

INSTALLATION, MAINTENANCE

AND

THEORY OF OPERATION

KENBAK-1 COMPUTER

KENBAK CORPORATION
P.O. BOX 49324
LOS ANGELES, CALIFORNIA 90049

INSTALLATION & MAINTENANCE

Your KENBAK-1 computer is a quality instrument which is built with modern proven components. With reasonable care you should expect months or years of operation without significant failures.

WARRANTY

The KENBAK-1 computer is warranted to be free from defects caused by materials, workmanship, and construction for a period of one year from the date of shipment. The liability of Kenbak Corporation under this warranty is limited to replacing or repairing any computer returned by the buyer during such period, provided:

1. Buyer promptly notifies Kenbak Corporation in writing requesting authorization to return the computer as per our warranty policy. Letters should itemize complaints.
2. The defective computer is returned to Kenbak Corporation, transportation charges prepaid.
3. The examination by Kenbak Corporation shall disclose to its satisfaction that defects have not been caused by misuse, neglect, accident, or improper treatment.

Under no condition shall Kenbak Corporation be liable for collateral or consequential damages. This warranty is in lieu of all other warranties expressed or implied.

Please note that the warranty requires approval from the factory to return the unit. This requirement is imposed because most apparent failures are not failures of the equipment. In 90% or more of the cases where it is first thought the computer is not working correctly, it will turn out that the user is in error. The computer is a piece of sophisticated electronic equipment, but it is usually more reliable than our understanding of it.

If you should have occasion to write about a failure, try to be very specific and to detail it exactly. Try to pinpoint the instruction or the feature that does not appear to be working correctly. Give the register and memory contents which are involved.

Write to: KENBAK CORPORATION
P.O. Box 49324
Los Angeles, California 90049

Should it ever be necessary to return the unit to the factory for repairs, the original shipping carton is the best way of protecting it during shipment. If possible, try to save the carton and the packing materials.

Keep a record of your serial number in a safe place. Should the unit be stolen, a record of the serial number can be valuable in proving ownership. In addition, the factory should be notified and they may be able to assist you. The serial number may be found on the name plate on the rear of the unit.

RECEIVING INSPECTION

When you receive your Kenbak-1 computer, inspect it for physical damage. Check the external surfaces for dents and unusual marks. Pay particular attention to those parts which protrude from the surfaces such as the switches and the fuse holder. Check for physical security of the external parts of the cabinet.

GROUNDING ADAPTOR

The Kenbak-1 computer is supplied with a 3 prong power connector. The purposes of the third prong, which provides a known ground path, are two-fold. The case is then grounded and the user is provided with an added measure of safety. Secondly, grounding of the case through this third wire gives a return path for static electricity discharges such as those which may be generated by the user walking across a carpet. This keeps the discharge from the circuits of the computer where it may cause a temporary malfunction.

IF YOUR ELECTRICAL SERVICE OUTLET IS NOT A THREE PRONG OUTLET, USE THE ADAPTOR WHICH IS SUPPLIED WITH THE COMPUTER.

To use the adaptor, the flexible wire with the hook on the end is to be connected to the screw which retains the cover plate on the electrical outlet. Using a screwdriver, back this screw out just enough to slip the hook under it. Tighten the screw. Plug the adaptor in the service outlet.

If you expect to use the computer in different locations, you will find it convenient to install an adaptor at each location which requires it.

The Kenbak-1 computer is designed for 60 cycle 115 volt circuits. This is the standard frequency and voltage almost everywhere in the U.S.A.

INITIAL CHECKOUT OF COMPUTER

The tests described in this section do not constitute a complete test of the computer. However, if these tests are concluded satisfactorily, there is a high probability that the computer is in a satisfactory condition.

First, turn the Power switch off. Connect the power cord to electrical service using the adaptor if necessary.

Turn power on. Probably some lights will come on but they may not. You should be able to hear the fan. If the fan is not running and no lights are on, the fuse has probably blown. Turn power off, disconnect the power cord, and examine the fuse (see the section on fuse replacement). If it is blown, then do not proceed any further. Contact the factory.

We now assume the fan is running with perhaps some lights on. Then make the following tests:

1. Push Start and hold it. The Run light should be on. If the Run light does not go out when Start is released, then push Stop.
2. Push the Address Display switch. The Address light should come on.
3. Push the Clear switch. The Input light should come on.
4. Push the Memory Read switch. The Memory light should come on.
5. Push the Clear switch again. Push the switch to the farthest left. The light above it should come on

and stay on. Push the switch adjacent to this. The light above this switch should turn on and stay on. Repeat this for all eight switches to the left of Clear. All eight lights to the left of the Input light should be on. Push Clear. All of the eight lights should go out.

Additional tests can be performed by trying Exercises 1 and 2 in the Laboratory Exercises Manual. Also, on a more advanced level, the section in the Programming Reference Manual on Console operations can be used.

The tests may indicate that a lamp is burned out. If so, see the later section on how to replace a lamp.

FUSE REPLACEMENT

Looking at the rear of the computer, the fuse holder is just above the power cord. To examine or replace a fuse, turn power off and disconnect the power cord. Turning the cap of the fuse holder counter-clockwise will release it. The fuse should come out with the cap.

Not all fuses are the same. They have different voltage and current ratings and they respond at different rates to overload conditions. Use only the same type of fuse as recommended. Fuses from the

following two manufacturers are recommended:

Littlefuse 3AG 3/4 ampere

Bussman AGC 3/4 ampere

Spare fuses have been included.

LAMP REPLACEMENT

Lamps may be replaced from the front panel. Long-nose or needle-nose pliers may be required. To replace a lamp, turn power off. Remove the lens by unscrewing it. If you find it too small to grasp securely with your fingers, use the pliers but use a very light touch so as not to damage the lens.

After removing the lens, the lamp bulb may be seen. It is removed by pulling it straight out. (It may be necessary to make a hook in a small piece of wire to pull the bulb out.) The new bulb is inserted by pushing it in. Before inserting it, straighten the leads and spread them to correspond to the fingers inside the holder. (It may be necessary to cut off the excess lead length.) Before inserting the bulb, rotate it so that the leads are aligned with the fingers in the holder.

The recommended lampbulb is a Type 680 (T-1 Wire Terminal) which is available directly from a number of sources or from Kenbak. A few spare bulbs are included with the computer. The type 680 bulb is rated at 100,000

hours life, but this rating is statistical and a few may fail early in their life.

FAN

A small fan draws air from outside the case through the perforations at the right of the rear panel. Inspect this region periodically and wipe it off if lint and dust are accumulating there. The air drawn into the cabinet is circulated through the cabinet and exits through the perforations at the center of the rear panel. In this general area, there should be some warming of the cabinet as the air comes in contact with the metal.

The fan motor is lubricated with a high-grade time-tested oil before leaving the factory. It should not require relubrication for the life of the computer. If it should be necessary to lubricate the motor, a premium grade of detergent or non-detergent SAE 20 or SAE 30 oil is satisfactory. Avoid over-lubrication; one or two drops will be enough.

There are two points to be lubricated. Looking from the fan toward the motor, there is a notch just left of the axis of the motor. Felt material, to hold the oil, may be observed through the notch. There is also a similar notch for the bearing at the opposite end of the motor.

OPENING THE CASE

To open the case proceed as follows:

1. Disconnect the power cord.
2. Loosen the three small screws in the bottom of each of the aluminum trim pieces on the extreme sides of the cabinet and remove the trim pieces.
3. Remove the three top screws which secure the front panel to the case. Leave the bottom three screws.
4. Remove the three top screws which secure the rear panel to the case. Leave the bottom three screws.
5. Lift the top half of the case up.

This is as far as disassembly would normally proceed in the field. For assembly reverse these steps. Note that the center screws in the aluminum trim pieces are designed to go through holes in the bottom and top cover pieces.

THEORY OF OPERATION

INTRODUCTION

The reader is assumed to be acquainted with general digital theory for computers. It is also assumed that the reader is familiar with the programming material. He need not be a programmer but he should understand what the individual instructions accomplish.

