITE diagram would be applied to the $\overline{\text{COUNT DOWN}}$ input (top of U17) on the Calculator Send Data diagram.

Handshake Circuitry. The handshake circuitry is similar to the handshake circuitry previously discussed. The remote address which starts the transfer is received in BCD format on lines UD1-UD8 and TD1-TD8. The address is decoded by BCD-to-decimal decoders U8 and U9. The remote address "18" has been selected in this example.

The UNITS digit and TENS digit are ANDED together at U10A and applied as a high to flip-flop U11 (point①). A short time later, $\overline{\text{AVS}}$ (ADDRESS VALID STROBE) from the calculator (point②) clocks the Q output high. This drives the outputs of U13B and U13C low, which asserts $\overline{\text{DMAWR}}$ (DMA WRITE point③) and $\overline{\text{BUSY}}$. Once $\overline{\text{DMAWR}}$ is asserted, the calculator asserts $\overline{\text{DMARY}}$ (DMA READY) which allows the transfer to begin. Shift register U12 is used to delay the release of $\overline{\text{DMAWR}}$ and $\overline{\text{BUSY}}$ and will be discussed later.

Clock Generator. The clock generator (U15 and U16) generates an enable pulse for the bus drivers (U17A-D and U18A-D) and a strobe ($\overline{\text{STBE}}$) for the External Bus. U15 and U16 are two decade counters set up to count in bi-quinary. Before $\overline{\text{DMARY}}$ (DMA READY) is asserted by the calculator, the normally high output from U7C is applied to U15 inputs R9 to set the output to a bi-quinary "9", and to U16 inputs R0 to set the output to a bi-quinary "0". (See the "Bi-quinary Count" charter – upper right corner.)

Bus Drivers. The bus drivers put keycodes on the External Bus. The keycodes from the external device are placed on the eight lines located in the top left corner of the diagram (labeled Octal Code from External Device). The bus drivers are enabled for 6 µs (3 T Times) by the pulse from U15 (point⑦).

The Transfer. The transfer is allowed to begin when the calculator asserts $\overline{\text{DMARY}}$, but won't start until the external device asserts DATA READY (point⑤). The output of U1B goes high; the output of U7C goes low (point④). This enables U15 and U16 to respond to the $\overline{\text{ECP2}}$ clock pulses when they are applied (point⑥). It also enables U10B to pass the enable pulse from U15 to the bus drivers.

After the external device puts the keycode on the lines, it asserts DATA READY. This enables U14A to pass $\overline{\text{ECP2}}$ clock pulses to the clock generator. Each $\overline{\text{ECP2}}$ pulse causes both counters to increment one count. Refer to the Binary Count chart. The C output of U15 is initially low and remains low when clocked with the first $\overline{\text{ECP2}}$ pulse. The B output of U16 is initially low and goes high when clocked with the first $\overline{\text{ECP2}}$ pulse. This drives U13A output low, which asserts $\overline{\text{STBE}}$. $\overline{\text{STBE}}$ tells the calculator to write the keycode into memory.

