THE DATA HANDLER

OWNER'S MANUAL

REV-B
C.R.-2-8-77
by C.A.Indihar

Western Data Systems

3650 Charles Street • Suite G • Santa Clara, Calif. 95050 • (408) 984-7804

WARRANTY

KITS:

Western Data Systems hereby warrants all Data Handler kits that, 1) during the first thirty (30) days of ownership all components supplied by W.D.S. to be free from defects in materials and workmanship, and 2) in the event of component failure or malfunction said components will be repaired or replaced free of charge at the descretion of W.D.S. when returned to the factory postage paid.

Any returned Data Handler kits which have been fully owner assembled and require factory repair due to normal use and service within the first thirty (30) days of ownership, will be repaired to operating order for a nominal shipping and handling charge of $10.00.

ASSEMBLED UNITS:

During the first three (3) months of ownership, any factory assembled units, that we determine, under conditions of normal use and service, fail to perform according to our published specification, will be repaired or replaced, at no charge to you, at the W.D.S. factory.

The above warranty applies to the original purchaser only, and does not cover damage by use of acid-core solder, incorrect assembly, misuse, fire, floods or acts of God. If W.D.S. or a representative there of determines that your Data Handler system needs repair due to said conditions, a charge greater than the nominal amount may be required.

-CONTENTS-

INTRODUCTION

The Data Handler is a complete microcomputer system on a single printed circuit board (P.C.B.) designed around the Mos Technology 6502 microprocessor.

This complete microcomputer system contains one thousand words (1K bytes) of random access memory (ram), one eight bit parallel input port with data latch and interrupt strobe, and one eight bit parallel output port with clearable data flag. The Data Handler contains a 26 keyboard switch hardware controlled front panel which will load data, examine data, perform single cycle and single instruction, initialize the system, run and halt the system all in hexidecimal (Base 16) format.

It contains a complete on board discrete L.E.D. hexidecimal address and data display which display the sixteen address bus lines and the eight data bus lines.

Also on the Data Handler P.C.B. are three discrete L.E.D.'s which indicates whether the Data Handler front panel is in the address or data entry mode and whether it is halted or running.

INTRODUCTION:

The left rear portion of the P.C.B. contains two etched rows of fifty pads to facilitate the installation of one, one hundred pin Altair/IMSAI type P.C.B. edge connectior, or suitable connector to allow bus expansion.

These pads contain the tri-state buffered data, address, memory, I/O (input/output) signals, and timing signals used by the 8800 Altair/IMSAI type peripheral boards. These bus lines are pin for pin compatable, and drive/receiver matched with the 8800 Altair/IMSAI microcomputer peripherals.

These peripheral boards are the least expensive, most widely available, and offer the greatest variety in microcomputer peripherals available on the face of the earth.

The Data Handler is ideally suited as a computer instructing device due to it's easy to understand and use qualities yet retains the capabilities for gross system expansion by the advanced user, further, the Data Handler is an excellent choice as an enexpensive tool for providing computer control for machinery, video displays, or electronic projects.

Note that the assembled Data Handler requires an external power source +5 volts DC at about 1.8 amps. However a power supply capable of supplying 4 to 6 amps is recommended for future expansion.

THE DATA HANDLER
SPECIFICATIONS:

Uses the Mos Technology 6502 microprocessor.

Directly addresses 65K bytes of memory.

Contains 1K bytes of 500 ns ram on the P.C.B. (address FC00-FFFF).

Slow down circuitry for slow memories.

Direct memory access (DMA) circuitry and DMA acknowledge control lines.

Dual interrupt lines-one maskable, and one non-maskable.

On board dual I/O ports, one 8 bit latching parallel input port with interrupt strobe, and one 8 bit latching parallel output port with a data flag.

Variable speed R/C clock, or optional crystal may be used for frequency stability to 2 mhz.

8800 Altair/IMSAI identical tri-state address and data bus.

Uses single +5 volts supply @ 1.8 amps.

Large on the board ground plane for noise suppression at high operating frequency.

Discrete L.E.D.'s for address and data display.

Full function hardware controlled front panel.

Keyboard type data and control switches in hexidecimal format.

SYSTEM INFORMATION:

SYSTEM SPEED: The maximum operating speed of the small or expanded Data Handler system is determined by two factors. 1. C.P.U. speed, and 2. memory access time.

The 6502 microprocessor is presently available in models with operating speeds to 4 mhz (250 ns cycle time). Due to the internal characteristics of the 6502, minimum memory access time (reading stable data from memory) is to be approximately half the cycle time. A system operating at 4 mhz needs a memory with a 125 ns access time, 250 ns for 2 mhz, 500 ns for 1 mhz (supplied in the Data Handler complete kit), and so on. To double the normal data throughput of the Data Handler (operate at 2 mhz), a 6502A microprocessor should be installed, change R65 to a 1K 1/4 W, and change the memory I.C.'s (D1-D8) to 2102A-2. For general system usage the kit supplied RC timing network is perfectly adequate. By setting the 50K ohm timing potentiometer (VR1) midway, a normal system clock speed of approximately 1 mhz will be in operation with a stability factor ± 20%. For time critical rock hard system stability greater than ± 20% a crystal may be incorporated into the processor timing.

CRYSTAL OPERATION: If crystal frequency stability is desired for the Data Handler system a user supplied crystal may be installed in the two mounting holes on both sides of XL1 on the right rear portion of the P.C.B..

SYSTEM INFORMATION:

(CON'T) CRYSTAL OPERATION: This XL1 layout was designed to accomidate an AUGAT 8000-D crystal holder, however the crystal may be soldered directly on the P.C.B. then laid down onto the adjacent ground plane area, and the can (crystal cas soldered to the ground plane directly beneath it with a short piece of resistor lead or other type of wire.

SPARE LOCATIONS: The Data Handler P.C.B. contains three spare I.C. locations (C1,C2,D11), these spares can be used fo any user designed changes made to the circuitry to tailor it for some particular use. Circuitry modifications using these spare I.C. locations can best be made with 30GA. wire wrap wire on the under side of the Data Handler P.C.B. using the point to point wireing technique.

1111 1100 0000 0000

MEMORY MAP:

The Data Handler contains one thousand words (1K bytes) of random access memory (ram) on the P.C.B. These are decoded as address FC00 thru FFFF, and are used for program storage and processor vectoring which is located in the following memory locations:

FFFA- Low order address for NMI
FFFB- High order address for NMI

FFFC- Low order address for front panel INT
FFFD- High order address for front panel INT

FFFE- Low order address for program interrupt
FFFF- High order address for program interrupt

When additional ram memory is used in the Data Handler system, it is recommended that this memory be located beginning at address 0000 to fully utilize the stack function of the processor.

The address locations 7F00 thru 7FFF are decoded on the Data Handler P.C.B. as the I/O device code area. (the area in memory designated for input and output devices).
It is recommended that memory not be decoded to this address area.

0000 0000 0000 0000
1111 1111 1111 1010
1111 1100 0000 0000
I/O DEVICE { 0111 1111 0000 0000
0111 1111 1111 1111

I/O PORT INFORMATION:

The two parallel data ports on the Data Handler are the immediately available means for communication with external devices, the input port receives the data and the output port transmits the data. These ports exit the Data Handler by means of solder pads on the right rear portion of the P.C.B.. The individual solder pads are labled for their signal identity and were designed for wire connection directly to the external device. If different devices are to be used at different times it is suggested that a cable connector (a 15 pin cinch connector works excellent) be wired on the bottom of the port solder pads with matting connectors on each external device. Examples of input devices are a TVT type video display, cassett interface, OP-80A papaer tape reader, radio shack 63 key keyboa ASCII encorder board, ect. Data entering the Data Handler fro an external device can be seen visually on the data L.E.D.'s by halting the Data Handler and examining the input port (address 7FFF).

The output port has an endless number of uses. Examples include control of solor heating, sprinkler control, computerized liquor/wine dispensor, video generation, machine control, ect.

I/O PORT INFORMATION: (EXAMPLES)

FIGURE 1. Ilustrates a very simple programable L.E.D. bit display which can be built in ½ hour for the cost of about $1.50 and can be used directly on the output port.

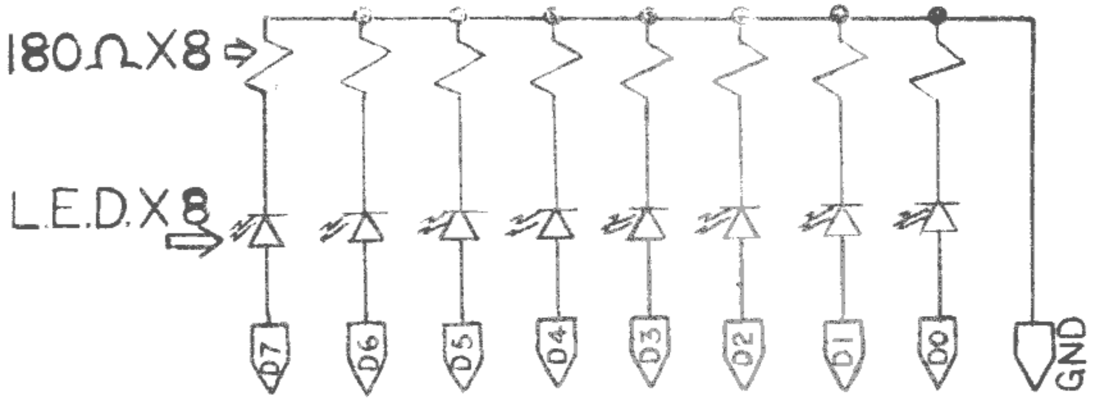
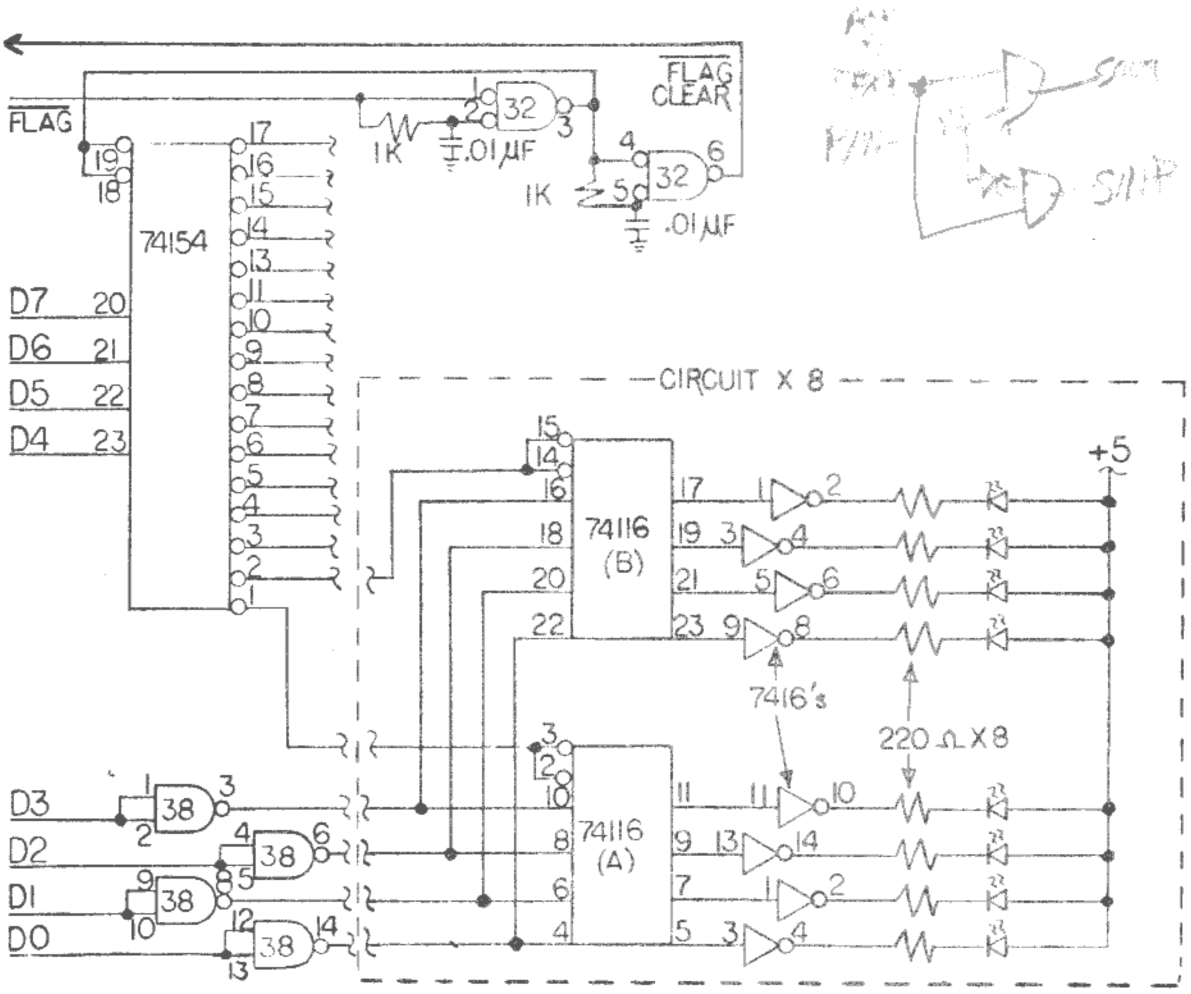


FIGURE 2. Shows a more complex programable display with 64 L.E.D.'s.



8

I/O PORT INFORMATION:

One eight bit parallel latching input port (I.C.E16) is available on the right rear portion of the P.C.B.. It is designated as address 7FFF and is loaded with positive data on the input strobe of a low to high pulse. This data strobe will output a data available flag to the Data Handler interrupt line, this interrupt line which is located to the rear of pin 24 on I.C.E16 looks like a double ended arrow. This if desired may be severed with a sharp knife. Cutting this line will isolate only the input port data available flag from the interrupt line.

Data loaded in this port and the interrupt flag will automatically be cleared to zero immediately after the cycle in which it is addressed by the user program.

One eight bit parallel latching output port (I.C.E12) is available to the left of the input port. This port is designated as address 7FFF and contains two output control signals,
(1) a data flag, and (2) a flag clear.
The data flag (data available signal) when in the high state, indicates that a data word has been latched up and is available in this port. The flag clear is an input signal to this port and when in the low state will clear the data and the data flag.