The broad concepts and the detailed schematics of the KENBAK-1 computer are given. The material is divided into these sections:

Glossary of Signals

Block Diagram

State Diagram

Definition of Circuit Elements

Kenbak-1 Computer Circuit Schematics

The Glossary of Signals is presented first to emphasize that it does exist and that it can be a useful working tool in studying other aspects of the presentation. However, it is a reference section and not a fundamental mode for presenting ideas. The local reference in the Glossary is the schematic page number where that signal originates.

GLOSSARY

<u>Name</u>	<u>Local Reference</u>	<u>Definition or Comment</u>
A	(20)	One of the two inputs to the adder/subtractor, see also B.
A7	(03)	The most significant address bit which selects one of the two delay lines which constitute the memory.
B	(20)	One of the two inputs to the adder/subtractor. During subtraction B is subtracted from A.
BD	(21)	A flipflop used to detect whether any bits in the A, B, or X registers are 1. Used in evaluating JC.
BM	(23)	An AND of the conditions <ol style="list-style-type: none"> 1. Skip or Set instructions, 2. The bit time (BT) of the selected bit, and 3. State SL. In the state diagram, BM is used in broad sense of 1 to mean Bit Manipulation instructions.
BT	(18)	The bit manipulation instructions designate a specific bit time. BT is true during that time.
BU	(05)	The eight data input switches are scanned by T0 through T7 and 0Red together to produce the serial signal BU.
C	(19)	The carry flipflop in the adder.
CL	(04)	True when the Clear pushbutton is depressed.
CM	(02)	The comparison between the memory address register and the desired address is true. CM is only valid during T7.
\overline{CP}	(01)	Major system clock pulse. A bit time starts with the negative going transition of \overline{CP} .
CR	(19)	Carry to the next bit before being delayed by C.

GLOSSARY

CT	(25)	Output of multivibrator.
CY	(21)	A flipflop which holds the carry from the previous byte.
$\overline{C7}$	(01)	Clock pulse that occurs only during T7. Has same phase as \overline{CP} .
\overline{DA}	(04)	Display Address (from pushbutton of that name). Goes to ground when button is pushed.
\overline{DD}	(04)	Display Data (from Read Memory pushbutton) after being cleaned up. Goes to ground when button is pushed.
DEX		Index address mode.
DLO	(24)	Output of memory delay line 0.
DL1	(24)	Output of memory delay line 1.
\overline{EA}	(04)	Enter Address (from Set Address pushbutton) after being cleaned up. Goes to ground when button is pushed.
ED	(22)	Automatic processing should End after the current instruction is finished.
\overline{EN}	(04)	Enter Data (from Store Memory pushbutton) after being cleaned up. Goes to ground when button is pushed.
F1/F2	(01)	Inputs to delay line clock drivers.
\overline{GO}	(04)	Signal from Start Pushbutton after being cleaned up. Goes to ground when button is pushed.
\overline{HT}	(04)	Signal from Stop Pushbutton. Goes to ground when button is pushed.
IMMED		Immediate or constant address mode.
IN	(04)	True during byte time when location 377 (Input) is being read from memory.
IND		Indirect address mode.

GLOSSARY

I0/I7	(14)	Generally the I register is the instruction register. I7 is the most significant bit and I0 is the least significant bit. As an instruction register, I holds the first byte of an instruction.
J		Any Jump instruction. See also JD, JI, JM, JP.
JC	(21)	Jump Conditions are true; the Jump should be made.
JD		A Jump instruction with direct addressing (not indirect).
JDLO	(24)	Input to delay line 0.
JDL1	(24)	Input to delay line 1.
JI		A Jump instruction with indirect addressing.
JM		A Jump and Mark instruction.
JP		A straight Jump instruction (no Mark).
JS0/JS4	(10)	Inputs to 5 state flipflops.
\overline{KP}	(22)	This signal is ground if a skip is required in the bit test instructions.
\overline{KS}	(06)	Shift control to K register. At ground, \overline{KS} indicates a shift.
K0/K7	(07)	The K Register controls the data lamps. K7 is the most significant bit.
LC	(02)	Carry flipflop for memory address (L) register. When LC is a 1 at T7 the next address is 177 (and 377).
\overline{LS}	(18)	Shift control signal for left shifting.
L0/L7	(02)	Memory address register. Read serially via L0.

GLOSSARY

MEM		Memory address mode.
MR	(03)	Memory read. Is the output of DLO or DL1 as selected by A7.
OF	(19)	Overflow as detected by the adder.
OV	(21)	Overflow delayed for one byte time.
PF	(22)	Four is to be added to the P register (Plus Four).
PO	(22)	One is to be added to the P register (Plus One).
PT	(22)	Two is to be added to the P register (Plus Two).
QB	(09)	State to wait for Start button to be released.
QC	(09)	Idle state for manual operations.
QD	(09)	State to seek address comparison for manual operations.
QE	(09)	State to execute manual operations.
QF	(09)	State to wait for control buttons to be released.
\bar{R}	(03)	Desired address in logic complement form.
RR	(18)	Control signal for right shifts and rotates.
SA	(09)	State to seek P register.
SB	(09)	State to read and update P register.
SC	(09)	State to seek instruction.
SD	(09)	State to read first byte of instruction.
SE	(09)	State to read second byte of instruction.
SF	(09)	State to seek indirect address.
SG	(09)	State to read indirect address.
SH	(09)	State to seek X register.

GLOSSARY

SJ	(09)	State to do index address computation.
SK	(09)	State to seek operand.
SL	(09)	State to read operand and execute bit manipulation instructions.
SM	(09)	State to seek A, B, or X register.
SN	(09)	State to execute change to A, B, or X (also P) register.
SP	(09)	State to load I with A, B, or X register.
SQ	(09)	State to modify P in Jump and Mark instructions.
SR	(09)	State to seek address for storing in memory.
SS	(09)	State to store data in memory.
ST	(09)	State to seek P register.
SU	(09)	State to seek A or B register.
SUM	(19)	Output of adder.
SV	(09)	State to read A or B register.
SW	(09)	State to perform shift/rotates in W register.
SX	(09)	State to seek A or B register.
SY	(09)	State to store W register in A or B register.
SZ	(09)	State to evaluate jump conditions.
S0/S5	(09)	Five flipflop state register.
T0/T7	(01)	Bit time counter. T0 is true during first bit time of a byte.
TM		A transfer to memory, i.e., a Store A, B, or X instruction.
TS	(21)	True for the byte time after SN in addition and subtraction. Controls writing of OV and CY.

GLOSSARY

TX	(18)	Intermediate signal used in shifts, rotates, and bit manipulation instructions.
UT	(04)	True during time location 200 (Output) is being read from memory.
WD	(23)	Write data. Data to be written in the memory.
WT	(23)	Write time. Time to write in the memory.
W0/W7	(17)	W register. W0 is least significant bit. A general utility register.