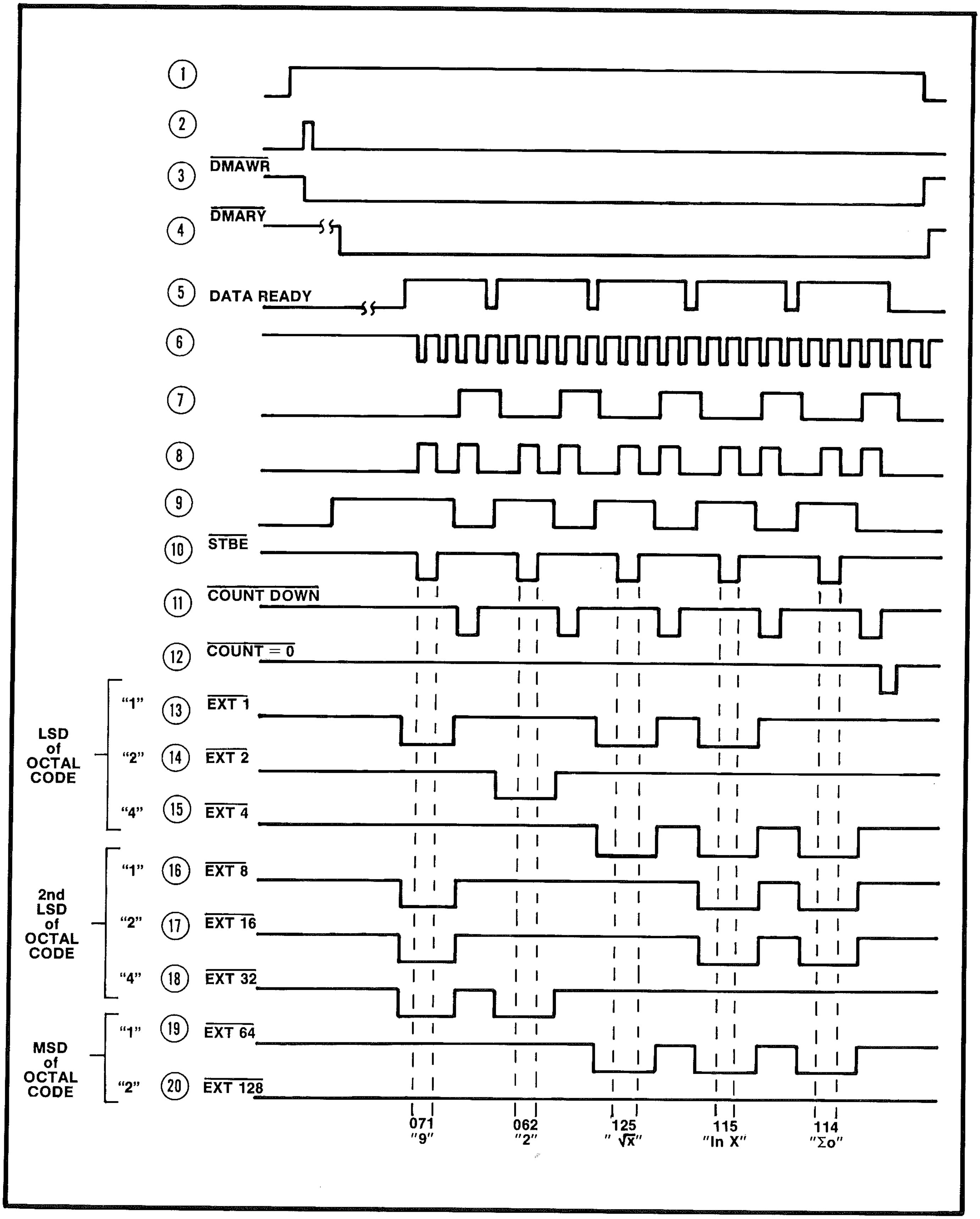


Fig. 5-6. DMA WRITE (Program Steps).

The second $\overline{ECP2}$ pulse increments both counters. The C output of U15 remains low and the B output of U16 goes low. This allows the keycode to remain on the lines for one T Time (2 μs) after $\overline{STBE}$. The third $\overline{ECP2}$ pulse clocks both counter outputs to a high state. The output of U10B (point⑨) goes low, which disables the bus drivers. This removes the keycode from the bus. The output of U14B also goes low, which clock the Down Counter down one count.

The following $\overline{ECP2}$ pulse releases $\overline{COUNT\ DOWN}$. DATA READY goes low until the next keycode is put on the lines. The count continues when DATA READY goes high again, which allows $\overline{ECP2}$ to drive U15 and U16.

In this diagram, the DMA transfer is sychronized to the calculator's clocking system. However, the clock generator can be driven from an asynchronous source at a rate equal to or slower than the rate shown in the timing diagram.

After the last keycode is transferred, the output of U14B clocks the down counter to zero and the $\overline{COUNT = 0}$ line goes low (point②). This low is applied to serial inputs S_A and S_B of U12.

Delaying the Release of $\overline{DMAWR}$. Shift register U12 delays the release of $\overline{DMAWR}$ and $\overline{BUSY}$. As stated previously, $\overline{DMAWR}$ must not be released before 8 μs after the last strobe, or the last byte of information will be lost. On power up, $\overline{DCLOW}$ from the calculator clears U12; the low output from U12-B clears U11. This insures that $\overline{DMAWR}$ and $\overline{BUSY}$ are not asserted prematurely. When the down counter is set prior to a DMA write operation, the $\overline{COUNT = 0}$ line goes high. $\overline{ECP2}$ pulses from U7B then clock all the outputs of U12 (A thru H) to a high state. This inactivates the CLEAR input to U11. The transfer begins when the remote address "18" and $\overline{AVS}$ set U11 output high. After the last byte of information is transferred, the $\overline{COUNT = 0}$ signal goes low. The trailing edge of the next $\overline{ECP2}$ pulse clocks the A output of U12 low. The following $\overline{ECP2}$ pulse clocks the B output low, which resets U11 and releases $\overline{DMAWR}$ and $\overline{BUSY}$. This small delay gives the calculator enough time to capture the last byte of information.

DMA READ

General. Refer to pullout sheet Fig. 6-6, DMA READ in section 6 and timing diagram Fig. 6-7 (also on a pullout). The DMA READ schematic and timing diagram show how data stored in an R-register is read by an external device. In this example, the value π (3.141592653) is read from R-register 100. Before the remote command is given to start the transfer, the calculator's sequencer is set to R100 and a down counter set to "8" — the number of bytes to be transferred.

Handshake Circuitry. The handshake circuitry is similar to the handshake circuitry previously discussed. The remote address which starts the transfer is received in BCD format on lines UD1-UD8 and TD1-TD8. The address is decoded by BCD to decimal decoders U8 and U9, ANDED together at U10A and applied as a high to flip-flop U11. The ADDRESS VALID STROBE ($\overline{AVS}$) clocks the output of U11 high, which drives the outputs of U13B and U13C low to assert $\overline{DMARD}$ (DMA READ, point①) and $\overline{BUSY}$. A short time later, the calculator asserts $\overline{DMARY}$ (DMA READY) and the transfer begins.