Both the input and output ports are latched/buffered with 8212 (74S412) port buffers, and are capable of driving 40 MA each making them suitable for directly driving power transistors (2N3055's), SCR's ect.

OPERATING INSTRUCTIONS:

The Data Handler hardware controlled keyboard is composed of 26 keyboard switches which are divided into one group of ten, and one group of sixteen. The group of ten contains the function switches

DP (Deposit)
EX (Examine)
CL (Clear)
AD (Address)
DA (Data)
SI (Single Instruction)
SC (Single Cycle)
HT (Processor Halt)
RN (Processor Run)
INT (Initialize)

The group of sixteen switches contains 0 thru F (hexidecimal) for data and address loading.

The Data Handler information display is comprised of 27 discrete L.E.D.'s (light emitting diodes), 26 red and one yellow. This display is divided into a row (bottom row) of 16 L.E.D.'s which represent in hex format the current address, and the top row of 8 represent the current data. the M.S.B. (most signifigant bit) of each row is the left most L.E.D.

OPERATING INSTRUCTIONS:

In the group of three L.E.D.'s to the left of the keyboard, the top one labeled data indicates that the keyboard switches 0 thru F will be displayed on the data L.E.D.'s. Pressing the AD function switch will change the loading to the address L.E.D.'s. The bottom yellow L.E.D. indicates that the Data Handler is free running when it is on, and in the halted mode when it is off.

TO DEPOSIT DATA:

Press HT--Halt the Data Handler (disregard if already halted)

Press CL--Clear the internal mode and the data holding registers.

Press AD--To switch the loading to the address lines (address L.E.D.'s)

Key in the desired address to be modified.

Press DA--To switch the loading to the data lines (data L.E.D.'s)

Key in the desired data to be loaded in the present displayed address.

Press DP--This will deposit the data currently on the data lines into the address that is currently on the address lines.

After making the deposit into memory the Data Handler will automatically increment to the next address location, so that only the data needs to be altered and deposit pressed again for loading subsequent instructions.

OPERATING INSTRUCTIONS:

The Data Handler has a deposit lock out when in the examine mode to prevent inadvertant depositing of data. To switch from the examine mode to the deposit mode or vise versa the Data Handler must be exercised through one complete instruction (press SI once) before the desired mode can be selected.

TO EXAMINE DATA:

PRESS HT--Halt the Data Handler (disregard if already halted).

PRESS EX--Put the internal registers in the examine mode.

PRESS AD--To place the keyboard entry on the address lines.

Load the address to be examined onto the address lines, and the data L.E.D.'s will then display the data in that location of memory.

When in the examine mode, each time the EX function switch is pressed the next address and it's data will be displayed.

Depositing and examining data will not affect the PC counter of the 6502 microprocessor, in that the Data Handler can be halted, data examined and or deposited without disturbing the existing running program sequence.

OPERATING INSTRUCTIONS:

SINGLE INSTRUCTION:

PRESS HT--Halt the Data Handler (disregard if already halted)

PRESS SI--The Data Handler will execute the next instruction and stop. Each time the SI switch is pressed the Data Handler will execute the next instruction and stop.

SINGLE CYCLE:

PRESS HT--Halt the Data Handler (disregard if already halted)

PRESS SC--The Data Handler will execute one cycle of the next instruction and stop.

Using the SI and SC modes to "walk" through a newly loaded or newly written program can save countless hours in the debuging of software, in that branches, jumps, all steps of the program are displayed and executed under full control of the operator.

INITIALIZE:

When the INT switch is pressed the 6502 MP will begin the initialize process which consist of 6 cycles in which it will dump it's present address on the stack, and load it's program counter (PC) from it's initialize vector locations. The Data Handler will then begin executing at that memory location when run is pressed.

OPERATING INSTRUCTIONS:

(To demonstrate this let's load a small loop program)

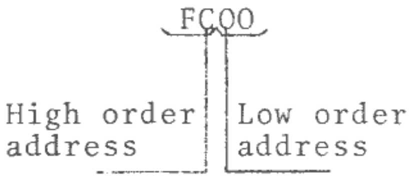
Deposit the following data starting at address location FC00 by using the format discussed for depositing data.

```
ADDRESS          DATA                 INSTRUCTION

FC00--------------EA                      NOP
  01--------------EA                      NOP
  02--------------4C                      JMP
  03--------------00                      LOW ORDER ADDRESS
  04--------------FC                      HIGH ORDER ADDRESS
```

In the halt condition, to instruct the Data Handler to run the above program you would simply load the low order address of your program starting location into the first location of the initialize vector (FFFC), and the high order address into the second location of the initialize vector (FFFD), shown below.

Our loop program starts at FC00 so



So we load;

```
ADDRESS    DATA

   FFFC----00    (low order address)
   FFFD----FC    (high order address)
```

OPERATING INSTRUCTIONS:

To get the Data Handler to begin execution of our loop program press INT, begin pressing SC and watch the Data Handler step through the initialize process (6 steps). It will step to the initialize vectors (FFFC, and FFFD) and arrive at its new starting address FC00 which should be displayed on the address display. If your address is displaying 00FC you loaded your low and high order address backwards, if so go back and try again.

If you arrive at the correct address (FC00), continue to single cycle through your loop. If you are convinced that it is looping properly you can discontinue "walking" through and let it free run by pressing RN. The Data Handler is now executing your loop program.

You will notice that the Data Handler continuously displays the exact data that we loaded in our loop program. Excellent! It's running our program exactly as we wrote it. We instructed it to perform two NOP's (EA, no operations) and then JMP (4C, jump to a new starting location) to begin execution of new data, which as we planned it turned out to be the same program we left. Very cleaver is this program, this could be the basis for some sort of keyboard monitor or other program that looped while waiting for an interrupt from an external device (possibly connected onto our input port).

SYSTEM EXPANSION:

The Data Handler address, data, and control lines exit the P.C.B. by way of two rows of fifty pads on the rear left portion of the Data Handler P.C.B..

These bus lines are pin for pin compatable, and driver/receiver matched with the 8800 Altair/IMSAI microcomputer peripherals.

Limited system expansion may be achieved with the addition of a 100 pin connector on the bus line pads on the left rear portion of the Data Handler P.C.B. and the installation of any particular 8800 Altair/IMSAI type of peripheral board which fits the individual need.

Larger system expansion may be accomplished with the use of an Altair type mother board or a suitable card cage setup mounted next to the Data Handler P.C.B. with the bus lines connected to the mother board with ribbon cable. Another excellent choice would be the hard wire mounting of an Altair expander board directly perpindicular below the Data Handler P.C.B.. This would enable the use of more than one peripheral board.

The Data Handler may be used with a power supply greater than +5 volts D.C. when it is configured in such a manner as to incorporate the mounting of a +5 volt 5A regulator on the P.C.B.. The left rear part of the P.C.B. directly adjacent to the bus line pads, is laid out to accept a TO-3 78H05, or LM 323, (+5 volts @ 5 amps) type regulator.

SYSTEM EXPANSION:

This regulator configuration is ideal when used with an Altair type (+8 volts) power supply to provide power for peripheral boards of the 8800 Altair/IMSAI type. Note that these peripherals were designed to be powered by +8 volts with regulators dropping the voltage to +5 volts on each individual board, however, jumpers may be used in place of these regulators and a +5 volt power supply may be incorporated.

The Data Handler L.E.D. displays are driven directly from the address and data bus lines. When the system is expanded to a large degree it is suggested that a video type peripheral be added, or that the L.E.D.'s be buffered, or seven segment displays added to prevent over loading of these bus lines.

Note that with the tremendous popularity towards 8800 Altair/IMSAI type peripherals there will be an increasingly large number of them available in the future and the possibility exists that some of them will not immediately work with the Data Handler.

*If this happens, we ask that we be contacted at Western Data Systems (408) 984-7804 so that we may investigate and resolve the problem.

DO NOT ASSEMBLE BEFORE READING COMPLETELY
****

PRE-ASSEMBLY NOTES:

(READ CAREFULLY)

1. Assembling of the Data Handler P.C.B. consists of mounting components to the board. The parts location list (see index) contains lettering identifying where the components are to be mounted on the board along with their proper position and direction. We recommend that you mount the components in groups, such as the integrated circuits, then the resistors, and then the capacitors. A step by step assembly procedure is provided for each major group of components.

2. The Data Handler printed circuit board is industrial grade, double sided , with plated through holes. This means that in a number of locations, the foil paths on the top of the board are connected to the foil paths on the bottom through holes which have plating on the inside. This plated through technique means you don't have to solder common foil areas on both sides. Soldering on the bottom side of the P.C.B. only normally will be adequate.

(*NOTE- DO NOT USE ACID-CORE SOLDER)

PRE-ASSEMBLY NOTES:

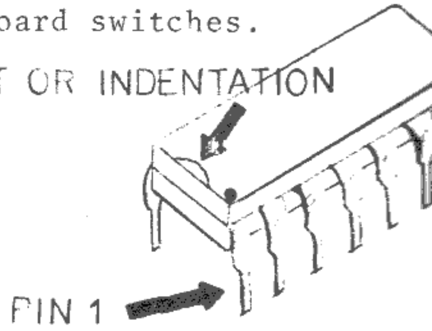
3. Due to the small foil around the printed circuit board holes and the small areas between the foil traces, you will have to use utmost care to prevent solder bridges between adjacent foil areas. Use only a low-wattage soldering iron with a small tip. Do not use a soldering gun. Keep the tip of your soldering iron clean (frequently wipe the tip on a damp clothe or solder sponge)and lightly coated with solder.

4. The Data Handler kit contains (8) 16 pin I.C. sockets, for the 2102 memory, and (1) 40 pin I.C. socket for the 6502 microprocessor. Sockets may be used for all the I.C.'s. However, for system reliability sake they are not recommended.

5. The side with the name Data Handler is the side on which all the components are to be loaded.(component side)

6. On all I.C.'s facing the rear of the P.C.B., pin 1 is the left rear pin (denoted by the clipped pad on the P.C.B.) The microprocessor (E7), and the 8212's (E12, and E16) face to the left with pin 1 located on the front left row of pins (denoted by clipped pad on the P.C.B.).

7. I.C. B7 is the only one mounted with pin number one to the right rear facing toward the keyboard switches.



PRE-ASSEMBLY NOTES:

8. When installing the L.E.D.'s be absolutely sure of their polarity before installation. (See figure 3) Test them first if necessary, but be sure. The base of the L.E.D. contains a small notch or flat area, this denotes the cathode of the L.E.D. as the lead closest to the flat area.

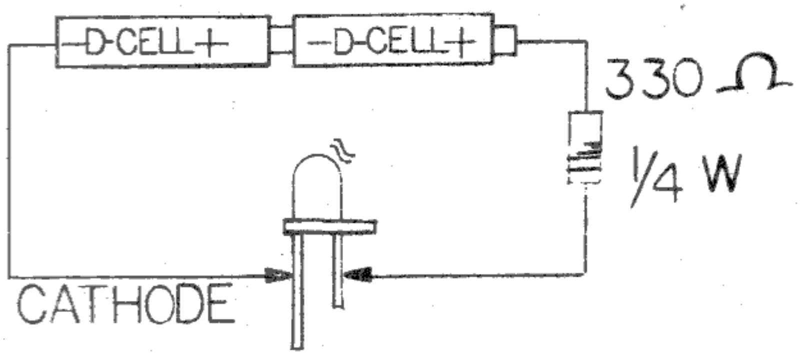


FIG.3

9. The cathode is to be connected to ground on all of the L.E.D.'s. On M4 thru M27 the cathodes face the front of the P.C.B., on M1, M2, and M3 they face toward the keyboard.

CAUTION*

```
***********************************************************
*        When inserting the static sensitive 6502 micro-  *
*                                                         *
*processor into it's socket, it is best done with one hand*
*                                                         *
*wrapped around a cold water pipe while the other hand    *
*                                                         *
*gently inserts the I.C. into it's socket. Avoid handling *
*                                                         *
*this I.C. until actual insertion time.                   *
*                                                         *
***********************************************************
```

ASSEMBLY INSTRUCTIONS:

1.() With the P.C.B. component side up (see preassembly note 5). Insert and solder pins 1 and 9 of the memory I.C. sockets (D1 thru D8) on the P.C.B..

2.() Insert the microprocessor socket and tack solder the corner pins (1 & 21) onto the P.C.B..

3.() Refering to the component layout diagram, open one package at a time, and tack solder the corner pins on all the I.C.'s except B7 onto the P.C.B..

Do not install the memory I.C.'s or the microprocessor in their sockets yet.

4.() Install I.C. B7 according to preassembly note 7.

5.() Check all I.C.'s and sockets for alignment, full insertion, bent under pins, and orientation (is B7 correct?).

6.() With all the I.C.'s being correct, go back and finish soldering all the I.C. pins. Make sure of good clean solder joints, and lack of solder bridges.

7.() With all the I.C. sockets being correctly orientated, solder all the pins as in step (6).

8.() Again refering to your component layout diagram, open the package of resistors and istall them by groups in the following manner on the P.C.B.. Install the 4.7K ohm (yellow-violet-red) resistors in R1-R24 and solder in place.

9.() Insert R26-R51 (220 ohm, red-red-brown) and solder into place

10.() Insert R54-R64 (1K ohm, brown-black-red) and solder into place.

11.() Again refering to the components layout diagram insert and solder the remaining resistors along with the 50K ohm potentiometer (VR1).

12.() With the resistors properly in place, mount the capacitors being careful to observe proper polarity on the electrolytics C20, and C21. (Plus on the P.C.B. should correspond with the plus end on the capacitor) (Refer to component layout).

ASSEMBLY INSTRUCTIONS:

13.() Double check the placement, polarity alignment, ect. of the capacitors, and solder them into place. Trim (if needed) the excess lead length from these components from the underside of the board at this time.

14.() Insert the data L.E.D.'s M20 thru M27 (back row of 8 red L.E.D.'s) with the cathode (see preassembly notes). Toward the front edge of the P.C.B. (the long edge with numbers 1 thru 11 along it). Mount them all flush with the surface of the P.C.B.. First solder one lead in place, then after making sure of their alignment and uniformity solder the other lead.

15.() Install the row of 16 red address L.E.D.'s M4 thru M19 using the procedure previously described (step 14) for the data L.E.D.'s (cathodes toward the fromt edge of the P.C.B.). Also mount them flush with the P.C.B. and solder into place.