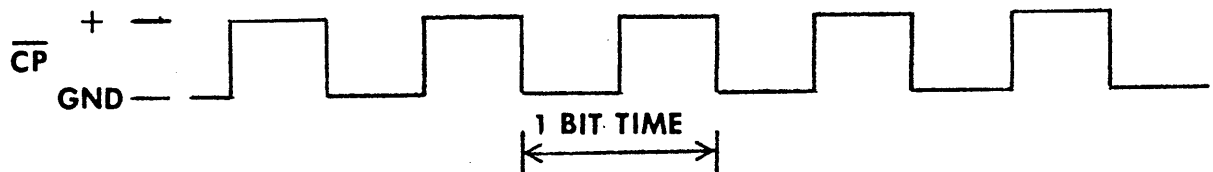
BLOCK DIAGRAM

The major logical elements of the system are shown in the block diagram on page _____. The KENBAK-1 computer differs internally in its structure from the majority of computers in these two ways:

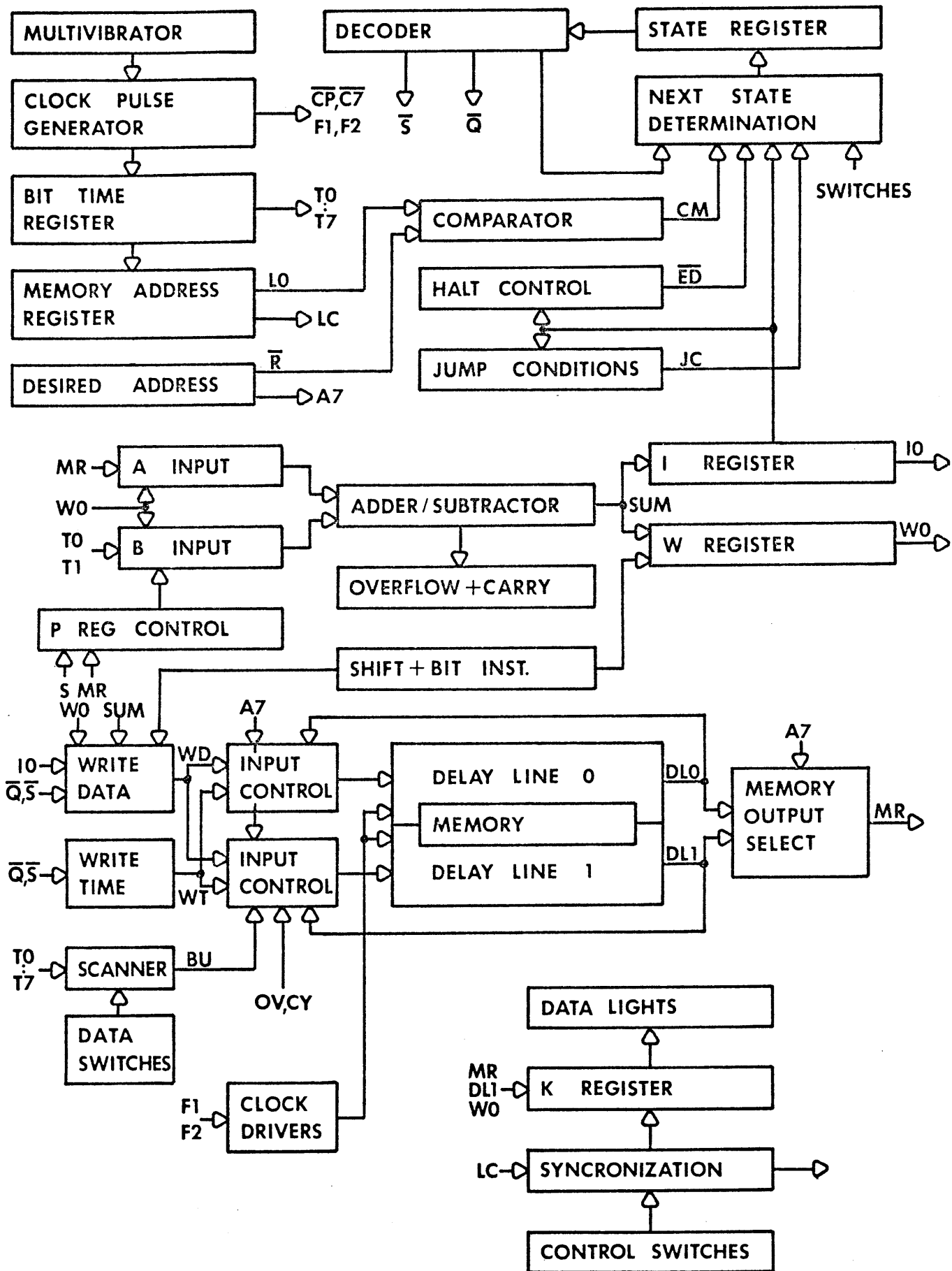
1. The computer is serial. Data is read, or written, from the memory one bit at a time. Thus it takes eight clock times to read a byte from the memory.

2. The programming registers (A, B, X and P) do not exist as actual hardware registers. Instead, a fewer number of hardware registers are time-shared to provide the necessary data manipulation.

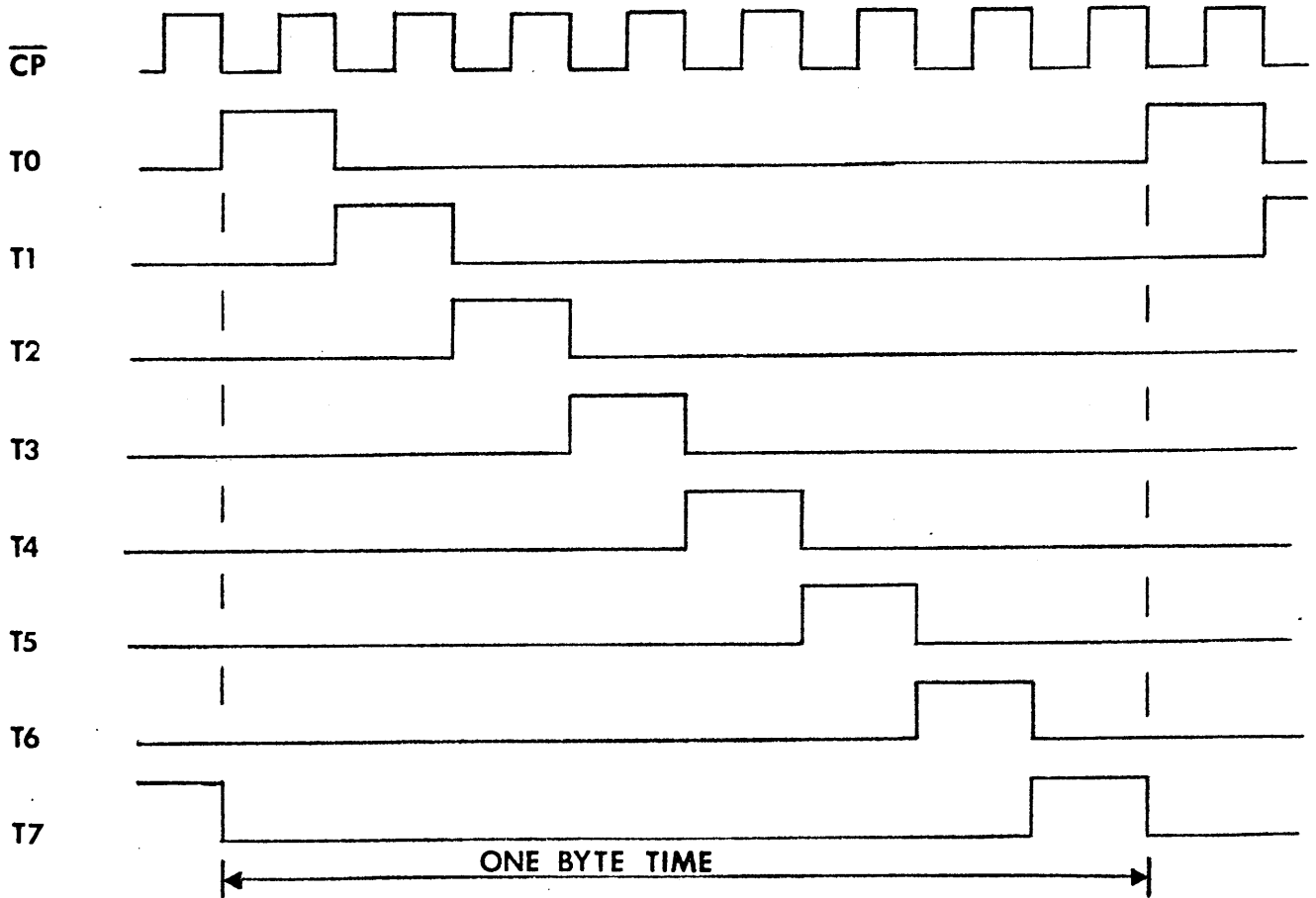
In the block diagram, the source of the clock signals is an RC multivibrator which runs at twice the fundamental system frequency. The clock pulse generator produces the basic system clock, \overline{CP} , which is a square wave. One bit time lasts from one positive-to-ground transition to the next such transition:



There are eight bit times: $T_0, T_1, T_2, T_3, T_4, T_5, T_6,$ and T_7 . These are generated by the bit time register in which a single bit shifts. Following T_7, T_0 is generated

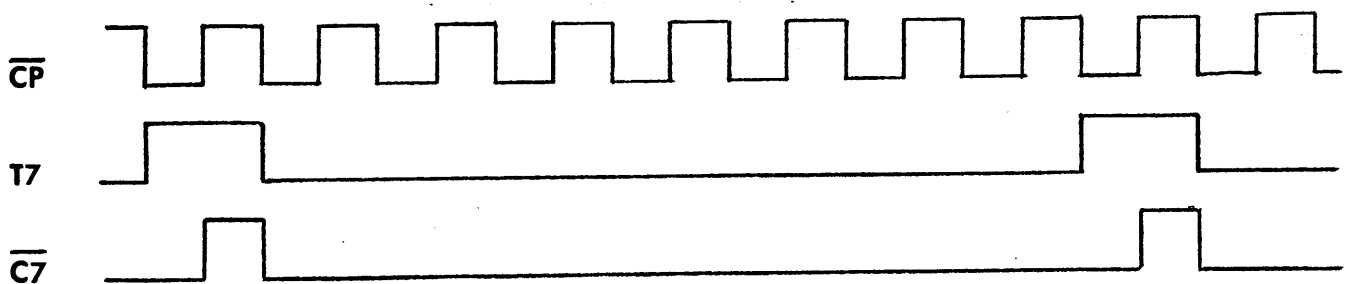


again and the cycle repeats indefinitely.

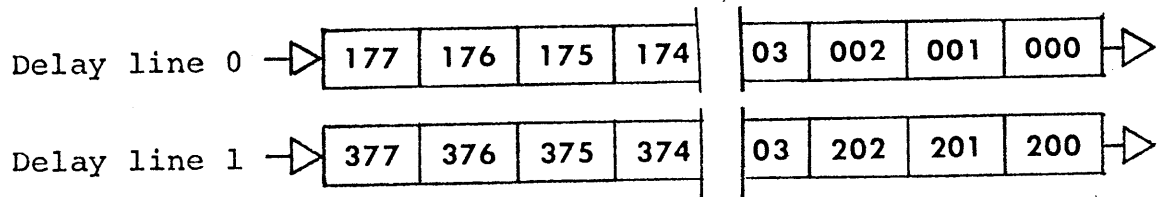


A byte time lasts eight bit times, from the start of T0 to the next start of T0.

It is useful to have a clock pulse which only occurs during T7 bit time. This is $\overline{C7}$:



The memory for the computer consists of two delay lines or shift registers, each of 1,024 bits. Therefore each will hold 128 bytes of information. Bytes with addresses 000 to 177 (octal) are in delay line 0 and bytes with addresses 200 to 377 (octal) are in delay line 1.



After byte 177 is read, byte 000 will appear in delay line 0. At the same time that byte 000 is read from delay line 1, byte 200 is read from delay line 2. Circulation of the information within the delay lines is continuous. It never stops.

Eight bits are required to address a byte since $2^8 = 256$. The most significant address bit, A7, selects the delay line. The seven least significant address bits select a position or a time slot within the delay lines.

The memory address register (the L register) shifts serially and counts one each byte time. The counting, or adding one, is a serial process.

Whenever reading or writing is to occur, the desired address is set up as the serial signal R. When the seven least significant bits of R are equal to the seven least significant bits of the memory address register, then the

next byte time is when the data for the desired address will be at the input and output of the memory. The most significant bit of R is used to determine A7, the delay line selection signal. The memory address register is, in effect, one byte time ahead of the memory.

The comparison between the memory address register and R is summarized by the signal CM during T7. Comparison exists if CM is true at this time. Address comparison does not exist if CM is false at this time.

As an instruction is executed, the computer proceeds through a series of steps. For example, the computer has to find address 003 (the P register), then it has to read and update the value of P, then it has to find the instruction, read the instruction, etc. While it is doing these things, the computer is in certain states. The fact that the computer is in a given state defines what it is to do while it is in that state. The data read from the memory and other conditions including the current state of the computer defines the next state. Thus the computer advances from state to state.

The computer has 29 valid states and 3 invalid states. Should an invalid state occur, which it shouldn't except when power is turned on, a valid state is generated next. The 32 total states result from the combinations of five flipflops, S4, S3, S2, S1, S0. The five signals are applied to decoders which generate 32 outputs. Only one of these

outputs can be true at any given time. This is the current state of the computer.

A state always starts and ends with the end of T7 (or the beginning of T0). It may last one byte time or it may last for many byte times.

The names for the decoded state outputs are two letter symbols which start with S or Q. No signal other than a state signal is assigned a two letter code starting with S or Q.

The nature of the decoder is such that the single true output is at ground while all of the rest of the decoded state signals are positive. Hence the outputs of the decoder are labelled $\overline{S_i}$ and $\overline{Q_i}$ to conform to positive logic.

The next state of the computer and when it advances are determined by many factors:

1. The current state,
2. Has the desired address been found,
3. Is a halt required,
4. The instruction and its addressing mode,
5. Are the jump conditions met,
6. The panel or control switches.

There is a serial adder/subtractor with two inputs, A and B. In subtraction, B is subtracted from A. Overflow and carry out of a byte are detected, and in addition and subtraction instructions, used to update the OF and CA bits

in locations 201, 202, and 203.

Most of the time, the I register is used as the instruction register to hold the first byte of the instruction. It is also used to hold the data for storing into the memory and for manual console operations. Except when information is being transferred in or out serially, the I register does not shift.

The W register is a general purpose working register which holds data, addresses and operands. It will be discussed at length later. Shift operations, for the Shift and Rotate instructions, are performed in it. Except for some bit times during these instructions, the W register is always shifting.

The memory (two delay lines) have their outputs connected to their inputs so that data is retained or held. New information is entered in different ways, but the usual way is through the use of the Write Data and Write Time signals. New data, equal to Write Data, is entered into the delay line specified by the A7 address bit when Write Time is true.

The input control for delay line 1 allows some special information to be entered into it alone. The states of the eight data switches are converted to a serial signal, BU. This is entered into address 377. The Clear pushbutton will erase the contents of address 377 to all zeroes. The overflow

and carry bits are entered into location 201, 202, 203 by direct entry through delay line 1 input control.

The actual delay lines used are MOS shift registers which require two clock signals out of phase with each other. One occurs during the even bit times and the other occurs during the odd bit times.