Clock Generator. The clock generator (U15 and U16) generates an enable pulse for the bus receivers and a strobe ($\overline{STBE}$) for the External Bus. U15 and U16 are two decade counters which count in bi-quinary. Before DMA READY is asserted by the calculator, the normally high output of U7B is applied to U15 inputs R9 to set the output to a bi-quinary "9", and to U16 inputs R0 to set the output to a bi-quinary "0". (See the "Bi-quinary Count" chart on the right side of the schematic.)

Bus Receivers. The bus receivers "buffer" the External Bus lines so the interface does not draw power from the calculator. The receivers are enabled by a 6μs (3 T Time) pulse generated from U15 (point④).

The Transfer. The transfer begins when the calculator asserts $\overline{DMARY}$. The output of U1B goes high; the output of U7B goes low (point②). This enables U15 and U16 to respond to the $\overline{ECP2}$ clock pulse (point③). It also enables U10B to pass the enable pulse from U15 to the External Bus receivers (point⑥).

The calculator puts the first byte of information on the External Bus when it asserts $\overline{\text{DMARY}}$; however, an initialization strobe must be generated before the byte is transferred. This strobe is generated by the clock generator in the same fashion as a "byte" strobe, however, one-shot multivibrator U12 prevents the strobe from incrementing the down counter (point ⑧).

One-shot U12 is triggered when $\overline{\text{DMARY}}$ is asserted. The Q output goes high, which blocks U10C for 14 μs. This prevents the first strobe from incrementing the down counter. After 14 μs, the output of U12 goes low, which enables U10C to pass the next pulse to down count the counter (as described in the previous text).

The clock generator (U15 and U16) operates the same as the clock generator previously discussed. The output of U15 (point ④) is initially set low and remains low after the first $\overline{\text{ECP2}}$ pulse. This drives the U10B output high, which is inverted by U7C and applied as a low to the bus receivers. This enables the bus receivers to pass the information from the External Bus. The first $\overline{\text{ECP2}}$ pulse also clocks the U16 output high, which drives the output of U13A low to assert $\overline{\text{STBE}}$. Again, the first series of counts produce the initialization strobe; the second series transfers the first byte of information. $\overline{\text{STBE}}$ tells the calculator that the information has been accepted and also tells it to put the next byte on the lines.

In this diagram, the DMA transfer is synchronized to the calculator's clocking system. However, the clock generator can be driven from an asynchronous source at a rate equal to or less than the rate shown in the timing diagram.

After the last byte of information is read, the counter is incremented to zero and the COUNT = 0 lines goes low (point ⑱). U11 is reset, which releases $\overline{\text{DMARD}}$ and $\overline{\text{BUSY}}$. The calculator releases $\overline{\text{DMARY}}$ and goes on to another task if under program control.

Section 6

DIAGRAMS

Symbols and Reference Designators

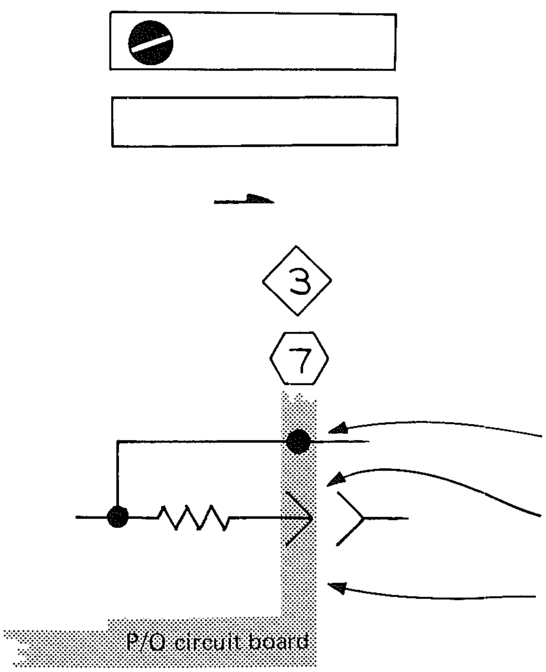
Electrical components shown on the diagrams are in the following units unless noted otherwise:

Capacitors = Values one or greater are in picofarads (pF).
Values less than one are in microfarads (μF).
Resistors = Ohms (Ω)

Symbols used on the diagrams are based on USA Standard Y32.2-1967.

Logic symbology is based on MIL-STD-806B in terms of positive logic. Logic symbols depict the logic function performed and may differ from the manufacturer's data.

The following special symbols are used on the diagrams:



External Screwdriver adjustment.

External control or connector.

Clockwise control rotation in direction of arrow.

Refer to diagram number indicated in diamond.

Refer to waveform number indicated in hexagon.

Connection soldered to circuit board.

Connection made to circuit board with interconnecting pin.

Blue tint encloses components located on circuit board.

The following prefix letters are used as reference designators to identify components or assemblies on the diagrams.