16.() Install the red addr L.E.D. M2 and red data L.E.D. M3 (located just to the left of the keyboard). Mount them flush with the P.C.B. with their cathodes towards the keyboard, and solder into place.

17.() Insert the yellow L.E.D. (M1) in its place directly below M2. Mount it with the cathode toward the keyboard and it also flush with the P.C.B., with alignment being correct solder it into place making sure to produce good solder joints.

18.() Install any user provided P.C.B. edge connector (Altair bus type), crystal, connectors for the I/O ports or what have you at this time and solder into place. Turn the P.C.B. over and ensure that all component leads are clipped and properly soldered.

19.() The P.C.B. without the switches installed, may be cleaned of any flux, dirt ect. at this time. At this point be careful not to let this sludge accumulate in the microprocessor or memory sockets. There are special chemicals available for cleaning P.C.B.'s (freon TF, flux solvent, M.E.K. ect.) but they should be used with proper ventilation If the keyboard switches are already installed be extremely careful about keeping this flux remover off of them, as it could result in destroying the switches. It is recommended that you remove the switches before the board is cleaned.

ASSEMBLY INSTRUCTIONS:

20.() Install the 26 keyboard switches one at a time soldering them from the bottom side when alignment, placement, and registration are verified with the component layout diagram.

21.() At this time insert the Data Handler P.C.B. into the top track of the Data Handler P.C.B. case.

22.() Connect and solder the wires from the user supplied power supply (plus 5 volts D.C., producing a minimum of 1.8 amps). The Data Handler requires only a plus 5 volts for power, however, consult the system expansion section (7) for other power configurations.

23.() Turn power to the Data Handler on at this time and quickly look around the P.C.B. and check for smoke, excessive heat or any abnormality. If any abnormality is found procede immediately to smoke, heat, and abnormalities in the system checkout section.

24.() If a D.C. volt meter is available check for the plus 5 volts at pin 8 on the microprocessor socket and ground (0 volts) at pin 1. If a meter is not available wait an extra moment to ensure that an abnormality will not suddenly pop up later on.

25.() All steps being valid up to now, turn off the 5 volt supply to the Data Handler and install the memory I.C.'s into their sockets. Install the microprocessor into its socket (read how in the preassembly notes) being careful to use the correct pin number 1 for the registration.

26.() Ready to turn the power on? wait, once more check that the memory I.C.'s and the microprocessor are properly in place according to the component layout diagram.

27.() O.K. turn the power on and procede to the system checkout section.

THEORY OF OPERATION:

*The following is a detailed theory of operation of the of the complete Data Handler and its associated circuitry. Refer to the schematic diagrams and the block diagram while reading this theory.

KEYBOARD:

The complete keyboard control is accomplished with the 10 control keyswitches, and the 16 data/address entry keyswitches (3 of 6). The set of 16 data/address entry keyswitches are hardware hex encoded by A8 thru A11 (2 of 6) and "switch sensed" by B6 (2 of 6). I.C. B6 sends a"switch sensed" strobe line signal to the one shot keyswitch debounce circuitry made up of A7, and A6 (1 of 6) which then is decoded by D12 (1 of 6) as being a strobe pulse from one of the 16 possible keyswitches.

FUNCTION GENERATOR:

The strobe pulse from the debounce circuitry activates a four pulse function generator formed from B1,A2, and A3 (2 of 6) and is mode decoded (address mode or data mode) by A4 and the second half of A1 (2 of 6). These 4 address pulses or 2 data pulses then latch (B7 for data and B2-B5 for address) the hexadecimal encoded data presently being displayed thru A8-A11 (2 of 6) by any-one of the 16 possible keyswitches.

THEORY OF OPERATION:

ADDRESS/DATA ENTRY MODE:

Two control keyswitches (3 of 6) control the entry mode of the front panel. The entry mode is selected by either the DA (data) or AD (address) switches. This switch information sets or resets the RS latch formed from the first half of A1 (2 of 6) and decodes the function generator pulses as being either address or data loading pulses. The state of this RS latch is displayed on L.E.D.'s M2 and M3 (2 of 6) and can be changed at anytime.

SINGLE CYCLE AND SINGLE INSTRUCTION:

The SC (single cycle) and SI (single instruction) control keyswitches are debounced by A7 and A6 (1 of 6), and decoded by half of I.C. D12 (1 of 6). The debounced keystroke pulse from the SI (single instruction) keyswitch sets the D15 (1 of 6) flip-flop which in turn allows the RDY line on the 6502 microprocessor to go to the high state. This enables the Data Handler to run until the D15 (1 of 6) flip-flop receives a sync pulse (stat of new instruction) from the 6502 at which time D15 (1 of 6) resets and thru D13 (1 of 6) halts system operation. The SC (single cycle) keyswitch is debounced and wired to RDY in the same way as is the single instruction, however it allows the Data Handler to run only from the start of a Ø1 cycle to the start of the next Ø 1 cycle in which D14 (1 of 6) is reset and system operation is halted.

THEORY OF OPERATION:

CLEAR:

The CL (clear) switch accomplishes two functions when it is actuated.

(1) It resets the RS latch C11 (1 of 6) which gives control of the S-100 bus to the front panel.

(2) It clears the front panel data (B7) and address (B2-B5) latches.

The CL, as is INT, HT,RN,AD, and DA switches are not debounced since they are primarily used to set and reset latches in the associated function circuitry of the Data Handler front panel.

RUN/HALT:

The RN (run) and HT (halt) keyswitches enable us to ultimately control the operation of the Data Handler. These switches either set or reset run latch B9 (1 of 6) which in turn will halt operation of the C.P.U.. The state of this run latch is displayed by L.E.D. M1 (1 of 6) which when lit indicates that the C.P.U. is running. Further more when in the run mode this run latch (B9) disables the debounce circuitry to kill any spurious oscillations which might lead to system noise.

THEORY OF OPERATION:

INITIALIZE:

The INT (initialize) keyswitch serves to accomplish two functions;

(1) It acts as an override to reset both the single instruction and single cycle circuitry.

(2) It pulls the reset line low on the 6502 microprocessor (4 of 6) which initiates the initialization procedure for the Data Handler (see the operating instructions for further information).

BUS:

The S-100 bus is a UNI-directional bus which is the bulk work of the Data Handler. It is driven/received with tri-state buffers by both the front panel and the 6502 microprocessor. This bus drives the discreet L.E.D. display and connects the I/O ports, memory, 6502, front panel, and edge connector pads together.

SYSTEM CHECKOUT:

Smoke, heat, and abnormalities:

O.k., here you are with a problem, so lets be quick. None of the I.C.'s after 15 seconds (with power on), should be too hot to hold your finger on, so touch each one and remember which ones burned your finger. Turn the power off and check the hot (or smokey) I.C. for proper pin registration, alignment, solder blobs, a cut lead resting on the I.C. pins, ect. If I.C. B7 is hot chances are it's installed backwards. If all the I.C.'s are hot check the polarity of the power supply to the P.C.B., the polarity is either backwards (plus 5 volts is connected to ground on the P.C.B.), or the plus 5 volts is not actually plus 5 volts. If your power supply exhibited the smoke or abnormalities, chances are that there is a direct short on the P.C.B., this is easiest found with an ohm meter (look for solder blobs, lead touching under the P.C.B., ect).

*Consult the troubleshooting section for further information.

SYSTEM CHECKOUT:

NORMAL CHECKOUT:

1.() With the HT (halt) button pressed, turn on the power supply to the Data Handler. The Data Handler will come up with a random address on the 16 address L.E.D.'s and random data on the 8 data L.E.D.'s. Either the addr or data L.E.D (closest to the keyboard) will be on but not both, and the run L.E.D. (yellow L.E.D.) should not be on. This run L.E.D. indicates that the Data Handler is halted when not lit and running when it is on.

2.() Press the AD (address) switch, then the DA (data) switch. Ensure that the associated L.E.D. changes with each particular keystroke.

3.() Press CL (clear), the 16 addr and 8 data L.E.D.'s should go off.

4.() Press AD, now press the 1 key. The addr 0 L.E.D. should come on and stay on. This should be the only row L.E.D. on. Press the 2 key, the 1 should have shifted left into the next set of four digits and was replaced by the 2 entry. Sequentially enter the rest of the 16 keys. Ensure that the proper key data is entered into the right most set of four L.E.D.'s each time a key is pressed, verify that this keyswitch data is only entered once per keystroke.

5.() Press DA and sequentially enter all the 16 individual keys as in the previous step. Ensure that they enter the correct data row L.E.D.'s and shift left with each single keystroke.

6.() Enter all F's into both the address and data rows of L.E.D.'s. All the data and address (24 L.E.D.'s) L.E.D.'s should now be on. This test indicates dead or weak L.E.D.'s.

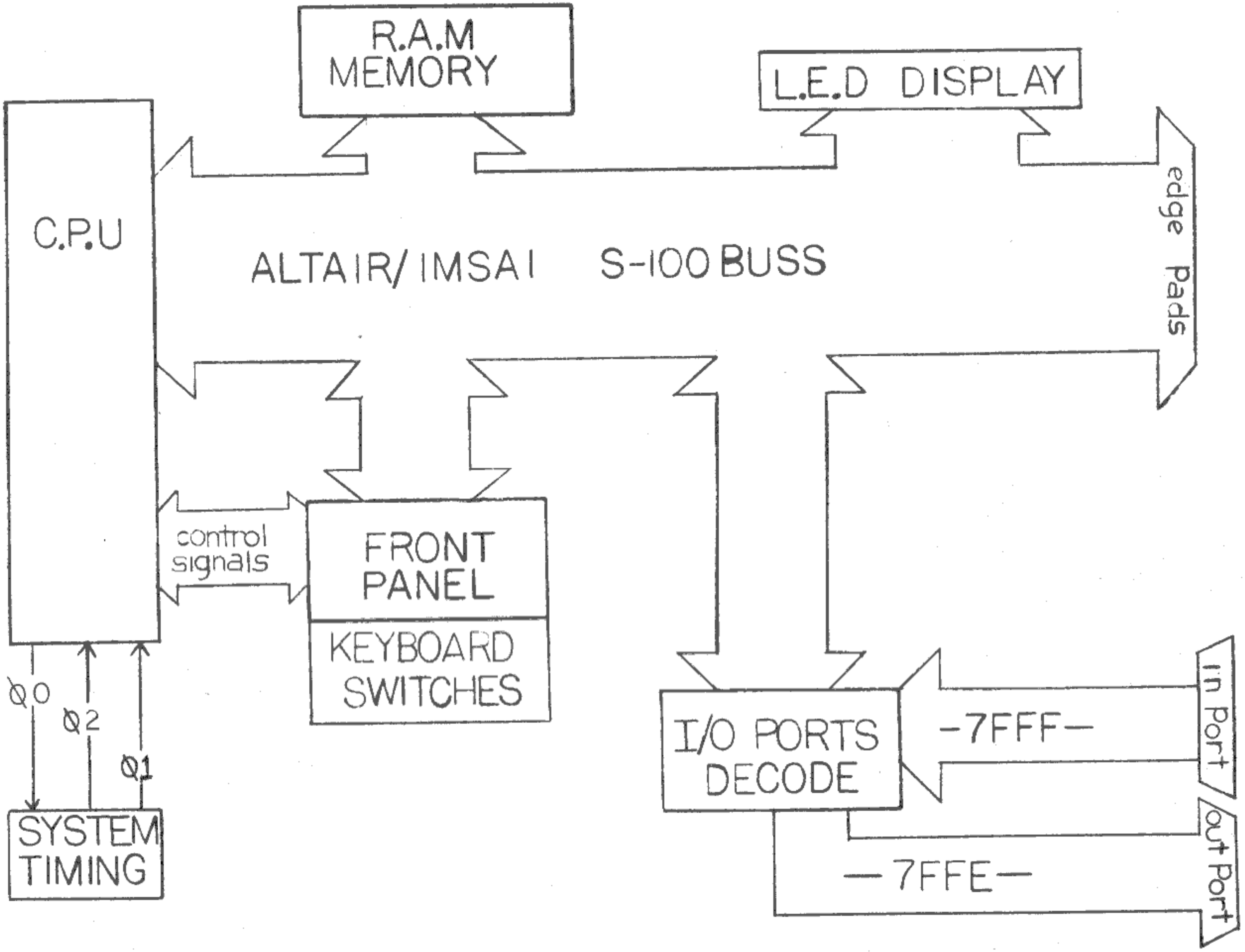
7.() With the Data Handler halted (press HT), press clear, press AD, press the 1 key. The address 0 L.E.D.(right most L.E.D. in the row) should come on. Press (deposit). The address row of L.E.D.'s should increment by one (base 16) each and every time DP is pressed.

SYSTEM CHECKOUT:

NORMAL CHECKOUT:

8.() Duplicate test 7 (on previous page). However, press the EX (examine) keyswitch in place of the DP keyswitch. The address row of L.E.D.'s should increment by one (base 16) and only one each time EX is pressed.

9.() Jump to the operating instructions section (section 6) and become familiar with the correct procedures for operating your Data Handler. The loop program at the end of section 6 will verify proper system operation.



DATA HANDLER BLOCK DIAGRAM

TROUBLESHOOTING:

Before consulting the troubleshooting chart to solve a particular problem, do the following:

(A) Give your Data Handler a complete visual inspection and ensure that all the I.C.'s are properly orientated and that the ones in the sockets are inserted sufficiently without bent-under, or out of socket pins.

(B) Check for solder bridges, cut leads laying on or under the P.C.B., and any solder flakes that can possibly cause a short to the Data Handler.

(C) If steps A & B failed to cure the problem list all visible symptoms and procede to the troubleshooting chart. Remember it is virtually impossible to troubleshoot a problem without proper knowledge of the symptoms.

TROUBLESHOOTING CHART

SYMPTOM/INDICATION	PROBABLE CAUSE
(1) No lights (L.E.D.'s).	1. Major short on the P.C.B 2. Power supply dead. (check the +5 volts) 3. L.E.D.'s installed backwards.
(2) Lights (L.E.D.'s) dim.	1. Insufficient voltage from power supply (4.5 volts to 5.5 volts) 2. Insufficient current from power supply (1.8 amps min). 3. Tri-state bus buffers bucking one another. Either the front panel buffers (C3-C6) should be on or the C.P.U. buffers (E3,E4,E5,D9,D10 but not both. 4. S-100 bus is shorted.