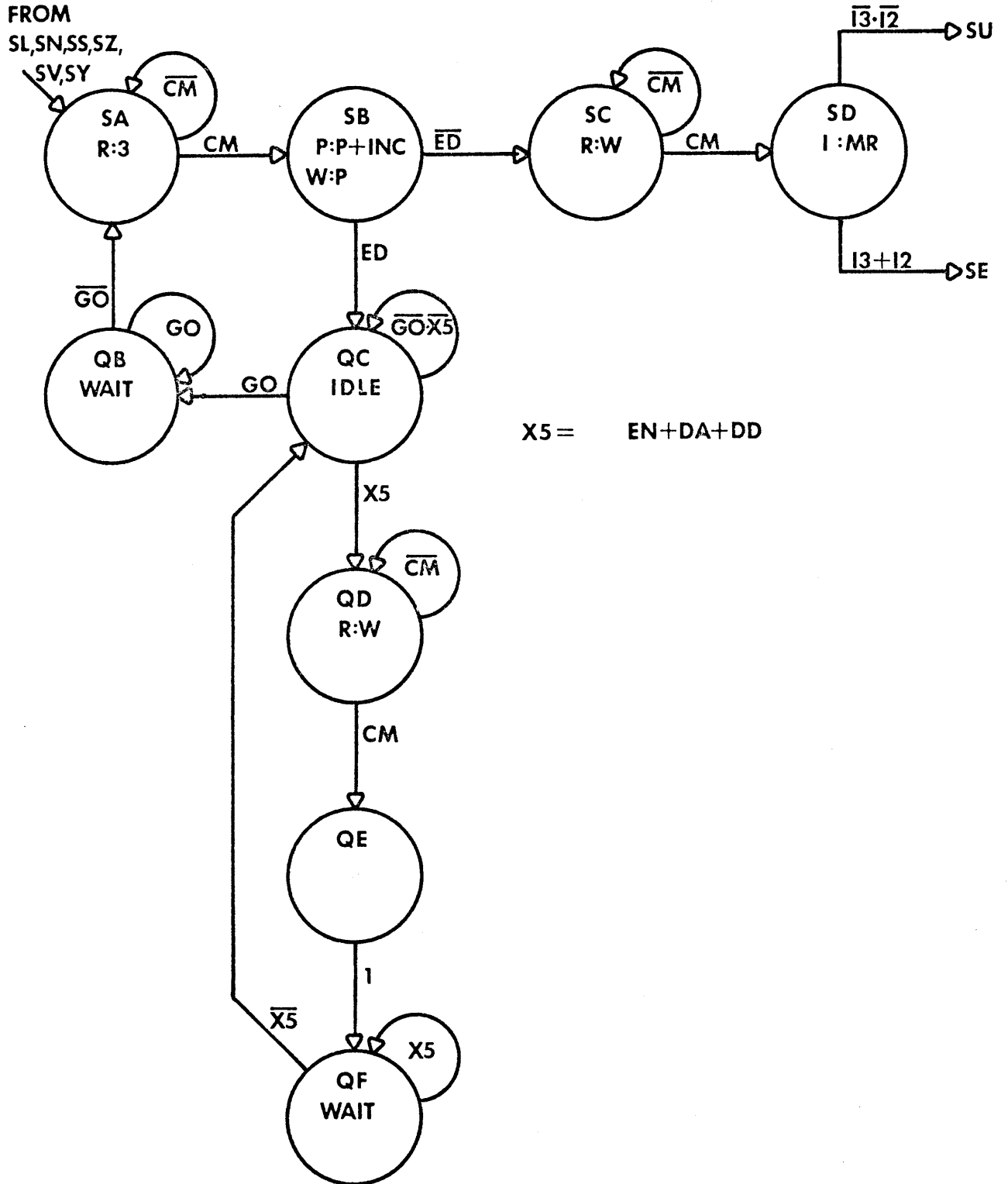
Several of the front panel control switches require some "cleaning up" before they can be used. After the contact bounce is eliminated they are used in the next state determination to control the reading and entry of data, the display of data, and starting and stopping of the computer. The data to be displayed in the lights may originate in the W register or in the memory.

STATE DIAGRAM

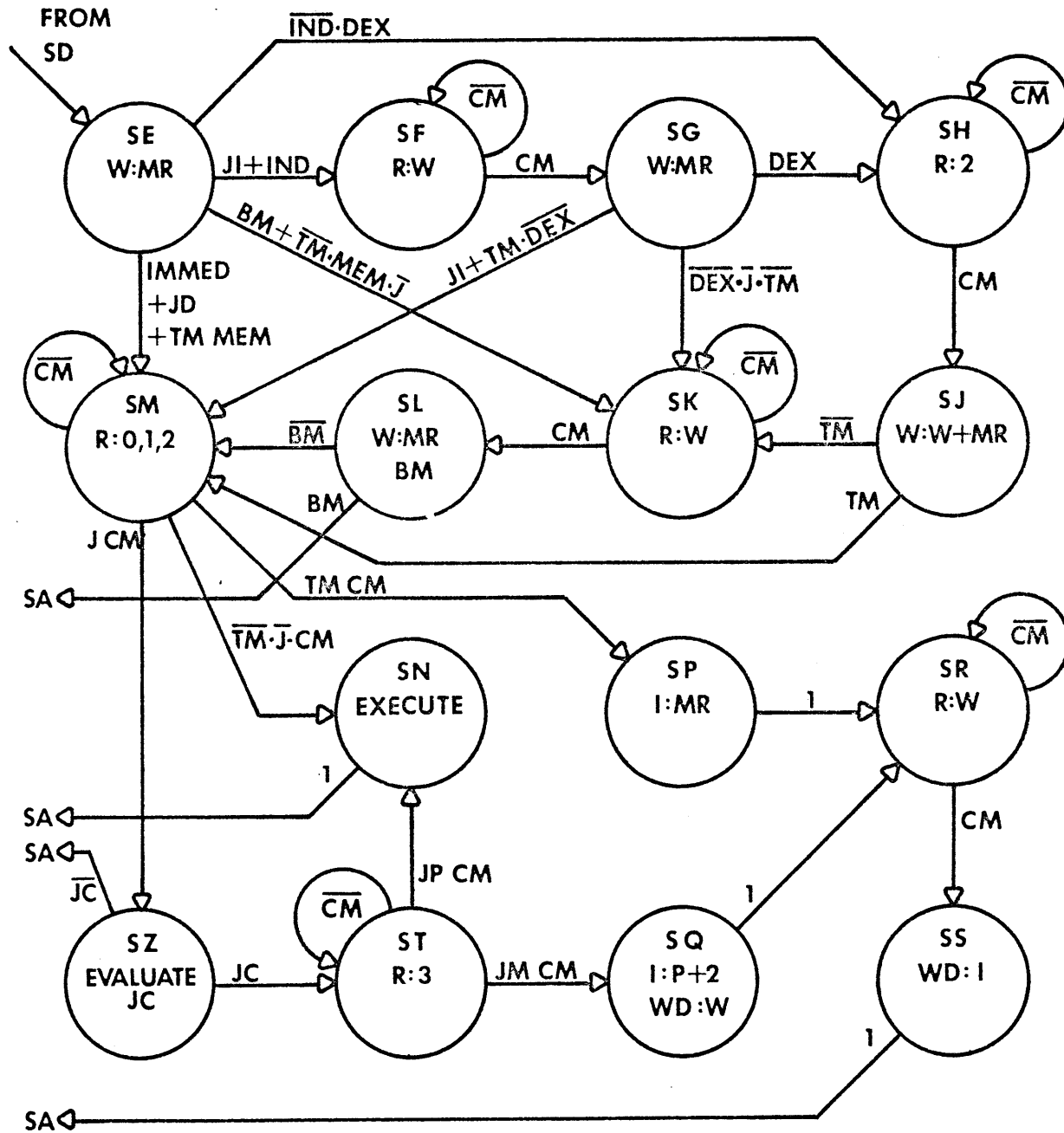
The state diagram for the computer is presented on pages 25, 26, and 27. State names which start with S pertain to automatic operations of the computer or the Run mode. State names which pertain to manual operations start with the letter Q. Of the three pages of state diagrams, the first page pertains to manual operations and automatic operations which are common to one and two byte instructions. The second page involves only two byte instructions and the third page is for one byte instructions.

A state lasts for one or more byte times. It always starts with the beginning of a T0 and it always ends with the end of T7. A state is represented in these diagrams as a circle. Directed lines, the state transitions, interconnect the states. Each line is labelled with logical conditions under which the transition occurs. All of the lines which leave a state must have a logical sum which is equal to 1. Otherwise the computer might become trapped in a state with no valid exit. Also, if more than one line leaves a state, the logical product of any two must be 0. Otherwise the logical transition would not be clearly defined.