A	Assembly, separable or repairable (circuit board, etc.)
AT	Attenuator, fixed or variable
B	Motor
BT	Battery
C	Capacitor, fixed or variable
CR	Diode, signal or rectifier
DL	Delay line
DS	Indicating device (lamp)
F	Fuse
FL	Filter
H	Heat dissipating device (heat sink, heat radiator, etc.)
HR	Heater
J	Connector, stationary portion
K	Relay
L	Inductor, fixed or variable
LR	Inductor/resistor combination
M	Meter
Q	Transistor or silicon-controlled rectifier
P	Connector, movable portion
R	Resistor, fixed or variable
RT	Thermistor
S	Switch
T	Transformer
TP	Test point
U	Assembly, inseparable or non-repairable (integrated circuit, etc.)
V	Electron tube
VR	Voltage regulator (zener diode, etc.)
Y	Crystal

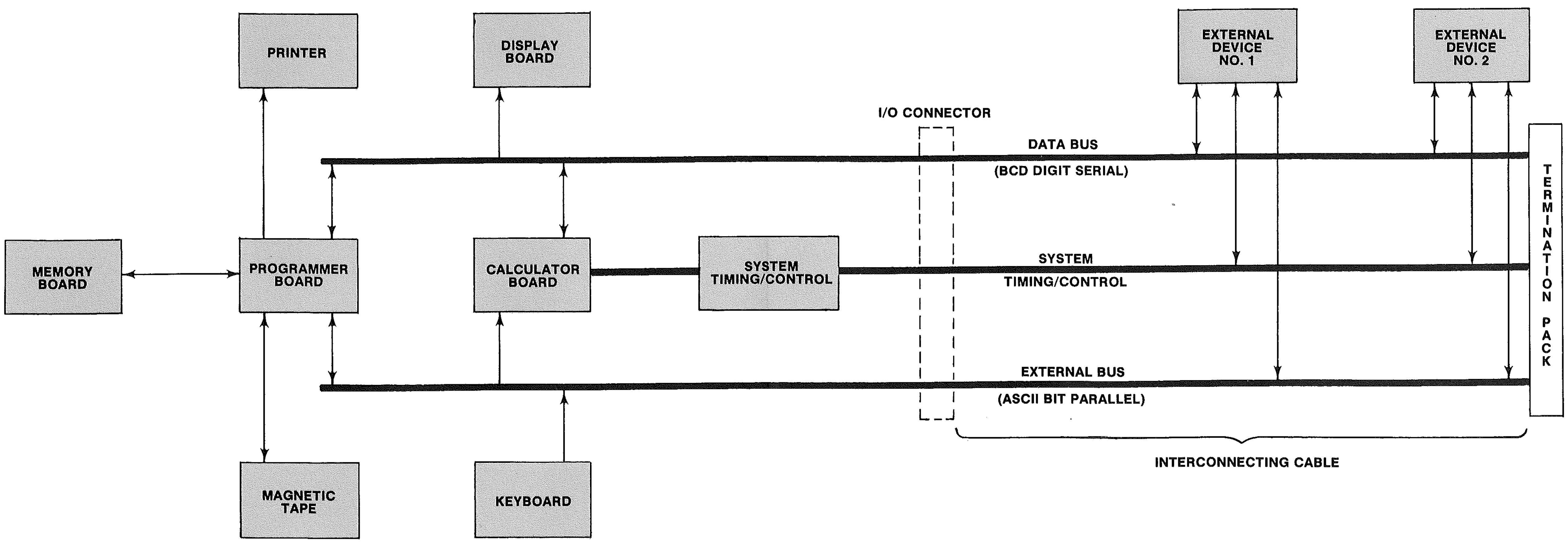


Fig. 6-1. TEKTRONIX 31 CALCULATOR SYSTEM BLOCK DIAGRAM.