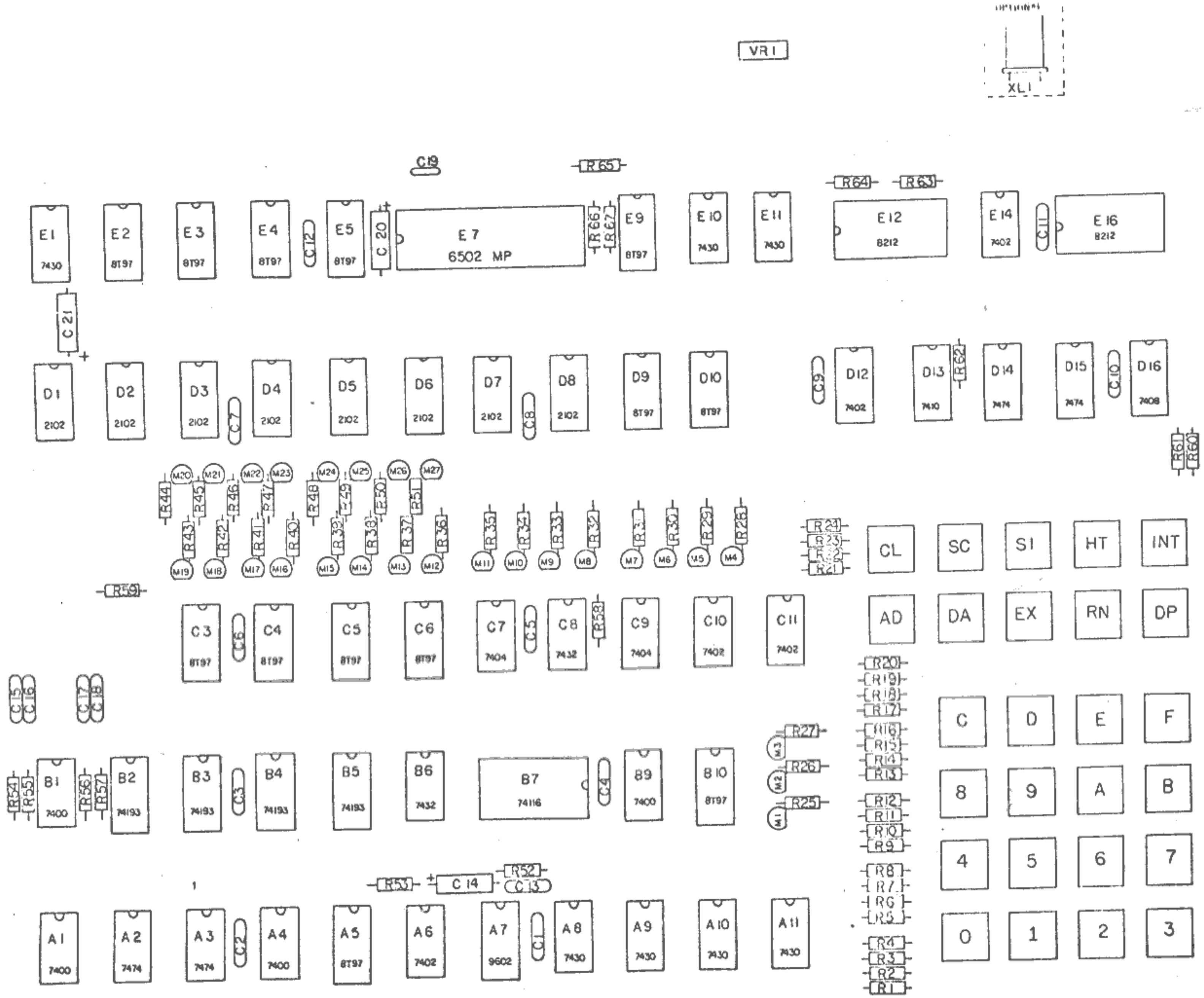
TROUBLESHOOTING CHART

SYMPTOM/INDICATION	PROBABLE CAUSE
(3) Run L.E.D. not on or won't go off.	1. Microprocessor clock not running. 2. B9 not switching on switch contact. 3. Run L.E.D. is dead.
(4) Addr and data L.E.D.'s both on or won't switch off.	1. One or both L.E.D.'s dead or installed backwards. 2. A1 not switching. 3. B10 not driving the L.E.D.'s
(5) CL won't clear the addr and data L.E.D.'s.	1. The Data Handler is not in the halted mode. 2. C11 not functioning properly.
(6) Incorrect data entered from front panel.	1. One or more dead L.E.D.'s (do test 6). 2. A8-A11 not decoding properly. 3. Front panel buffers (C3-C6) not providing drive or shorted. 4. B6 is dead (check strobe line 2 of 6).
(7) A keystroke enters more than one character.	1. A7(A) runs too fast (increase R53). 2. Sticky keyswitch. 3. Function generator (A2,A3,& B1) functioning improperly.
(8) Not able to enter any address or data digits.	1. A6,A7 debounce circuitry is dead. 2. B6 is dead 3. A8-A11-one or all indicate a false key closure. 4. Stuck keyswitch.

* Part numbers-Refer to I.C. locations (C3 is an I.C. location not capacitor C3 unless stated otherwise).

TROUBLESHOOTING CHART

SYMPTOM/INDICATION	PROBABLE CAUSE
(9) Unable to enter any data	1. Check the orientation of B7 (see preassembly note 7)
(10) EX or DP increment more than one 16.	1. A 7(A) runs too fast (increase R53). 2. Keyswitch sticky.
(11) Data Handler looses running sequence.	1. Clock speed is too fast. 2. Memory I.C. failure. 3. Interrupt line (from input port) continuously held low. 4. Memory I.C. in backwards.
(12) C.P.U. won't run at all.	1. The 6502 (I.C.E7) installed backwards. 2. VR1 broken or not installed.



COMPONENT LAYOUT DIAGRAM

THE DATA HANDLER

I.C. LOCATION/PARTS LIST

A1-	7400
A2-	7474
A3-	7474
A4-	7400
A5-	8T97
A6-	7402
A7-	9602
A8-	7430
A9-	7430
A10-	7430
A11-	7430
B1-	7400
B2-	74193
B3-	74193
B4-	74193
B5-	74193
B6-	7432
B7-	74116
B9-	7400
B10-	8T97
C1-	SPARE
C2-	SPARE
C3-	8T97
C4-	8T97
C5-	8T97
C6-	8T97
C7-	7404
C8-	7432
C9-	7404
C10-	7402
C11-	7402
D1-	2102
D2-	2102
D3-	2102
D4-	2102
D5-	2102
D6-	2102
D7-	2102
D8-	2102
D9-	8T97
D10-	8T97
D11-	SPARE
D12-	7402
D13-	7410
D14-	7474
D15-	7474
D16-	7408
E1-	7430
E2-	8T97
E3-	8T97
E4-	8T97
E5-	8T97
E7-	6502
E9-	8T97
E10-	7430
E11-	7430
E12-	8212
E14-	7402
E15-	8212

R1-R24---4.7K ohm
R25------150 ohm
R26-R51--220 ohm
R52-R65--10K ohm
R53------27K ohm
R54-R64--1K ohm
R66-R67--3.3K ohm
VR1------50K ohm
(TRIMPOT)

C1-C12---.01MF
C13------68PF
C14------4.7MF
C15-C18--220PF
C19------10PF
C20------10MF
C21------47MF

4-	7400	
5-	7402	
2-	7404	
1-	7408	
1-	7410	
7-	7430	
2-	7432	
4-	7474	
1-	74116	(93116-9308)
4-	74193	
2-	8212	(74S412)
13-	8T97	(8097-74367)
1-	9602	
8-	2102	
1-	6502 MP	

CAPACITORS

1-47MF--15V ELECTROLYTIC
1-10MF--10V ELECTROLYTIC
1-4.7MF-10V TANTALUM
12-.01MF
4-220PF
1-68PF
1-10PF

RESISTORS

1-27K ohm
1-50K ohm TRIMPOT
2-10K ohm
24-4.7K ohm
2-3.3K ohm
11-1K ohm
26-220 ohm
1-150 ohm

L.E.D.'s

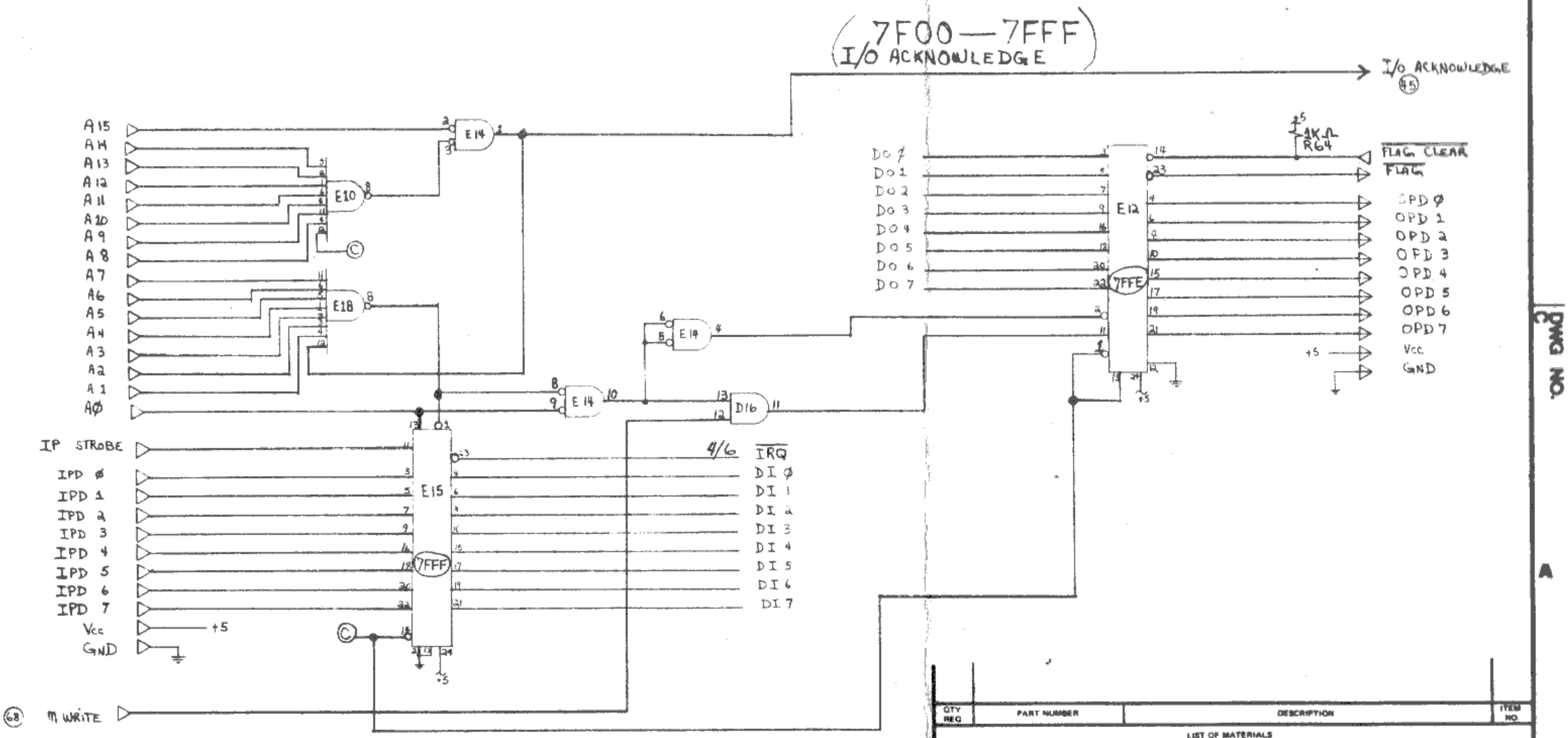
26-RED L.E.D.'s
1--YELLOW L.E.D.'s

*** ALL RESISTORS ARE ¼W UNLESS OTHERWISE NOTED ***

DASH NO.	NEXT ASSY

ZONE	ISSUE	DESCRIPTION	DATE	APPROVAL

REVISIONS



QTY REQ	PART NUMBER	DESCRIPTION	ITEM NO.