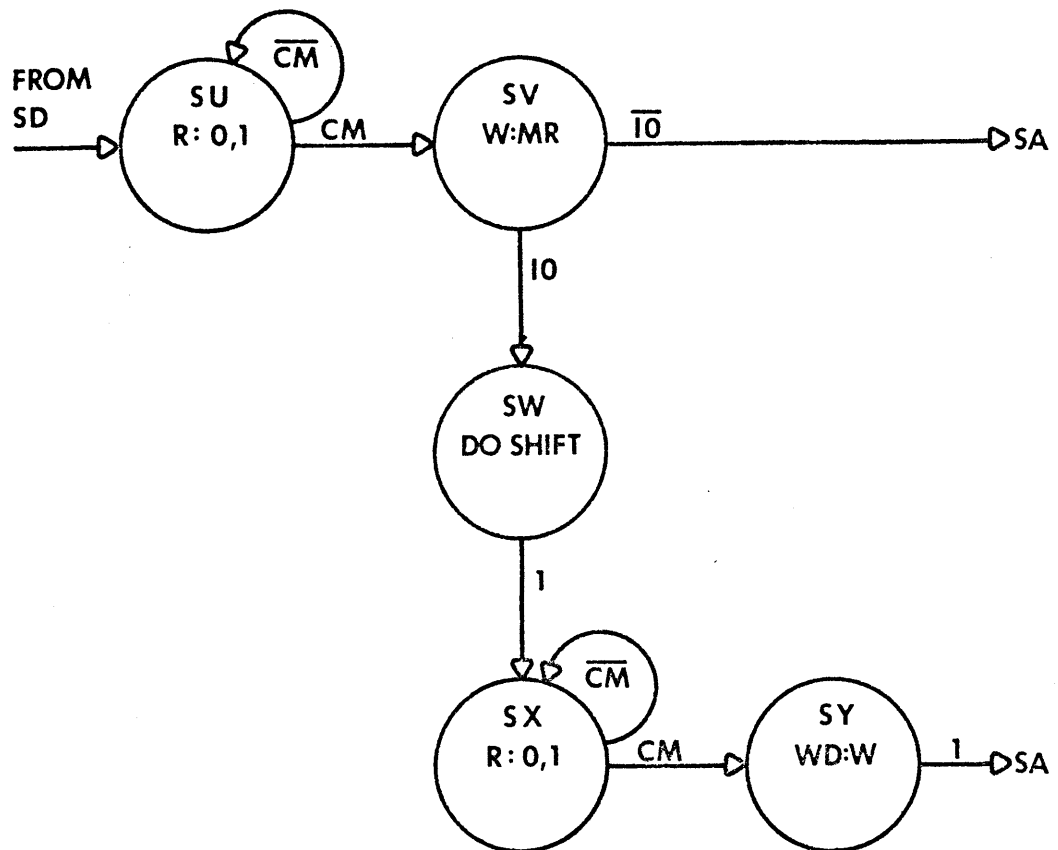
COMMON AND MANUAL OPERATIONS



TWO BYTE INSTRUCTIONS



ONE BYTE INSTRUCTIONS



SA

When the computer finishes any instruction, the SA state results. This may be regarded as the start of the next instruction. In state SA, the desired address is 3 which is the address of the P register. Therefore the function of state SA is to locate the P register. If, during T7, the comparison signal, CM, is false, state SA will be regenerated for another byte time. If the comparison signal is true, then the next state will be SB.

SB

State SB lasts one byte time. While it is true, the P register is being read from the memory. The value of the P register being read is the address of the instruction which was last completed. The P register increment control circuits, through signal PO, have a value which tells how many memory locations this last instruction occupied. This increment is added to the old contents to obtain the address of the next instruction. The sum is written back into the memory and is also transferred to the W register. At the conclusion of state SB, location 3 in the memory and the W register both contain the address of the next instruction.

If signal ED is true at the end of SB, this indicates the computer should stop, either because the last instruction was a Halt or because the Stop button was depressed.

Then the next state is QC. If the signal ED is false, the computer continues its automatic operations by going to state SC.

SC

The function of state SC is to find the next instruction. The desired address is in the W register and this is compared to the memory address register until a comparison is found. When CM is true at T7, the state register advances to state SD.

SD

During state SD, a one byte time state, the first byte of the instruction is transferred to the I register. The data path for this is through the A input of the adder. If $I_3 + I_2$ is true during T7, the instruction has two bytes. In this case the next state is SE. If $I_3 + I_2$ is not true, the instruction being read from the memory consists of only one byte and then the next state is SU. (The signal to use is $I_3 + I_2$, and not $I_2 + I_1$, because the instruction still has one bit position to shift in the I register. This will occur at the same time that SD ends.)

SE

During SE, which lasts one byte time, the second byte of the instruction is transferred to the W register. There is one exception to this rule. If the instruction is Store Constant (also called Store Immediate), the W register is

loaded with the address of the second byte of the instruction and not with the contents of the second byte. This is accomplished through the adder inputs.

	<u>A</u>	<u>B</u>	<u>SUM</u>
Store Immediate	WO	T0	Address of second byte
Not Store Immediate	MR	0	Contents of second byte

Since the W register contained the address of the first byte of the instruction, in the Store Immediate instructions the sum will be the address of the second byte. In all other cases the sum will be equal to the contents of the second byte. In either case it is the sum which is transferred to the W register.

Depending upon the addressing mode and the type of instructions, state SE may be followed by four other states. For indirect addressing, SF follows. For indexed addressing (without indirect addressing) SH is next. For direct memory addresses to find operands, state SK follows. If no operand is to be found, state SM follows.

Generally, regardless of the addressing mode, all of the two byte instructions will involve state SM which is a search for the A, B, or X register. If the instruction will change A, B, or X, then during SM the W register will contain the operand. If A, B, or X is to be stored in memory, then the W register will contain the address of

the location where A, B, or X is to be stored. If the instruction is a Jump, then W will contain the target address during SM.

The bit manipulation instructions are two bytes and have memory addressing, but they do not advance to state SM because they test or modify a location in the memory and not the A, B, or X registers.

SF

State SF is a search for an indirect address which is in the W register. When found, state SG results.

SG

During SG, the contents of the indirect address location are transferred to the W register. If indexing is required also, state SH follows. For a Jump Indirect or a Store without indexing, then SM is next. Otherwise state SK for finding the operand follows.

SH

In state SH a search for the X register, address 2, is made. When found, the state counter advances to SJ. State SH is preceded by SE if there is indexing without indirect addressing. It is preceded by SG if indirect addressing is also involved.

SJ

During SJ, a one byte time state, the contents of the X register are added to the contents of W. The result is

stored in W. For Store instructions, state SM follows.
For other instructions, SK is the next state.

SK

On entering state SK, the W register contains the address of the operand. This may have been directly from the instruction or it may have been derived from indirect and indexing operations. Using this operand address, a search is made for the operand. When found, state SL follows.

SL

For the non-bit manipulation instructions, the W register is loaded with the operand. At the end of the byte time, for these instructions, state SM is next. For the Set 0 or Set 1 instructions, the designated bit is set during SL. For the Skip on 0 and Skip on 1 instructions, the P register increment control is set as necessary. The bit manipulations are then finished and state SA is the next state.

SM

State SM is a search for the A, B, or X register as defined by the instruction. (For Jump Unconditional instructions, a search is made for the A register.)

For instructions which modify the A, B, or X registers, the W register contains the operand. State SN follows.

For instructions which will store A, B, or X in the memory, the W register contains the address where the data is to be stored. State SP follows.

For the Jump instructions, the W register contains the target address. The target address is the jump address or the "mark" address. State SZ follows.

SN

The instructions which change A, B, or X do so during SN. The W register contains the operand. A, B, or X is read from the memory. The required operation is performed and the results are stored into A, B, or X. One byte time is required and then these instructions are finished. State SA is next.

In the Jump instructions (see SZ and ST), the address in W is transferred to the memory to become the new value of the P register.

SP

To store A, B, or X in the memory, the I register is loaded with the contents of A, B, or X. Since this destroys the instruction code in I, all of the remaining parts of the instruction can only be dependent upon the state. Upon leaving SP for state SR, the I register contains a byte to be stored in the memory and the W register contains the address where it is to be stored.