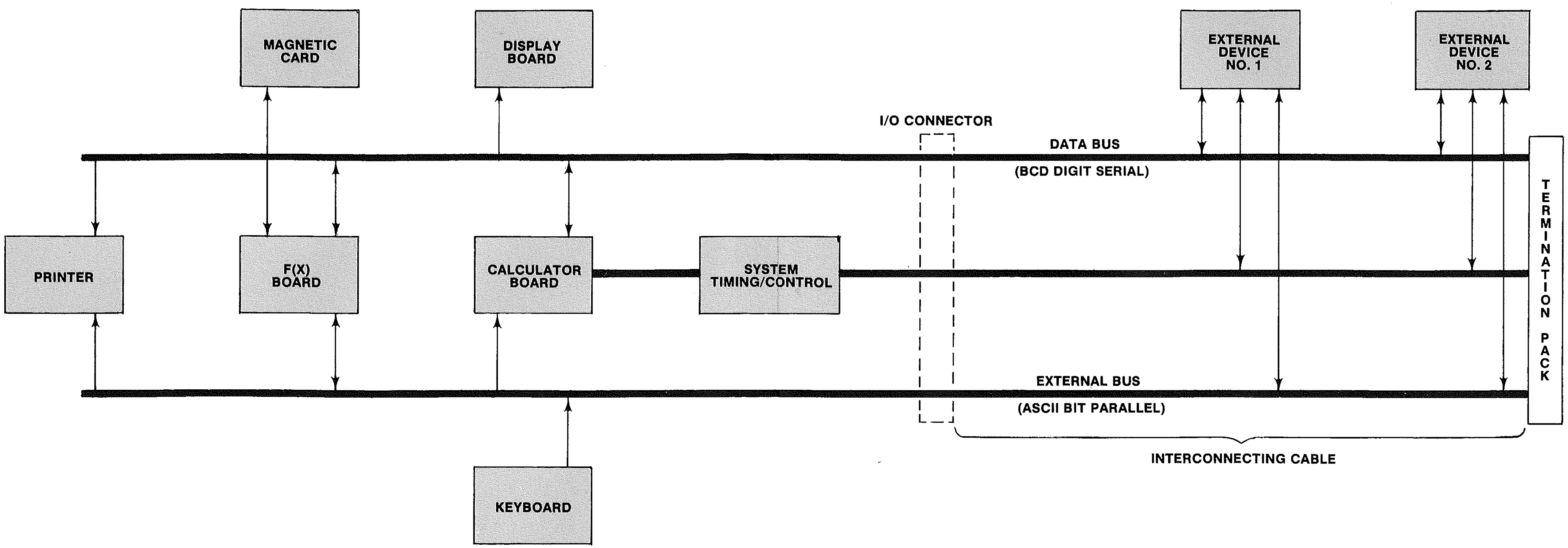


Fig. 6-2. TEKTRONIX 21 CALCULATOR SYSTEM BLOCK DIAGRAM.