LIST OF MATERIALS

DRAWN K.SER
DESIGNER DEINDIHAR
CHECKED
APPROVED

TOLERANCES
UNLESS OTHERWISE SPECIFIED
ANGULAR ± 30'
FRAC ± 1/64 XXX ± .005
.XX ± .01 XXXX ± .0005

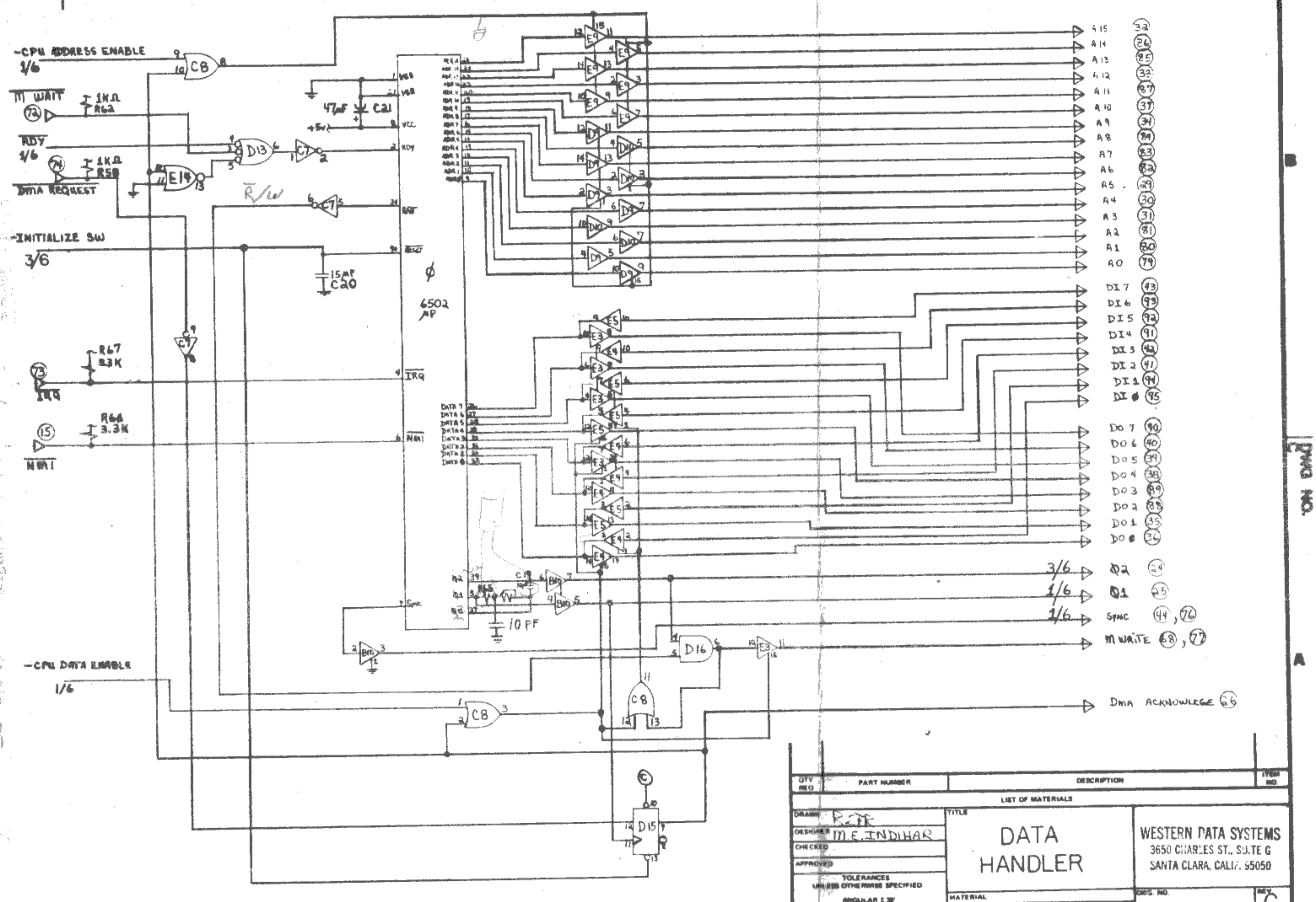
TITLE
DATA
HANDLER

MATERIAL
FINISH
FACILITIES JOB NO.

WESTERN DATA SYSTEMS
3650 CHARLES ST., SUITE G
SANTA CLARA, CALIF. 95050

DWG. NO.
REV. C
SCALE
SHT 6 OF 6

DWG NO.



QTY REQ	PART NUMBER	DESCRIPTION	ITEM NO.

LIST OF MATERIALS

DRAWN R.C.K.
DESIGNER M.E. INDIHAR
CHECKED
APPROVED

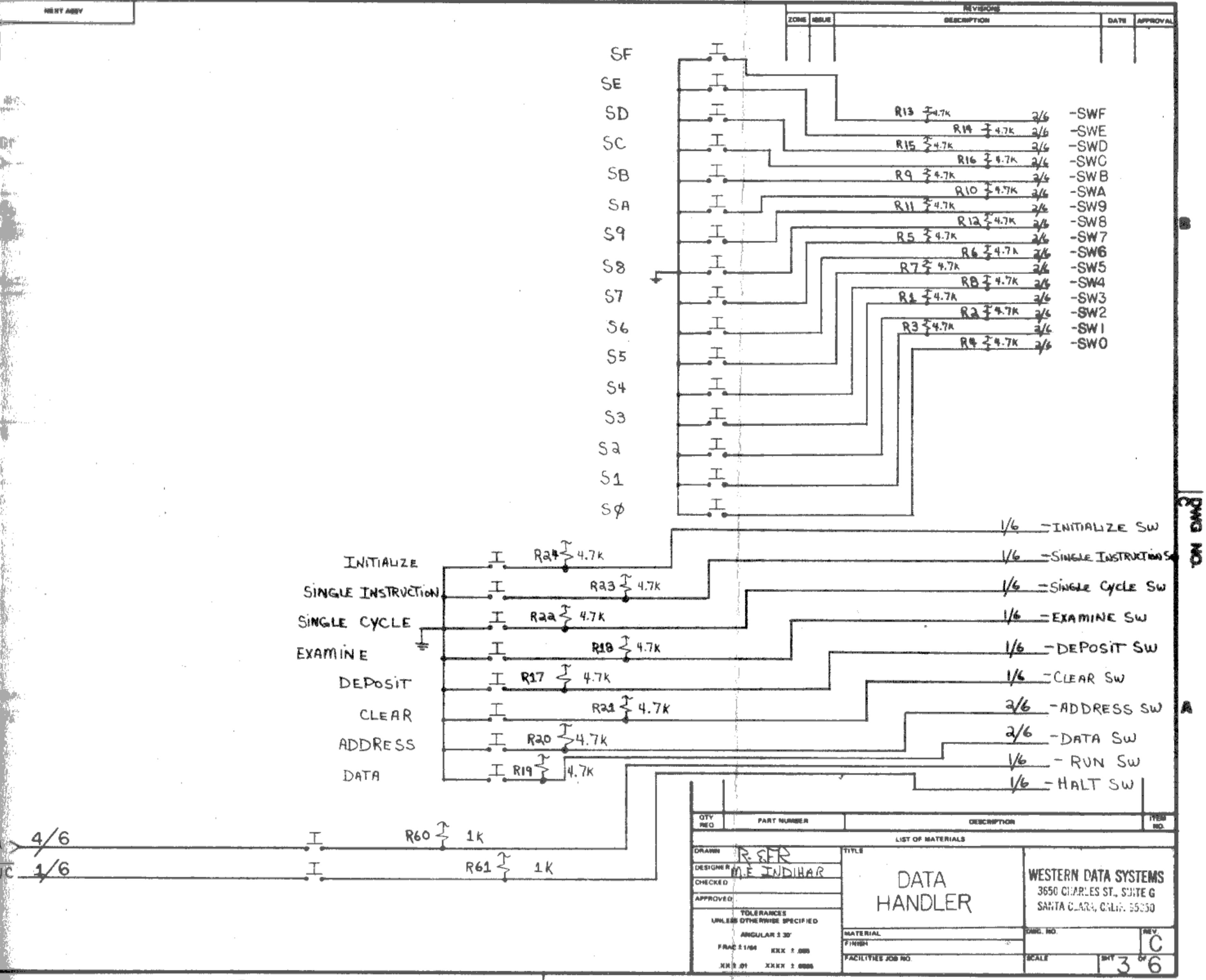
TOLERANCES
UNLESS OTHERWISE SPECIFIED
ANGULAR ± 30'
FRAC ± 1/64 XXX ± .005
XX ± .01 XXXX ± .0005

TITLE
DATA
HANDLER

MATERIAL
FINISH
FACILITIES JOB NO.

WESTERN DATA SYSTEMS
3650 CHARLES ST., SUITE G
SANTA CLARA, CALIF. 95050

DWG. NO.
REV C
SCALE
SHT 4 OF 6



QTY REQ	PART NUMBER	DESCRIPTION	ITEM NO.

LIST OF MATERIALS

DRAWN R. SFR

DESIGNER M.E. INDIHAR

CHECKED

APPROVED

TOLERANCES
UNLESS OTHERWISE SPECIFIED
ANGULAR ± 30'
FRAC ± 1/64 XXX ± .005
XX ± .01 XXXX ± .0005

TITLE

DATA HANDLER

MATERIAL

FINISH

FACILITIES JOB NO.

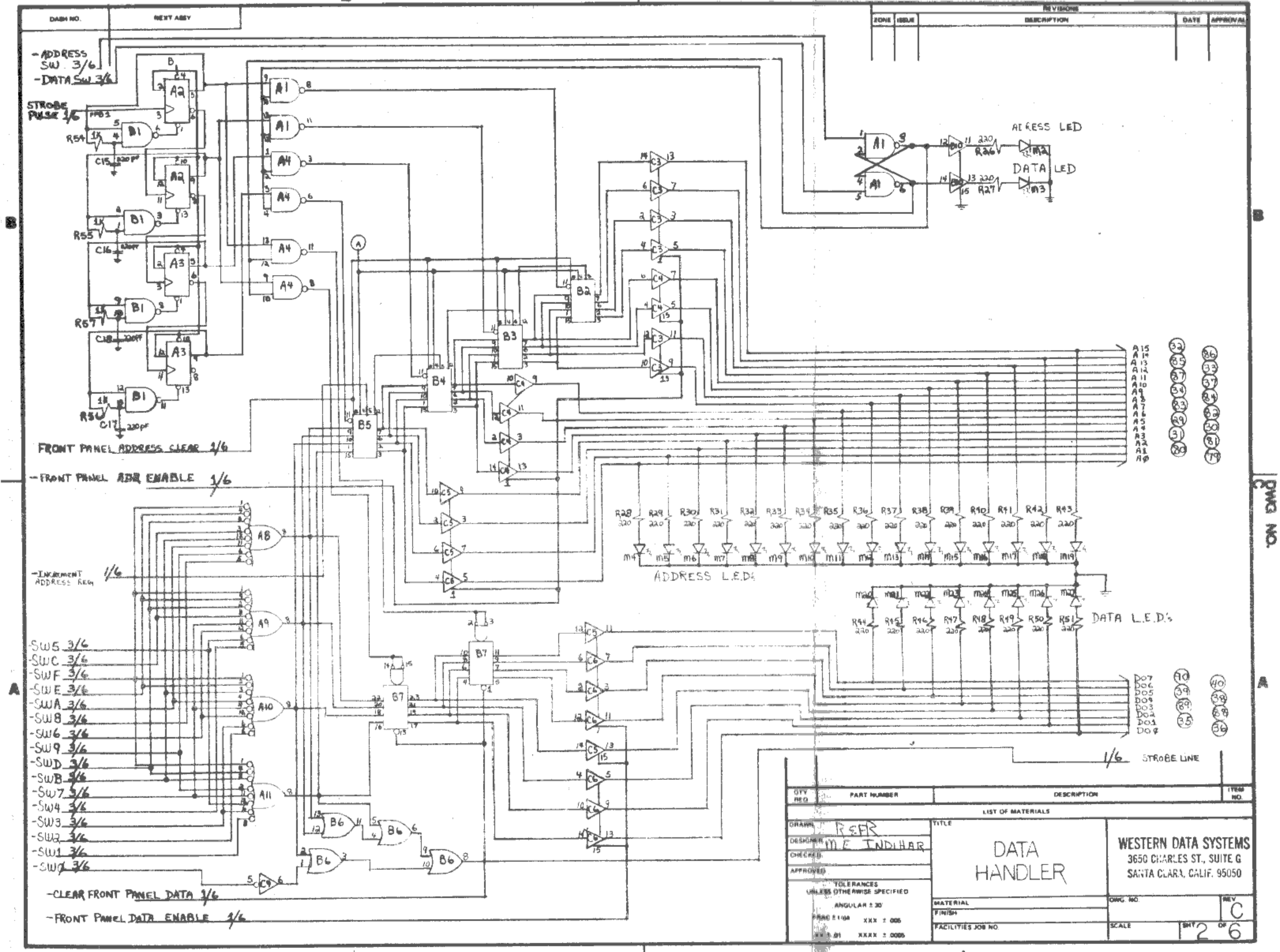
WESTERN DATA SYSTEMS
3650 CHARLES ST., SUITE G
SANTA CLARA, CALIF. 95050

DWG. NO.

REV. C

SCALE

SHT 3 OF 6



QTY REQ	PART NUMBER	DESCRIPTION	ITEM NO.

LIST OF MATERIALS

DRAWN R EFR
DESIGNER M E INDIHAR
CHECKED
APPROVED

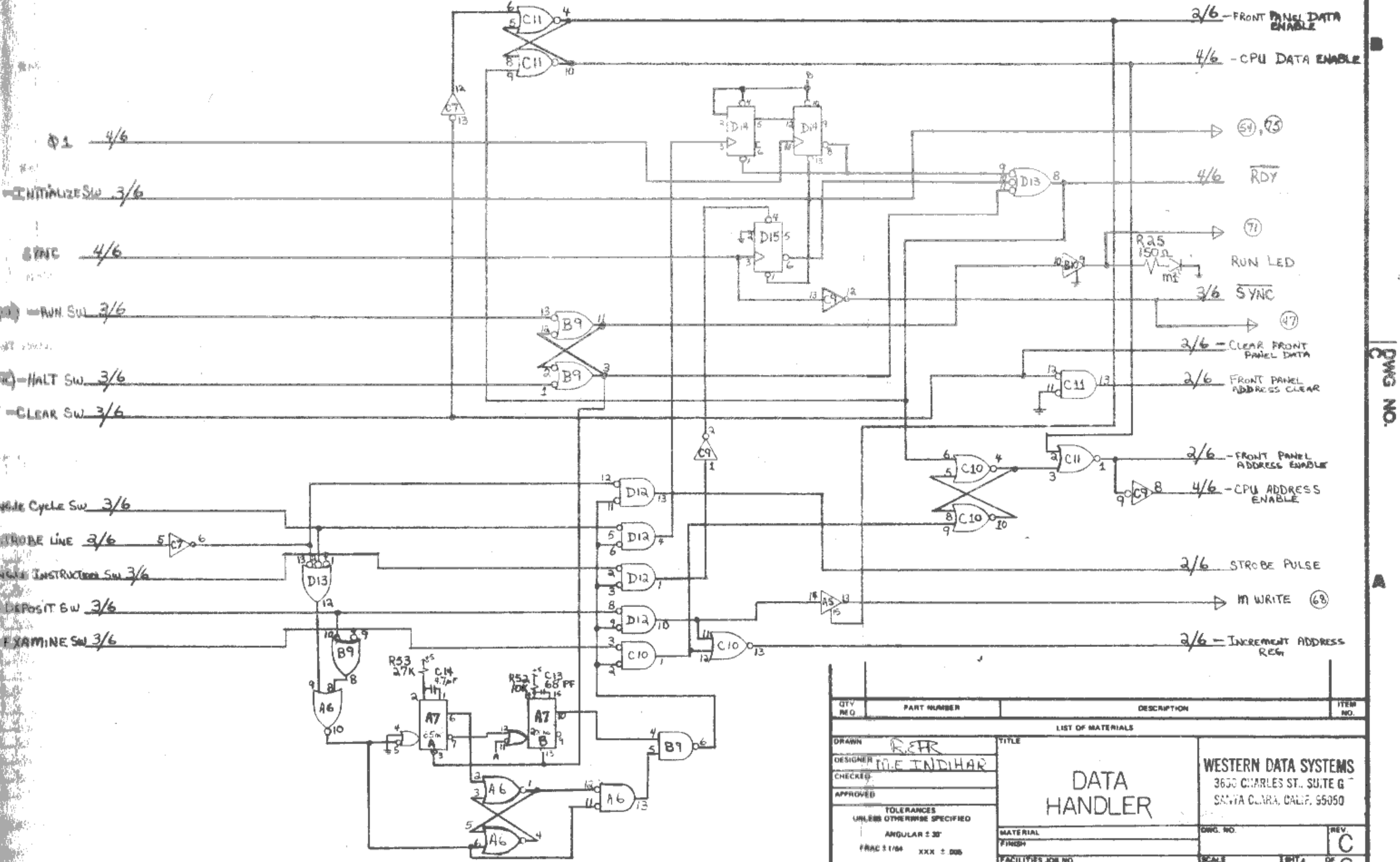
TOLERANCES
UNLESS OTHERWISE SPECIFIED
ANGULAR ± 30'
FRAC ± 1/64 XXX ± .005
XX ± .01 XXXX ± .0005

TITLE
DATA HANDLER

MATERIAL
FINISH
FACILITIES JOB NO.