SQ (See after ST)

SR

In state SR the address corresponding to the contents of the W register is sought. When found, state SS is generated.

SS

In state SS the contents of the I register are transferred to the memory. This state is used in the Store instructions or in the Jump and Mark instructions. In all cases it concludes the instruction and state SA follows at the end of the byte time.

SZ (Out of sequence)

In the Jump instructions, after the A, B, or X register is located in state SM, state SZ follows. The jump conditions are evaluated during this byte time while A, B, or X is being read from the memory. If no jump is to be made, state SA follows SZ. If the jump is to be made, state ST follows.

ST

State ST is a search for the P register, address 3. In the Jump instructions without "marking", state SN follows after address 3 is found. For the Jump and Mark instructions, state SQ follows.

SQ (Out of sequence)

Two things occur during SQ. The I register is loaded with the value of the P register plus 2. This is the return address. At the same time the P register,

address 3, is set to the value of the target address in the W register. This target address is where the return address will be deposited. The P register increment control is set to add one so that the next instruction will be taken from the address following the target address. After state SQ, states SR, SS, and SA, all discussed previously, follow.

SU

(The five remaining S states pertain to one byte instructions.) If the instruction read in state SD is a one byte instruction, then state SU follows. A search is made for address 0 or 1 (A or B registers). When found, the state register advances to SV.

SV

In state SV, the A or B register is transferred to the W register. For the Halt and No-op instructions this state is the last and state SA follows.

SW

Shifts and rotates are executed in state SW which lasts one byte time. Following SW is state SX.

SX

A search for address 0 or 1 (A or B) is made. When found, the next state is SY.

SY

The shifted or rotated value of the original A or B register is written back into location 0 or 1. This finishes the Shift and Rotate instructions and the next state is SA.

SZ (See after SS)

QA (Not used)

QB (See after QF)

QC

When the computer is halted, state QC is the idle state. If one of the control switches (Store Memory, Display Address, or Read Memory) is pushed, the control advances to state QD.

During IN time of QC, the I register is always loaded with the contents of location 377. If the Set Address switch is pushed, generating EA, the contents of the I register are transferred to the W register.

If the Start button is pushed, control advances to state QB.

QD

In state QD the address which is equal to the contents of the W register is sought. When found, state QE is next.

QE

In state QE, the action depends on which button was pushed:

Store Memory	(EN)	:	Transfer I to Memory, Add 1 to W
Display Address	(DA)	:	Transfer W to K
Read Memory	(DD)	:	Transfer Memory to K, Add 1 to W

State QF follows after one byte time.

QF

The state register control waits at QF until the control buttons are released. When they are released, state QC is generated.

QB (Our of order)

State QB "remembers" that Start was depressed but prevents control from advancing until Start is released. When Start is released, state SA is generated.

DEFINITION OF LOGIC ELEMENTS

The logical circuits for the computer are implemented with 14 types of integrated circuits. Each type has a distinctive schematic symbol. In the following pages these symbols are defined.

An integrated circuit package has 14 leads (16 leads for the type 7442). Pin 7 is ground and pin 14 is +5 volts (for type 7442, pin 8 is ground and pin 16 is +5 volts). A package may contain a number of independent circuits. Each circuit is defined by its input and output pins and a typical input-output relationship is given as a logical expression.

Additional information on these circuits can be found in manufacturers' data sheets or handbooks. Some manufacturers of these circuits include

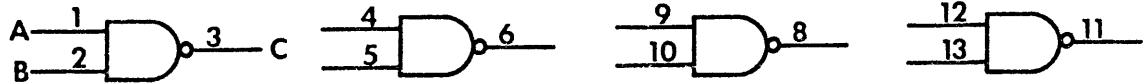
Motorola Semiconductor	National Semiconductors
Texas Instruments	Signetics
Fairchild Semiconductor	

though this is not an exhaustive list. Each manufacturer has a distinctive nomenclature for his units though they all generally involve the four digit 74-- number.

In the actual schematics, the input pins may be ordered differently from the definition.

QUAD 2-INPUT NAND GATE

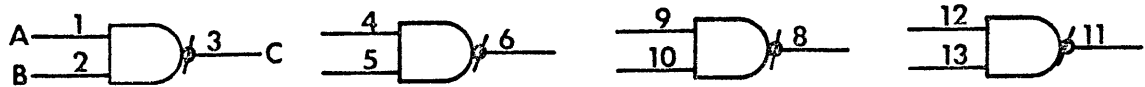
7400



$$C = \overline{A \cdot B}$$

QUAD 2-INPUT NAND GATE, OPEN COLLECTOR

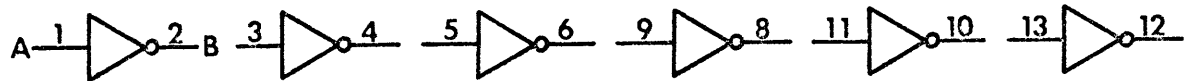
7403



$$C = \overline{A \cdot B} \text{ WITH EXTERNAL LOAD TO } +5\text{V}$$

HEX INVERTER

7404



$$B = \overline{A}$$

QUAD 2-INPUT AND GATE

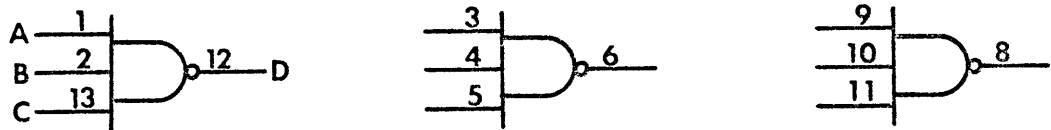
7408



$$C = A \cdot B$$

TRIPLE 3-INPUT NAND GATE

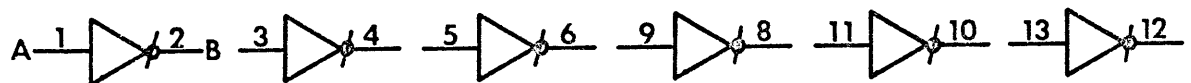
7410



$$D = \overline{A \cdot B \cdot C}$$

HEX INVERTER BUFFER

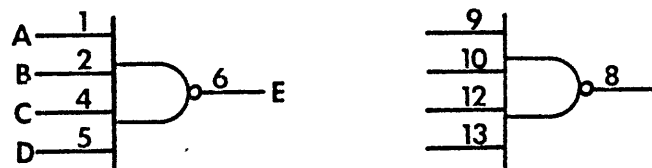
7416



$$B = \overline{\overline{A}} \text{ WITH EXTERNAL LOAD TO } +5\text{V}$$

DUAL 4-INPUT NAND GATE

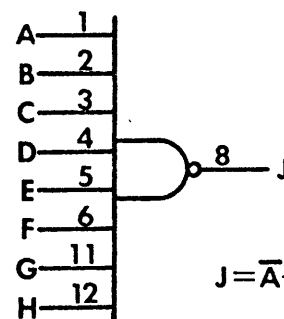
7420



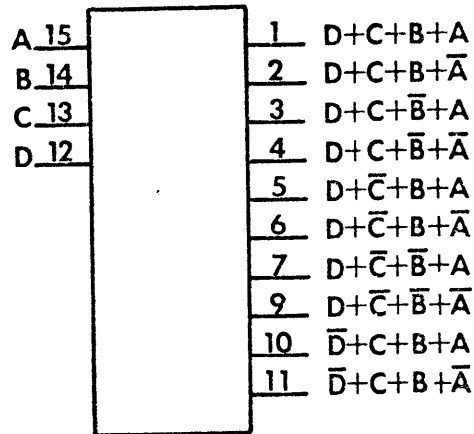
$$E = \overline{A \cdot B \cdot C \cdot D}$$