FIG. 6-2. 21 SYSTEM BLOCK DIAGRAM

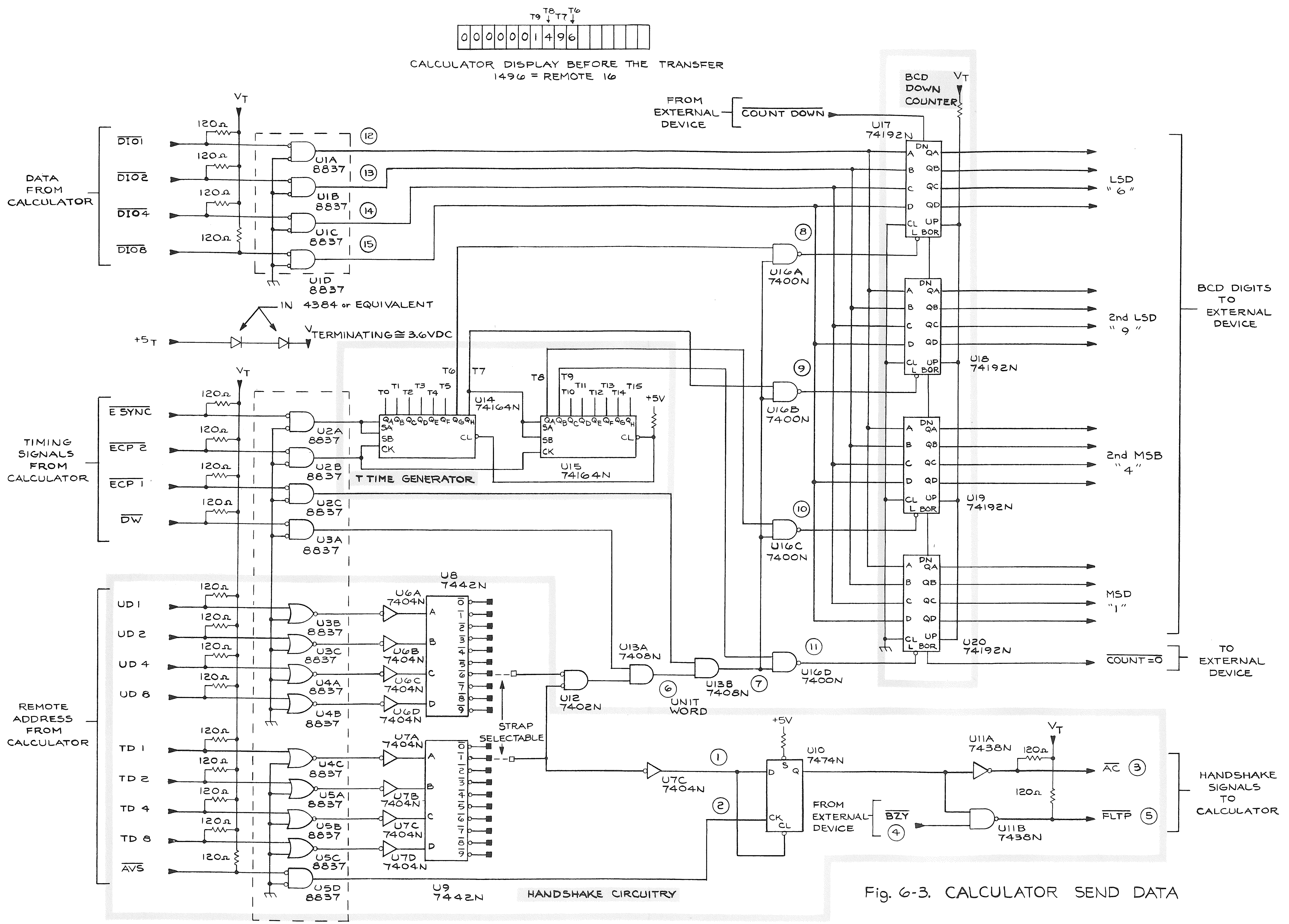


Fig. 6-3. CALCULATOR SEND DATA

FIG. 6-3. CALCULATOR SEND DATA

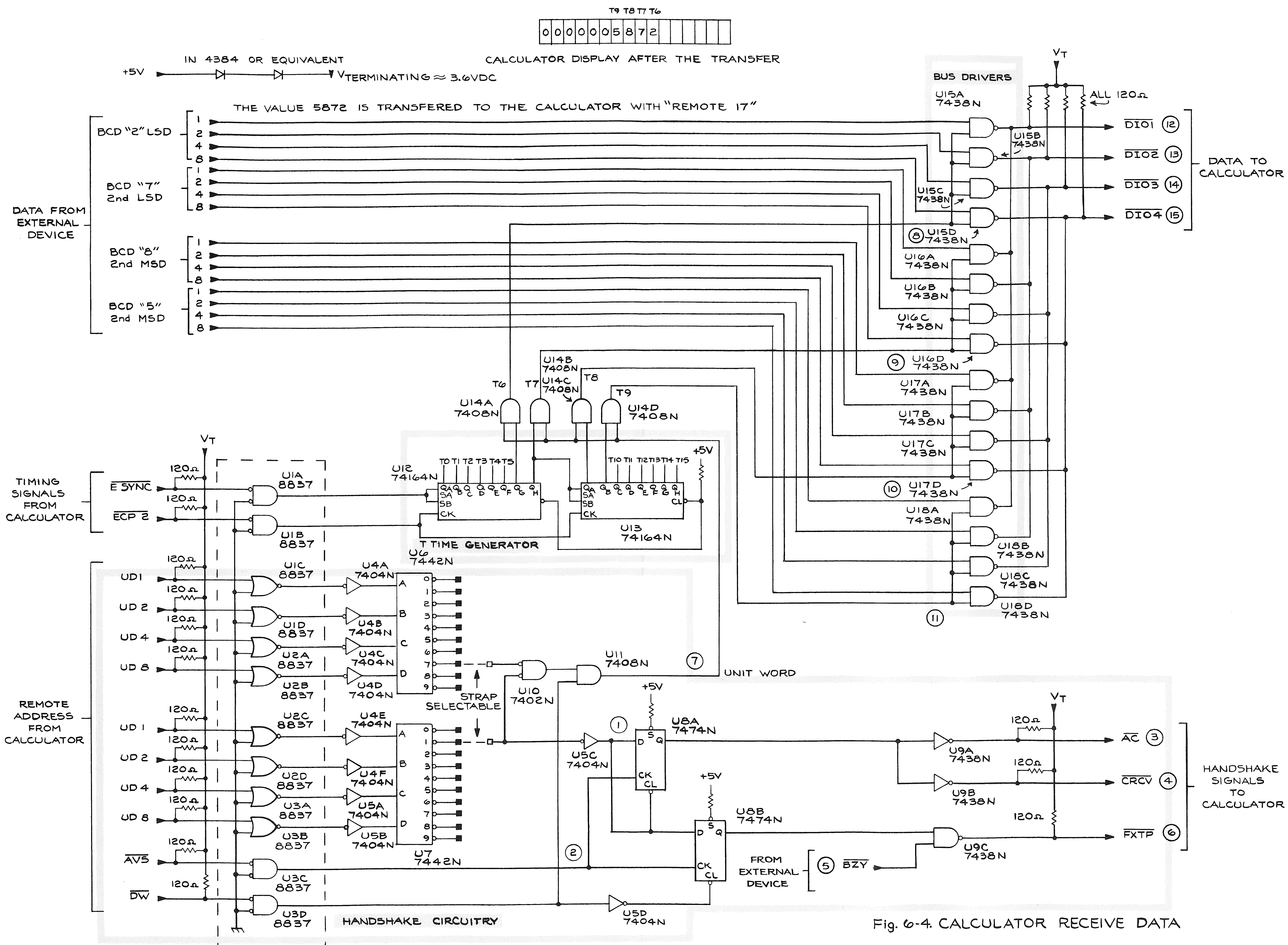