WESTERN DATA SYSTEMS
3650 CHARLES ST., SUITE G
SANTA CLARA, CALIF. 95050

DWG. NO.
REV C
SCALE
SHT 2 OF 6



QTY REQ	PART NUMBER	DESCRIPTION	ITEM NO.

LIST OF MATERIALS

DRAWN R.E.R.
DESIGNER M.E. INDIHAR
CHECKED
APPROVED

TOLERANCES
UNLESS OTHERWISE SPECIFIED
ANGULAR ± 30'
FRAC ± 1/64 XXX ± .005
XX ± .01 XXXX ± .0005

TITLE
DATA HANDLER

MATERIAL
FINISH
FACILITIES JOB NO.

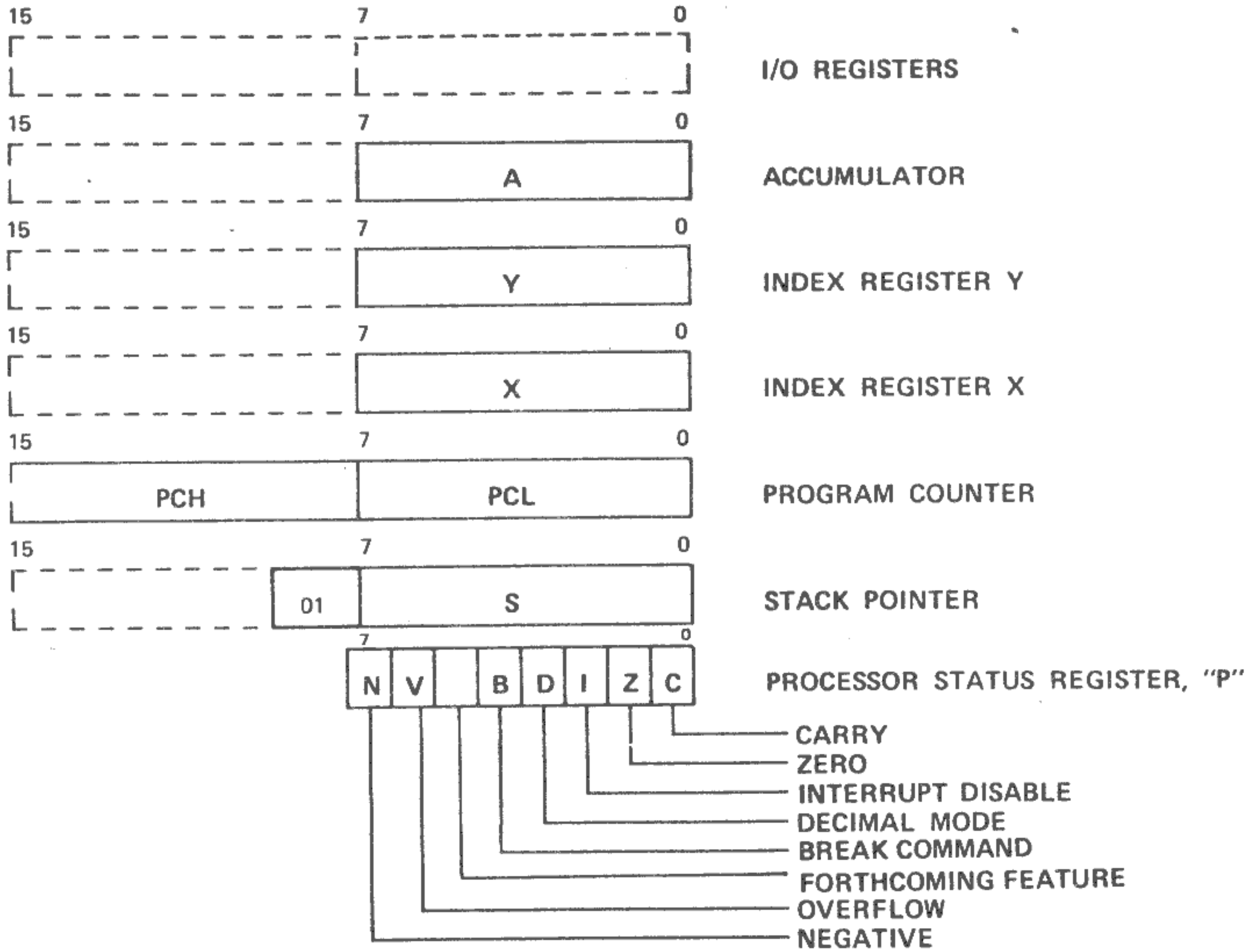
WESTERN DATA SYSTEMS
3650 CHARLES ST., SUITE G
SANTA CLARA, CALIF. 95050

DWG. NO.
REV. C
SCALE
SHT 1 OF 6

*CONSULT THE MOS TECHNOLOGY 65XX PROGRAMING MANUAL FOR USE OF THE FOLLOWING INSTRUCTION LIST AND FURTHER INFORMATION.

INSTRUCTION LIST
ALPHABETIC BY MNEMONIC
WITH OP CODES, EXECUTION CYCLES
AND MEMORY REQUIREMENTS

PROGRAMMING MODEL MCS650X



* Solid line indicates currently available features
Dashed line indicates forthcoming members of family

The following notation applies to this summary:

A	Accumulator
X, Y	Index Registers
M	Memory
P	Processor Status Register
S	Stack Register
✓	Change
_	No Change
+	Add
∧	Logical AND
-	Subtract
⊻	Logical Exclusive Or
↑	Transfer from Stack
↑	Transfer to Stack
→	Transfer to
←	Transfer to
V	Logical OR
PC	Program Counter
PCH	Program Counter High
PCL	Program Counter Low
OPER	OPERAND
#	IMMEDIATE ADDRESSING MODE

Note: At the top of each table is located in parentheses a reference number (Ref: XX) which directs the user to that Section in the MCS6500 Microcomputer Family Programming Manual in which the instruction is defined and discussed.

INSTRUCTION ADDRESSING MODES AND RELATED EXECUTION TIMES (in clock cycles)

	Accumulator	Immediate	Zero Page	Zero Page, X	Zero Page, Y	Absolute	Absolute, X	Absolute, Y	Implied	Relative	(Indirect, X)	(Indirect), Y	Absolute Indirect
ADC	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
AND	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
ASL	2	.	5	6	.	6	7	.	.	.	.	.	.
BCC	.	.	.	.	.	.	.	.	.	2**	.	.	.
BCS	.	.	.	.	.	.	.	.	.	2**	.	.	.
BEQ	.	.	.	.	.	.	.	.	.	2**	.	.	.
BIT	.	.	3	.	.	4	.	.	.	.	.	.	.
BMI	.	.	.	.	.	.	.	.	.	2**	.	.	.
BNE	.	.	.	.	.	.	.	.	.	2**	.	.	.
BPL	.	.	.	.	.	.	.	.	.	2**	.	.	.
BRK	.	.	.	.	.	.	.	.	.	.	.	.	.
BVC	.	.	.	.	.	.	.	.	.	2**	.	.	.
BVS	.	.	.	.	.	.	.	.	.	2**	.	.	.
CLC	.	.	.	.	.	.	.	.	2	.	.	.	.
CLD	.	.	.	.	.	.	.	.	2	.	.	.	.
CLI	.	.	.	.	.	.	.	.	2	.	.	.	.
CLV	.	.	.	.	.	.	.	.	2	.	.	.	.
CMP	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
CPX	.	2	3	.	.	4	.	.	.	.	.	.	.
CPY	.	2	3	.	.	4	.	.	.	.	.	.	.
DEC	.	.	5	6	.	6	7	.	.	.	.	.	.
DEX	.	.	.	.	.	.	.	.	2	.	.	.	.
DEY	.	.	.	.	.	.	.	.	2	.	.	.	.
EOR	.	2	3	4	.	4	4*	4*	.	.	6	5	.
INC	.	.	5	6	.	6	7	.	.	.	.	.	.
INX	.	.	.	.	.	.	.	.	2	.	.	.	.
INY	.	.	.	.	.	.	.	.	2	.	.	.	.
JMP	.	.	.	.	.	3	.	.	.	.	.	.	5
JSR	.	.	.	.	.	6	.	.	.	.	.	.	.
LDA	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
LDX	.	2	3	.	4	4	.	4*	.	.	.	.	.
LDY	.	2	3	4	.	4	4*	.	.	.	.	.	.
LSR	2	.	5	6	.	6	7	.	.	.	.	.	.
NOP	.	.	.	.	.	.	.	.	2	.	.	.	.
ORA	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
PHA	.	.	.	.	.	.	.	.	3	.	.	.	.
PHP	.	.	.	.	.	.	.	.	3	.	.	.	.
PLA	.	.	.	.	.	.	.	.	4	.	.	.	.
PLP	.	.	.	.	.	.	.	.	4	.	.	.	.
ROL	2	.	5	6	.	6	7	.	.	.	.	.	.
RTI	.	.	.	.	.	.	.	.	6	.	.	.	.
RTS	.	.	.	.	.	.	.	.	6	.	.	.	.
SBC	.	2	3	4	.	4	4*	4*	.	.	6	5*	.
SEC	.	.	.	.	.	.	.	.	2	.	.	.	.
SED	.	.	.	.	.	.	.	.	2	.	.	.	.
SEI	.	.	.	.	.	.	.	.	2	.	.	.	.
STA	.	.	3	4	.	4	5	5	.	.	6	6	.
STX*	.	.	3	.	4	4	.	.	.	.	.	.	.
STY**	.	.	3	4	.	4	.	.	.	.	.	.	.
TAX	.	.	.	.	.	.	.	.	2	.	.	.	.
TAY	.	.	.	.	.	.	.	.	2	.	.	.	.
TSX	.	.	.	.	.	.	.	.	2	.	.	.	.
TXA	.	.	.	.	.	.	.	.	2	.	.	.	.
TXS	.	.	.	.	.	.	.	.	2	.	.	.	.
TYA	.	.	.	.	.	.	.	.	2	.	.	.	.

* Add one cycle if indexing across page boundary

** Add one cycle if branch is taken, Add one additional if branching operation crosses page boundary

ADC *Add memory to accumulator with carry* ADC

Operation: A + M + C → A, C

N Ƶ C I D V
✓ ✓ ✓ — — ✓

(Ref: 2.2.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	ADC # Oper	69	2	2
Zero Page	ADC Oper	65	2	3
Zero Page, X	ADC Oper, X	75	2	4
Absolute	ADC Oper	6D	3	4
Absolute, X	ADC Oper, X	70	3	4*
Absolute, Y	ADC Oper, Y	79	3	4*
(Indirect, X)	ADC (Oper, X)	61	2	6
(Indirect), Y	ADC (Oper), Y	71	2	5*

* Add 1 if page boundary is crossed.

AND *"AND" memory with accumulator* AND

Logical AND to the accumulator

Operation: A ∧ M → A

N Ƶ C I D V
✓ ✓ — — — —

(Ref: 2.2.3.0)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	AND # Oper	29	2	2
Zero Page	AND Oper	25	2	3
Zero Page, X	AND Oper, X	35	2	4
Absolute	AND Oper	2D	3	4
Absolute, X	AND Oper, X	3D	3	4*
Absolute, Y	AND Oper, Y	39	3	4*
(Indirect, X)	AND (Oper, X)	21	2	6
(Indirect), Y	AND (Oper), Y	31	2	5

* Add 1 if page boundary is crossed.

ASL ASL *Shift Left One Bit (Memory or Accumulator)* ASL

Operation: C ← |7|6|5|4|3|2|1|Ø| ← Ø

N	Ƶ	C	I	D	V
✓	✓	✓	—	—	—

(Ref: 10.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Accumulator	ASL A	ØA	1	2
Zero Page	ASL Oper	Ø6	2	5
Zero Page, X	ASL Oper, X	16	2	6
Absolute	ASL Oper	ØE	3	6
Absolute, X	ASL Oper, X	1E	3	7

BCC BCC *Branch on Carry Clear* BCC

Operation: Branch on C = Ø

N	Ƶ	C	I	D	V
—	—	—	—	—	—

(Ref: 4.1.1.3)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BCC Oper	9Ø	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to different page.

BCS

BCS *Branch on carry set*

BCS

Operation: Branch on C = 1

N Ƶ C I D V

— — — — — —

(Ref: 4.1.1.4)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BCS Oper	BØ	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to next page.

BEQ

BEQ *Branch on result zero*

BEQ

Operation: Branch on Ƶ = 1

N Ƶ C I D V

— — — — — —

(Ref: 4.1.1.5)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BEQ Oper	FØ	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to next page.

BIT

BIT *Test bits in memory with accumulator*

BIT

Operation: $A \wedge M$, $M_7 \rightarrow N$, $M_6 \rightarrow V$

Bit 6 and 7 are transferred to the status register. If the result of $A \wedge M$ is zero then Z = 1, otherwise Z = Ø

N Z C I D V

M_7 ✓ — — — M_6

(Ref: 4.2.1.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Zero Page	BIT Oper	24	2	3
Absolute	BIT Oper	2C	3	4

BMI

BMI *Branch on result minus*

BMI

Operation: Branch on N = 1

N Z C I D V

— — — — — —

(Ref: 4.1.1.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BMI Oper	3Ø	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to different page.

BNE

BNE *Branch on result not zero*

BNE

Operation: Branch on Z = 0

N Ƶ C I D V

_ _ _ _ _ _

(Ref: 4.1.1.6)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BNE Oper	DØ	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to different page.

BPL

BPL *Branch on result plus*

BPL

Operation: Branch on N = Ø

N Ƶ C I D V

_ _ _ _ _ _

(Ref: 4.1.1.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BPL Oper	1Ø	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to different page.

BRK BRK *Force Break* BRK

Operation: Forced Interrupt PC ↓ P ↓

N	Z	C	I	D	V
—	—	—	1	—	—

(Ref: 9.11)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	BRK	ØØ	1	7

1. A BRK command cannot be masked by setting I.

BVC BVC *Branch on overflow clear* BVC

Operation: Branch on V = 0

N	Z	C	I	D	V
—	—	—	—	—	—

(Ref: 4.1.1.8)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BVC Oper	5Ø	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to different page.

BVS

BVS *Branch on overflow set*

BVS

Operation: Branch on V = 1

N Ƶ C I D V

— — — — — —

(Ref: 4.1.1.7)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Relative	BVS Oper	7Ø	2	2*

* Add 1 if branch occurs to same page.

* Add 2 if branch occurs to different page.

CLC

CLC *Clear carry flag*

CLC

Operation: Ø → C

N Ƶ C I D V

— — Ø — — —

(Ref: 3.0.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	CLC	18	1	2

CLD

CLD *Clear decimal mode*

CLD

Operation: Ø → D

N Ƶ C I D V
— — — — Ø —

(Ref: 3.3.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	CLD	D8	1	2

CLI

CLI *Clear interrupt disable bit*

CLI

Operation: Ø → I

N Ƶ C I D V
— — — Ø — —

(Ref: 3.2.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	CLI	58	1	2

CLV

CLV *Clear overflow flag*

CLV

Operation: Ø → V

N Ƶ C I D V

— — — — — Ø

(Ref: 3.6.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	CLV	B8	1	2

CMP

CMP *Compare memory and accumulator*

CMP

Operation: A - M

N Ƶ C I D V

✓ ✓ ✓ — — —

(Ref: 4.2.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	CMP #Oper	C9	2	2
Zero Page	CMP Oper	C5	2	3
Zero Page, X	CMP Oper, X	D5	2	4
Absolute	CMP Oper	CD	3	4
Absolute, X	CMP Oper, X	DD	3	4*
Absolute, Y	CMP Oper, Y	D9	3	4*
(Indirect, X)	CMP (Oper, X)	C1	2	6
(Indirect), Y	CMP (Oper), Y	D1	2	5*

* Add 1 if page boundary is crossed.

CPX

CPX *Compare Memory and Index X*

CPX

Operation: X - M

N	Ƶ	C	I	D	V
✓	✓	✓	—	—	—

(Ref: 7.8)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	CPX #Oper	EØ	2	2
Zero Page	CPX Oper	E4	2	3
Absolute	CPX Oper	EC	3	4

CPY

CPY *Compare memory and index Y*

CPY

Operation: Y - M

N	Ƶ	C	I	D	V
✓	✓	✓	—	—	—

(Ref: 7.9)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	CPY #Oper	CØ	2	2
Zero Page	CPY Oper	C4	2	3
Absolute	CPY Oper	CC	3	4

DEC DEC *Decrement memory by one* DEC

Operation: M - 1 → M

N Ƶ C I D V
✓ ✓ — — — —

(Ref: 10.7)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Zero Page	DEC Oper	C6	2	5
Zero Page, X	DEC Oper, X	D6	2	6
Absolute	DEC Oper	CE	3	6
Absolute, X	DEC Oper, X	DE	3	7

DEX DEX *Decrement index X by one* DEX

Operation: X - 1 → X

N Ƶ C I D V
✓ ✓ — — — —

(Ref: 7.6)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	DEX	CA	1	2

DEY DEY *Decrement index Y by one* DEY

Operation: Y - 1 → Y

N Z C I D V
✓ ✓ — — — —

(Ref: 7.7)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	DEY	88	1	2

EOR EOR *"Exclusive–Or" memory with accumulator* EOR

Operation: A ⊻ M → A

N Z C I D V
✓ ✓ — — — —

(Ref: 2.2.3.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	EOR #Oper	49	2	2
Zero Page	EOR Oper	45	2	3
Zero Page, X	EOR Oper, X	55	2	4
Absolute	EOR Oper	4D	3	4
Absolute, X	EOR Oper, X	5D	3	4*
Absolute, Y	EOR Oper, Y	59	3	4*
(Indirect, X)	EOR (Oper, X)	41	2	6
(Indirect),Y	EOR (Oper), Y	51	2	5*

* Add 1 if page boundary is crossed.

INC

INC *Increment memory by one*

INC

Operation: M + 1 → M

N	Ƶ	C	I	D	V
✓	✓	—	—	—	—

(Ref: 10.6)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Zero Page	INC Oper	E6	2	5
Zero Page, X	INC Oper, X	F6	2	6
Absolute	INC Oper	EE	3	6
Absolute, X	INC Oper, X	FE	3	7

INX

INX *Increment Index X by one*

INX

Operation: X + 1 → X

N	Ƶ	C	I	D	V
✓	✓	—	—	—	—

(Ref: 7.4)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	INX	E8	1	2

INY

INY *Increment Index Y by one*

INY

Operation: Y + 1 → Y

N	Z	C	I	D	V
✓	✓	—	—	—	—

(Ref: 7.5)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	INY	C8	1	2

JMP

JMP *Jump to new location*

JMP

Operation: (PC + 1)   PCL
           (PC + 2)   PCH

N	Z	C	I	D	V
—	—	—	—	—	—

(Ref: 4.0.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Absolute	JMP Oper	4C	3	3
Indirect	JMP (Oper)	6C	3	5

JSR

JSR *Jump to new location saving return address* **JSR**

Operation: PC + 2 ↓, (PC + 1) → PCL
(PC + 2) → PCH

N Z C I D V
— — — — — —

(Ref: 8.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Absolute	JSR Oper	20	3	6

LDA

LDA *Load accumulator with memory* **LDA**

Operation: M → A

N Z C I D V
✓ ✓ — — — —

(Ref: 2.1.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	LDA # Oper	A9	2	2
Zero Page	LDA Oper	A5	2	3
Zero Page, X	LDA Oper, X	B5	2	4
Absolute	LDA Oper	AD	3	4
Absolute, X	LDA Oper, X	BD	3	4*
Absolute, Y	LDA Oper, Y	B9	3	4*
(Indirect, X)	LDA (Oper, X)	A1	2	6
(Indirect), Y	LDA (Oper), Y	B1	2	5*

* Add 1 if page boundary is crossed.

LDX

LDX *Load index X with memory*

LDX

Operation: M → X

N Ƶ C I D V
✓ ✓ — — — —

(Ref: 7.0)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	LDX # Oper	A2	2	2
Zero Page	LDX Oper	A6	2	3
Zero Page, Y	LDX Oper, Y	B6	2	4
Absolute	LDX Oper	AE	3	4
Absolute, Y	LDX Oper, Y	BE	3	4*

* Add 1 when page boundary is crossed.

LDY

LDY *Load index Y with memory*

LDY

Operation: M → Y

N Ƶ C I D V
✓ ✓ — — — —

(Ref: 7.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	LDY #Oper	AØ	2	2
Zero Page	LDY Oper	A4	2	3
Zero Page, X	LDY Oper, X	B4	2	4
Absolute	LDY Oper	AC	3	4
Absolute, X	LDY Oper, X	BC	3	4*

* Add 1 when page boundary is crossed.

LSR

LSR *Shift right one bit (memory or accumulator)*

LSR

Operation: $\emptyset \rightarrow \boxed{7\,|\,6\,|\,5\,|\,4\,|\,3\,|\,2\,|\,1\,|\,0} \rightarrow C$

N	Ƶ	C	I	D	V
Ø	✓	✓	—	—	—

(Ref: 10.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Accumulator	LSR A	4A	1	2
Zero Page	LSR Oper	46	2	5
Zero Page, X	LSR Oper, X	56	2	6
Absolute	LSR Oper	4E	3	6
Absolute, X	LSR Oper, X	5E	3	7

NOP

NOP *No operation*

NOP

Operation: No Operation (2 cycles)

N	Ƶ	C	I	D	V
—	—	—	—	—	—

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	NOP	EA	1	2

ORA

ORA *"OR" memory with accumulator*

ORA

Operation: A V M → A

(Ref: 2.2.3.1)

N	Z	C	I	D	V
✓	✓	—	—	—	—

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	ORA #Oper	Ø9	2	2
Zero Page	ORA Oper	Ø5	2	3
Zero Page, X	ORA Oper, X	15	2	4
Absolute	ORA Oper	ØD	3	4
Absolute, X	ORA Oper, X	1D	3	4*
Absolute, Y	ORA Oper, Y	19	3	4*
(Indirect, X)	ORA (Oper, X)	Ø1	2	6
(Indirect), Y	ORA (Oper), Y	11	2	5

* Add 1 on page crossing

PHA

PHA *Push accumulator on stack*

PHA

Operation: A ↓

(Ref: 8.5)

N	Z	C	I	D	V
—	—	—	—	—	—

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	PHA	48	1	3

PHP

PHP *Push processor status on stack*

PHP

peration: P↓

N Ƶ C I D V

— — — — — —

(Ref: 8.11)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	PHP	Ø8	1	3

PLA

PLA *Pull accumulator from stack*

PLA

Operation: A ↑

N Ƶ C I D V

✓ ✓ — — — —

(Ref: 8.6)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	PLA	68	1	4

PLP

PLP *Pull processor status from stack*

PLP

Operation: P ↑

N Ƶ C I D V

From Stack

(Ref: 8.12)

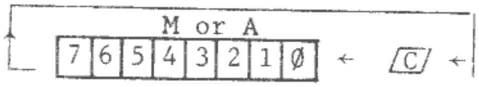
Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	PLP	28	1	4

ROL

ROL *Rotate one bit left (memory or accumulator)*

ROL

Operation:



N Ƶ C I D V

✓ ✓ ✓ — — —

(Ref: 10.3)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Accumulator	ROL A	2A	1	2
Zero Page	ROL Oper	26	2	5
Zero Page, X	ROL Oper, X	36	2	6
Absolute	ROL Oper	2E	3	6
Absolute, X	ROL Oper, X	3E	3	7

SBC

SBC *Subtract memory from accumulator with borrow*

SBC

Operation: $A - M - \overline{C} \to A$

Note: $\overline{C}$ = Borrow (Ref: 2.2.2)

N	Ƶ	C	I	D	V
✓	✓	✓	—	—	✓

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Immediate	SBC #Oper	E9	2	2
Zero Page	SBC Oper	E5	2	3
Zero Page, X	SBC Oper, X	F5	2	4
Absolute	SBC Oper	ED	3	4
Absolute, X	SBC Oper, X	FD	3	4*
Absolute, Y	SBC Oper, Y	F9	3	4*
(Indirect, X)	SBC (Oper, X)	E1	2	6
(Indirect), Y	SBC (Oper), Y	F1	2	5*

* Add 1 when page boundary is crossed.

SEC

SEC *Set carry flag*

SEC

Operation: 1 → C

(Ref: 3.0.1)

N	Ƶ	C	I	D	V
—	—	1	—	—	—

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	SEC	38	1	2

RTI

RTI *Return from interrupt*

RTI

Operation: P↑ PC↑

N Ƶ C I D V
From Stack

(Ref: 9.6)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	RTI	4Ø	1	6

RTS

RTS *Return from subroutine*

RTS

Operation: PC↑, PC + 1 → PC

N Ƶ C I D V
— — — — — —

(Ref: 8.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	RTS	6Ø	1	6

SED

SED *Set decimal mode*

SED

Operation: 1 → D

N	Z	C	I	D	V
—	—	—	—	1	—

(Ref: 3.3.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	SED	F8	1	2

SEI

SEI *Set interrupt disable status*

SEI

Operation: 1 → I

N	Z	C	I	D	V
—	—	—	1	—	—

(Ref: 3.2.1)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	SEI	78	1	2

STA — STA *Store accumulator in memory* — STA

Operation: A → M

N Ƶ C I D V
— — — — — —

(Ref: 2.1.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Zero Page	STA Oper	85	2	3
Zero Page, X	STA Oper, X	95	2	4
Absolute	STA Oper	8D	3	4
Absolute, X	STA Oper, X	9D	3	5
Absolute, Y	STA Oper, Y	99	3	5
(Indirect, X)	STA (Oper, X)	81	2	6
(Indirect), Y	STA (Oper), Y	91	2	6

STX — STX *Store index X in memory* — STX

Operation: X → M

N Ƶ C I D V
— — — — — —

(Ref: 7.2)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Zero Page	STX Oper	86	2	3
Zero Page, Y	STX Oper, Y	96	2	4
Absolute	STX Oper	8E	3	4

STY

STY Store index Y in memory

STY

Operation: Y → M

N Z C I D V
— — — — — —

(Ref: 7.3)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Zero Page	STY Oper	84	2	3
Zero Page, X	STY Oper, X	94	2	4
Absolute	STY Oper	8C	3	4

TAX

TAX *Transfer accumulator to index X*

TAX

Operation: A → X

N Z C I D V
✓ ✓ — — — —

(Ref: 7.11)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	TAX	AA	1	2

TAY

TAY *Transfer accumulator to index Y*

TAY

Operation: A → Y

N	Z	C	I	D	V
√	√	—	—	—	—

(Ref: 7.13)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	TAY	A8	1	2

TYA

TYA *Transfer index Y to accumulator*

TYA

Operation: Y → A

N	Z	C	I	D	V
√	√	—	—	—	—

(Ref: 7.14)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	TYA	98	1	2

TSX

TSX *Transfer stack pointer to index X* **TSX**

Operation: S → X

N	Ƶ	C	I	D	V
✓	✓	—	—	—	—

(Ref: 8.9)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	TSX	BA	1	2

TXA

TXA *Transfer index X to accumulator* **TXA**

Operation: X → A

N	Ƶ	C	I	D	V
✓	✓	—	—	—	—

(Ref: 7.12)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	TXA	8A	1	2

TXS

TXS *Transfer index X to stack pointer* **TXS**

Operation: X → S

N	Ƶ	C	I	D	V
—	—	—	—	—	—

(Ref: 8.8)