Fig. 6-4. CALCULATOR RECEIVE DATA

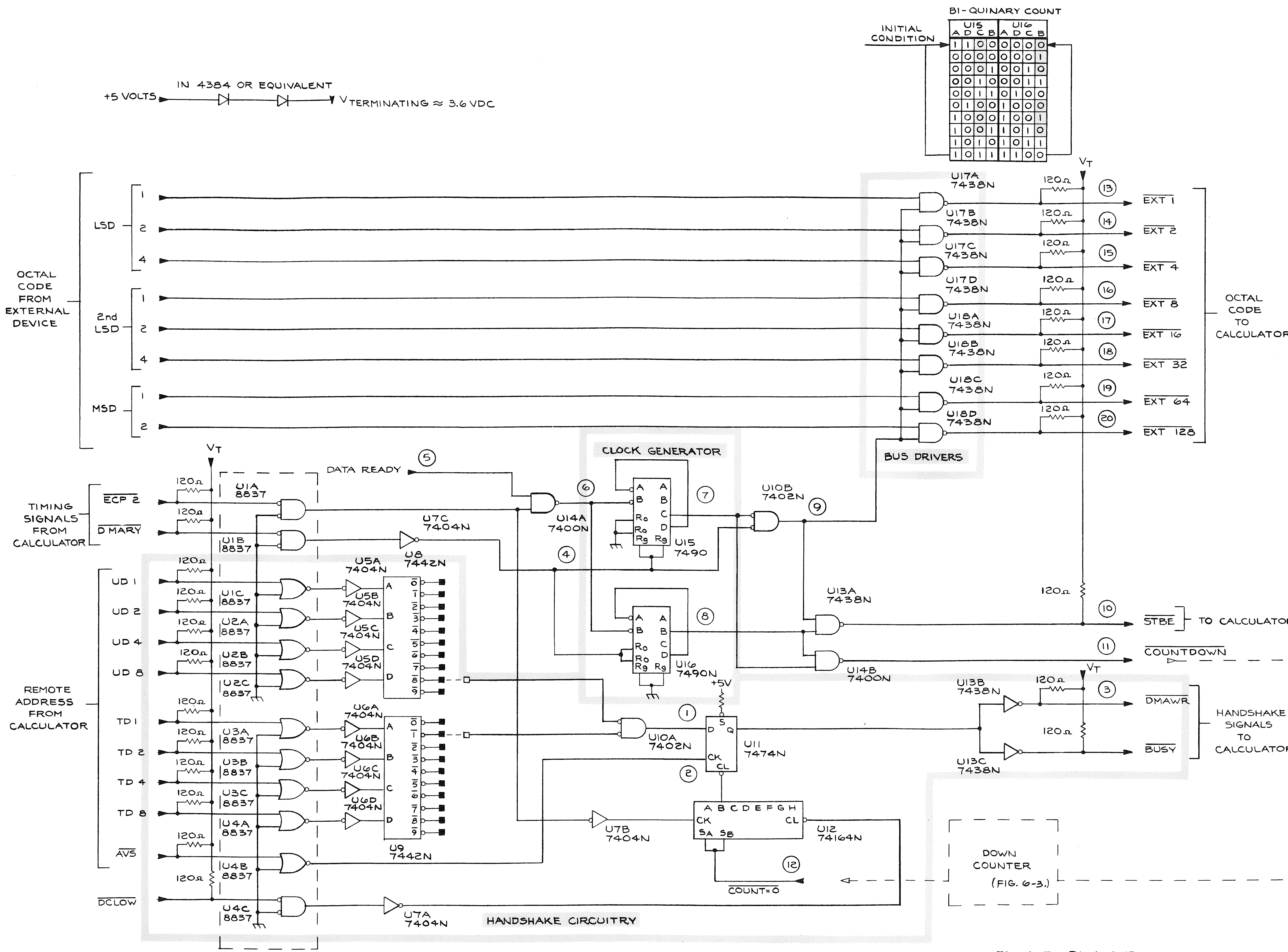


Fig. 6-5. DMA WRITE

FIG. 6-5. DMA WRITE

21 & 31 INTERFACING INFORMATION

Ⓐ

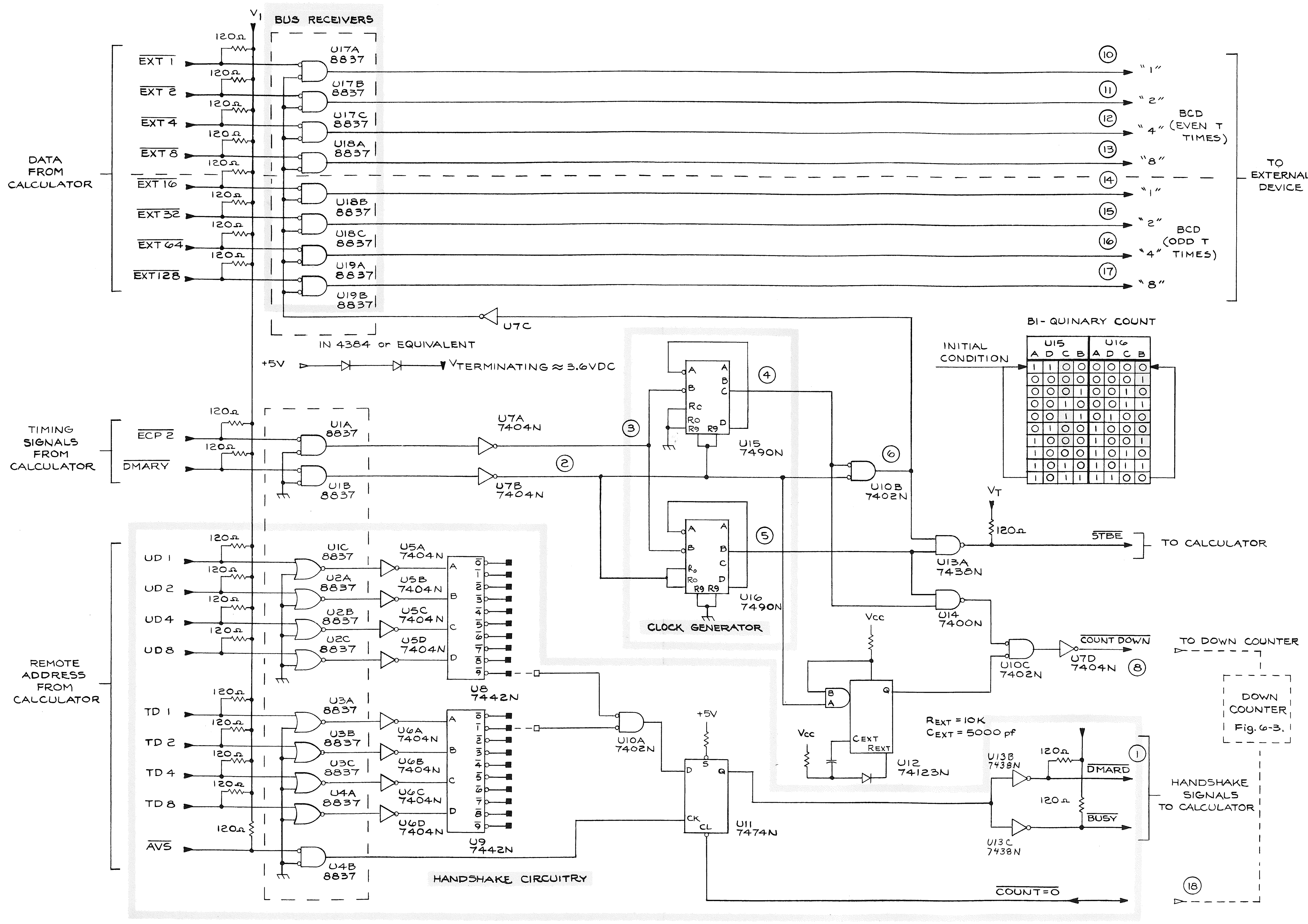