Addressing Mode	Assembly Language Form	OP CODE	No. Bytes	No. Cycles
Implied	TXS	9A	1	2

ØØ - BRK

Ø1 - ORA - (Indirect,X)

Ø2 - Future Expansion

Ø3 - Future Expansion

Ø4 - Future Expansion

Ø5 - ORA - Zero Page

Ø6 - ASL - Zero Page

Ø7 - Future Expansion

Ø8 - PHP

Ø9 - ORA - Immediate

ØA - ASL - Accumulator

ØB - Future Expansion

ØC - Future Expansion

ØD - ORA - Absolute

ØE - ASL - Absolute

ØF - Future Expansion

1Ø - BPL

11 - ORA - (Indirect),Y

12 - Future Expansion

13 - Future Expansion

14 - Future Expansion

15 - ORA - Zero Page,X

16 - ASL - Zero Page,X

17 - Future Expansion

18 - CLC

19 - ORA - Absolute,Y

1A - Future Expansion

1B - Future Expansion

1C - Future Expansion

1D - ORA - Absolute,X

1E - ASL - Absolute,X

1F - Future Expansion

2Ø - JSR

21 - AND - (Indirect,X)

22 - Future Expansion

23 - Future Expansion

24 - BIT - Zero Page

25 - AND - Zero Page

26 - ROL - Zero Page

27 - Future Expansion

28 - PLP

29 - AND - Immediate

2A - ROL - Accumulator

2B - Future Expansion

2C - BIT - Absolute

2D - AND - Absolute

2E - ROL - Absolute

2F - Future Expansion

3Ø - EMI

31 - AND - (Indirect),Y

32 - Future Expansion

33 - Future Expansion

34 - Future Expansion

35 - AND - Zero Page,X

36 - ROL - Zero Page,X

37 - Future Expansion

38 - SEC

39 - AND - Absolute,Y

3A - Future Expansion

3B - Future Expansion

3C - Future Expansion

3D - AND - Absolute,X

3E - ROL - Absolute,X

3F - Future Expansion

4Ø - RTI
41 - EOR - (Indirect,X)
42 - Future Expansion
43 - Future Expansion
44 - Future Expansion
45 - EOR - Zero Page
46 - LSR - Zero Page
47 - Future Expansion
48 - PHA
49 - EOR - Immediate
4A - LSR - Accumulator
4B - Future Expansion
4C - JMP - Absolute
4D - EOR - Absolute
4E - LSR - Absolute
4F - Future Expansion
5Ø - BVC
51 - EOR - (Indirect),Y
52 - Future Expansion
53 - Future Expansion
54 - Future Expansion
55 - EOR - Zero Page,X
56 - LSR - Zero Page,X
57 - Future Expansion
58 - CLI
59 - EOR - Absolute,Y
5A - Future Expansion
5B - Future Expansion
5C - Future Expansion
5D - EOR - Absolute,X
5E - LSR - Absolute,X
5F - Future Expansion

6Ø - RTS
61 - ADC - (Indirect,X)
62 - Future Expansion
63 - Future Expansion
64 - Future Expansion
65 - ADC - Zero Page
66 - Future Expansion
67 - Future Expansion
68 - PLA
69 - ADC - Immediate
6A - Future Expansion
6B - Future Expansion
6C - JMP - Indirect
6D - ADC - Absolute
6E - Future Expansion
6F - Future Expansion
7Ø - BVS
71 - ADC - (Indirect),Y
72 - Future Expansion
73 - Future Expansion
74 - Future Expansion
75 - ADC - Zero Page,X
76 - Future Expansion
77 - Future Expansion
78 - SEI
79 - ADC - Absolute,Y
7A - Future Expansion
7B - Future Expansion
7C - Future Expansion
7D - ADC - Absolute,X
7E - Future Expansion
7F - Future Expansion

8Ø - Future Expansion

81 - STA - (Indirect,X)

82 - Future Expansion

83 - Future Expansion

84 - STY - Zero Page

85 - STA - Zero Page

86 - STX - Zero Page

87 - Future Expansion

88 - DEY

89 - Future Expansion

8A - TXA

8B - Future Expansion

8C - STY - Absolute

8D - STA - Absolute

8E - STX - Absolute

8F - Future Expansion

9Ø - BCC

91 - STA - (Indirect),Y

92 - Future Expansion

93 - Future Expansion

94 - STY - Zero Page,X

95 - STA - Zero Page,X

96 - STX - Zero Page,Y

97 - Future Expansion

98 - TYA

99 - STA - Absolute,Y

9A - TXS

9B - Future Expansion

9C - Future Expansion

9D - STA - Absolute,X

9E - Future Expansion

9F - Future Expansion

AØ - LDY - Immediate

A1 - LDA - (Indirect,X)

A2 - LDX - Immediate

A3 - Future Expansion

A4 - LDY - Zero Page

A5 - LDA - Zero Page

A6 - LDX - Zero Page

A7 - Future Expansion

A8 - TAY

A9 - LDA - Immediate

AA - TAX

AB - Future Expansion

AC - LDY - Absolute

AD - LDA - Absolute

AE - LDX - Absolute

AF - Future Expansion

BØ - BCS

B1 - LDA - (Indirect),Y

B2 - Future Expansion

B3 - Future Expansion

B4 - LDY - Zero Page,X

B5 - LDA - Zero Page,X

B6 - LDX - Zero Page,Y

B7 - Future Expansion

B8 - CLV

B9 - LDA - Absolute,Y

BA - TSX

BB - Future Expansion

BC - LDY - Absolute,X

BD - LDA - Absolute,X

BE - LDX - Absolute,Y

BF - Future Expansion

CØ - CPY - Immediate

C1 - CMP - (Indirect,X)

C2 - Future Expansion

C3 - Future Expansion

C4 - CPY - Zero Page

C5 - CMP - Zero Page

C6 - DEC - Zero Page

C7 - Future Expansion

C8 - INY

C9 - CMP - Immediate

CA - DEX

CB - Future Expansion

CC - CPY - Absolute

CD - CMP - Absolute

CE - DEC - Absolute

CF - Future Expansion

DØ - BNE

D1 - CMP - (Indirect),Y

D2 - Future Expansion

D3 - Future Expansion

D4 - Future Expansion

D5 - CMP - Zero Page,X

D6 - DEC - Zero Page,X

D7 - Future Expansion

D8 - CLD

D9 - CMP - Absolute,Y

DA - Future Expansion

DB - Future Expansion

DC - Future Expansion

DD - CMP - Absolute,X

DE - DEC - Absolute,X

DF - Future Expansion

EØ - CPX - Immediate

E1 - SBC - (Indirect,X)

E2 - Future Expansion

E3 - Future Expansion

E4 - CPX - Zero Page

E5 - SBC - Zero Page

E6 - INC - Zero Page

E7 - Future Expansion

E8 - INX

E9 - SBC - Immediate

EA - NOP

EB - Future Expansion

EC - CPX - Absolute

ED - SBC - Absolute

EE - INC - Absolute

EF - Future Expansion

FØ - BEQ

F1 - SBC - (Indirect),Y

F2 - Future Expansion

F3 - Future Expansion

F4 - Future Expansion

F5 - SBC - Zero Page,X

F6 - INC - Zero Page,X

F7 - Future Expansion

F8 - SED

F9 - SBC - Absolute,Y

FA - Future Expansion

FB - Future Expansion

FC - Future Expansion

FD - SBC - Absolute,X

FE - INC - Absolute,X

FF - Future Expansion

PROGRAMS

The following is a list of program software that can be used on the Data Handler. By adding, deleting or combining portions of these together, variations or entirely new programs can be conceived.

Western Data Systems wishes to be instrumental in the organizing and distributing of these, and welcomes any software donations from users that could be placed in future manuals for use by others.

Submit any software for insertion into the owners manual to:

MANUAL SOFTWARE
WESTERN DATA SYSTEMS
3650 CHARLES ST. SUITE G
SANTA CLARA CALIF.
95050

8-BIT MULTIPLY

INITIAL LOAD: FD00 Multiplicand; FD01 Multiplier; FD02 Ø0;

```
LOCATION   DATA   LABEL   MNEMONIC             COMMENTS

  FFFC      00
  FFFD      FC
  FC0Ø      A2              LDX       LOAD X REG.
  FC01      08                        WITH 8
  FC02      D8              CLD       CLEAR DECIMAL MODE
  FC03      18              CLC       CLEAR CARRY
  FC04      ØE     LOOP     ASL       SHIFT PARTIAL PRODUCT LEFT
  FC05      02                        (IT'S IN FD02
  FC06      FD
  FC07      0E
  FC08      01
  FC09      FD
  FC0A      B0              BCS       BRANCH ON CARRY SET
  FC0B      06                        TO MULT
  FC0C      CA     CONT     DEX       DECREMENT X REG
  FC0D      D0              BNE       BRANCH IF NOT= ZERO
  FC0E      F5                        BACK TO LOOP
  FC0F      4C              JMP       JUMP IF DONE TO STOR
  FC10      1F
  FC11      FC
  FC12      18     MULT     CLC       CLEAR CARRY
  FC13      AD              LDA       LOAD ACCUMULATOR WITH
  FC14      02                        PARTIAL PRODUCT(IT'S IN FD02)
  FC15      FD
  FC16      6D              ADC       ADD ACCUMULATOR WITH CARRY
  FC17      00                        AND MULTIPLICAND(IT'S IN FD00)
  FC18      FD
  FC19      8D              STA       STORE RESULT BACK IN PARTIAL
  FC1A      02                        PRODUCT AT FD02
  FC1B      FD
  FC1C      4C              JMP       JUMP BACK CONT AT FC0C
  FC1D      0C
  FC1E      FC
  FC1F      4C     END      JMP       JUMP ON SELF
  FC20      1F
  FC21      FC
```

ANSWER WILL BE: in FD02

0 TO 99 NUMBER GUESSING GAME

The 0 to 99 number guessing game is a simple game program that takes a few minutes to load and play. The Data Handler randomly generates a number which the player tries to guess by loading his guess number (0-99) into memory location FC10. The average number of tries to correctly guess the randomly generated number is six. Try it and see if you can beat the average number of guesses.

```
FFFC-00
FFFD-FC

FC00-F8          SED
  01-A9,0C       LDA 1MM  ]
  03-8D,FC,FF    STA ABS  ]--STORE NEW INT LOC.
  06-69,01       ADC 1MM  ]
  08-AA          TAX      ]
  09-4C,06,FC    JMP      ]--RANDOM NUMBER GEN.
  0C-B8          CLV         CLEAR OV FLAG
  0D-18          CLC         CLEAR CARRY FLAG
  0E-8A          TXA
  0F-E9,GUESS    SBC 1MM     CALCULATE DIFFERENCE
  11-F0,2D       BEQ         BRANCH TO CORRECT
  13-10,0B       BPL         BRANCH TO LOW
  15-30,69       BM1         BRANCH TO HIGH

FC20-EA          NOP]
  21-4C,20,FC    NOP]--------LOW GUESS

FC40-A2,00       LDA 1MM]
  42-8D,FC,FF    STA ABS]----CORRECT GUESS
  45-4C,45,FC    JMP

FC80-EA          NOP]
  81-4C,80,FC    JMP]--------HIGH GUESS
```

(Con't)

0 TO 99 NUMBER GUESSING GAME

Load the preceding program data into the associated memory locations (using the procedure described in the Data Handler manual) press INT and RN. The Data Handler will enter a portion of the program which generates a random decimal number. Press HT (halt the Data Handler), press CL (to switch control to the front panel), press ADR, now enter the memory address location (FC10) in which we will place our guess number.

With FC10 on the address lights press DA, now enter your guess onto the data lights (the program will accept only decimal number between 0 and 99 for the guess). With your guess number on the data lights press DP. Press INT and RN, or to observe the Data Handler calculating the results of your guess press INT, and then the Data Handler may be walked through the program (by repeatedly pressing SC or SI).

The Data Handler will ultimately end up in one of three loops, the low guess (your guess was low) is most easily seen with the address light A5 on continuously, the high guess (your guess was high) is most easily seen with the address light A7 on continuously. The correct guess (your guess was 100% absolutely correct) is most easily seen with the address light A6 on continuously.

If you correctly guessed the computers random number in 5 or less tries you have done it above the average. If not try again.

To restart the game press INT. This vectors the Data Handler back into the random number generator to await your first guess.

MEMORY 1 TEST PROGRAM

This program is designed to test individual 256 word blocks of memory, this is written to reside in the top block of memory (FF00$_{16}$-FFFF$_{16}$) which for obvious reasons should not be tested without relocation of the memory 1 test program.

```
FFFC-00
FFFD-FF

FF00-AD,10,FF          LDA ABS.        PROGRAM
  03-8D,16,FF          STA ABS.        FORMATTING
  06-AD,0F,FF          LDA ABS.
  09-8D,15,FF          STA ABS.
  0C-A9,XX             LDA 1MM.        TESTING
  0E-8D,XX,BLOCK       STA ABS.        LOOP
  11-EA                NOP
  12-EA                NOP
  13-EA                NOP
  14-CD,XX,XX          CMP ABS.
  17-D0,OE             BNE             TEST FOR EQUAL
  19-EE,0D,FF          INC ABS.        INC. DATA
  1C-D0,EE             BNE             TEST OVER FLOW
  1E-EE,0F,FF          INC ABS.        INC ADDRESS
  21-EE,15,FF          INC ABS.
  24-4C,0C,FF          JMP ABS.
  27-EA                NOP
  28-4C,27,FF          JMP ABS.        BAD LOOP
```

Load the memory block address to be tested into the memory location FF10$_{16}$. (EXAMPLE: FC$_{16}$ will test FC00 to FCFF). Press INT then press RN. Under normal running conditions A5,A6,and A7 will slowly increment off and on. If a data error is encountered in the test, the Data Handler will enter the bad loop which is easily seen as A6, and A7 continuously off.

For the following data examine these locations.

FF0D-CORRECT DATA
FF0F-LO ADR. OF BAD DATA
FF10-HIGH ADR. OF BAD DATA

TO EXECUTE THE ABOVE PROGRAM PRESS INT AND THEN PRESS RN.

X=DON'T CARE.