Fig. 6-6. DMA READ
(R-REGISTER DATA)

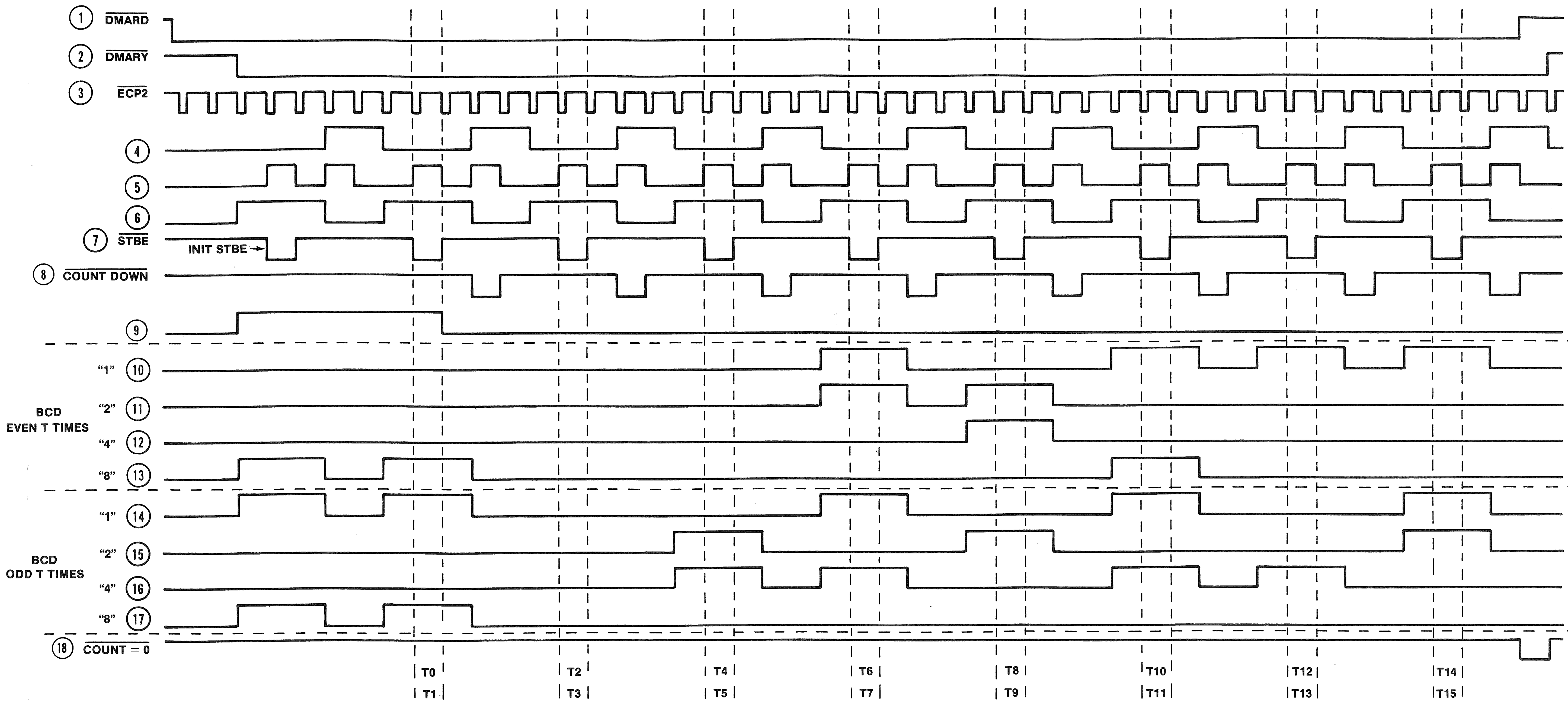


Fig. 6-7. DMA READ (R-REGISTER DATA) TIMING DIAGRAM.

Ⓐ

FIG. 6-7. DMA READ
TIMING DIAGRAM