8-INPUT NAND GATE

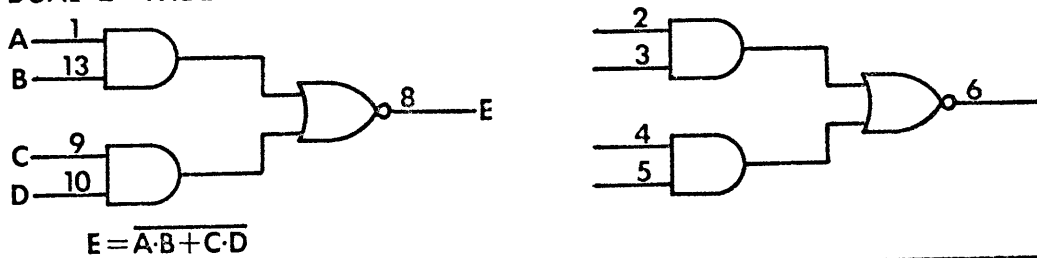
7430



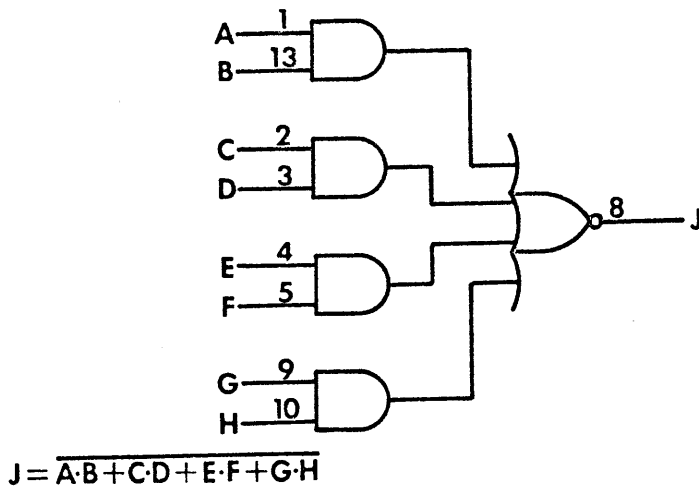
$$J = \overline{A \cdot B \cdot C \cdot D \cdot E \cdot F \cdot G \cdot H}$$



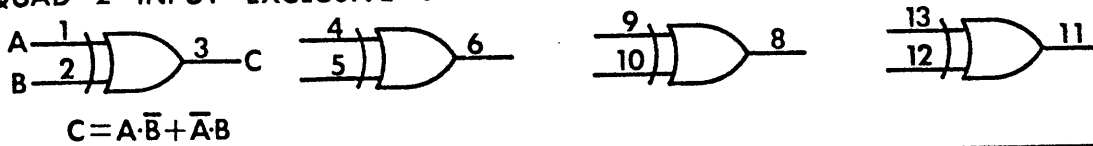
DUAL 2-WIDE 2-INPUT AND-OR-INVERT GATES

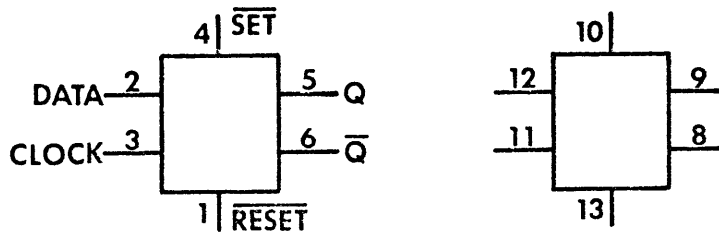


4-WIDE 2-INPUT AND-OR-INVERT GATE

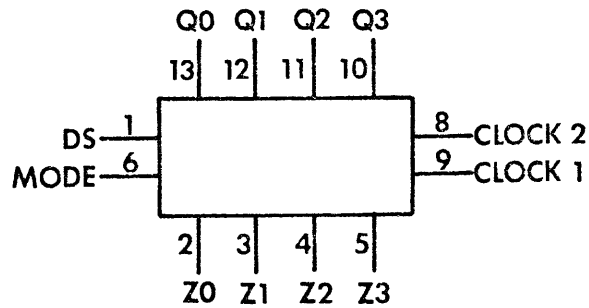


QUAD 2-INPUT EXCLUSIVE OR GATES

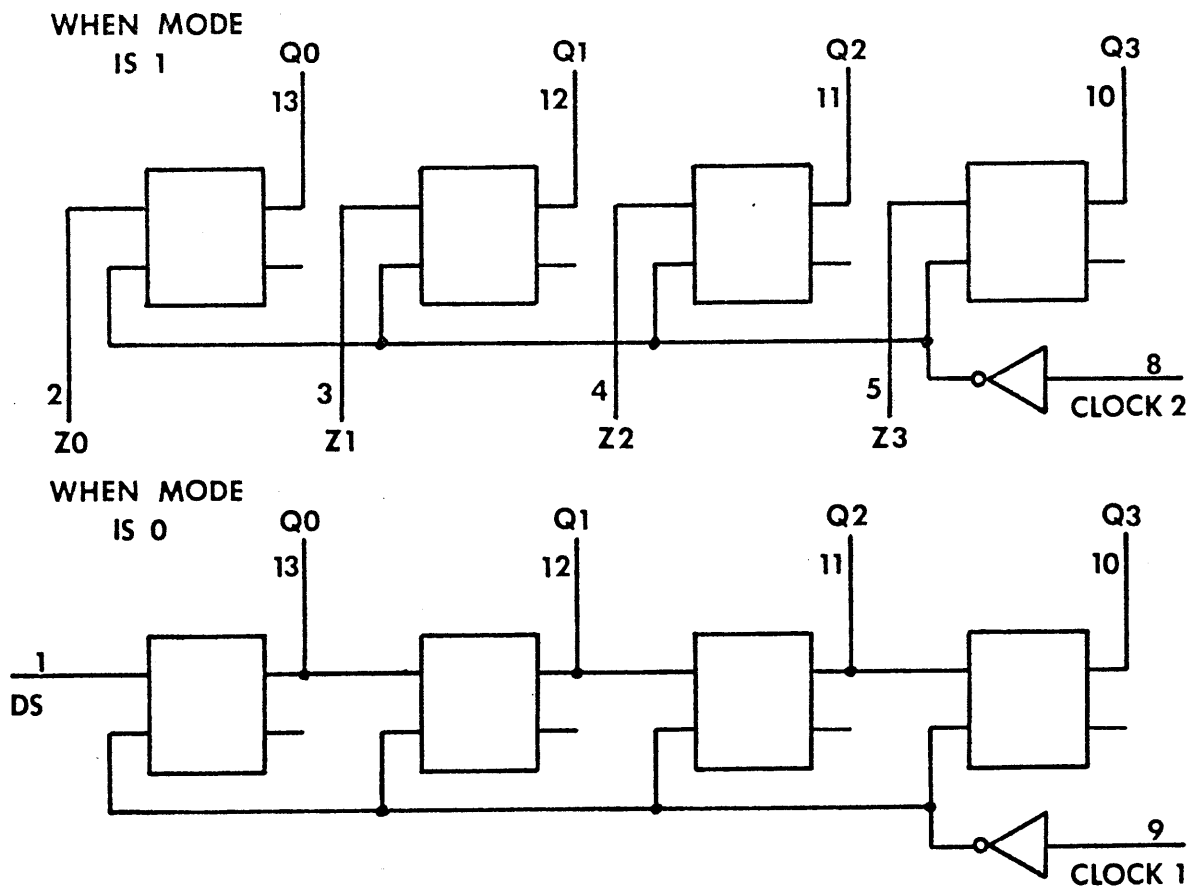




Q IS EQUAL TO DATA JUST PRIOR TO LAST 0 TO 1 CLOCK TRANSITION
 $\overline{\text{SET}}$ AND $\overline{\text{RESET}}$ ARE NOT USED



EQUIVALENT CIRCUIT



LOGIC SCHEMATICS

In the following pages, the complete logic schematics for the computer are given. Each page is given a "local" two digit reference number which is used to cross reference the sheets. Some signals are so widely used that they are not cross referenced. These include

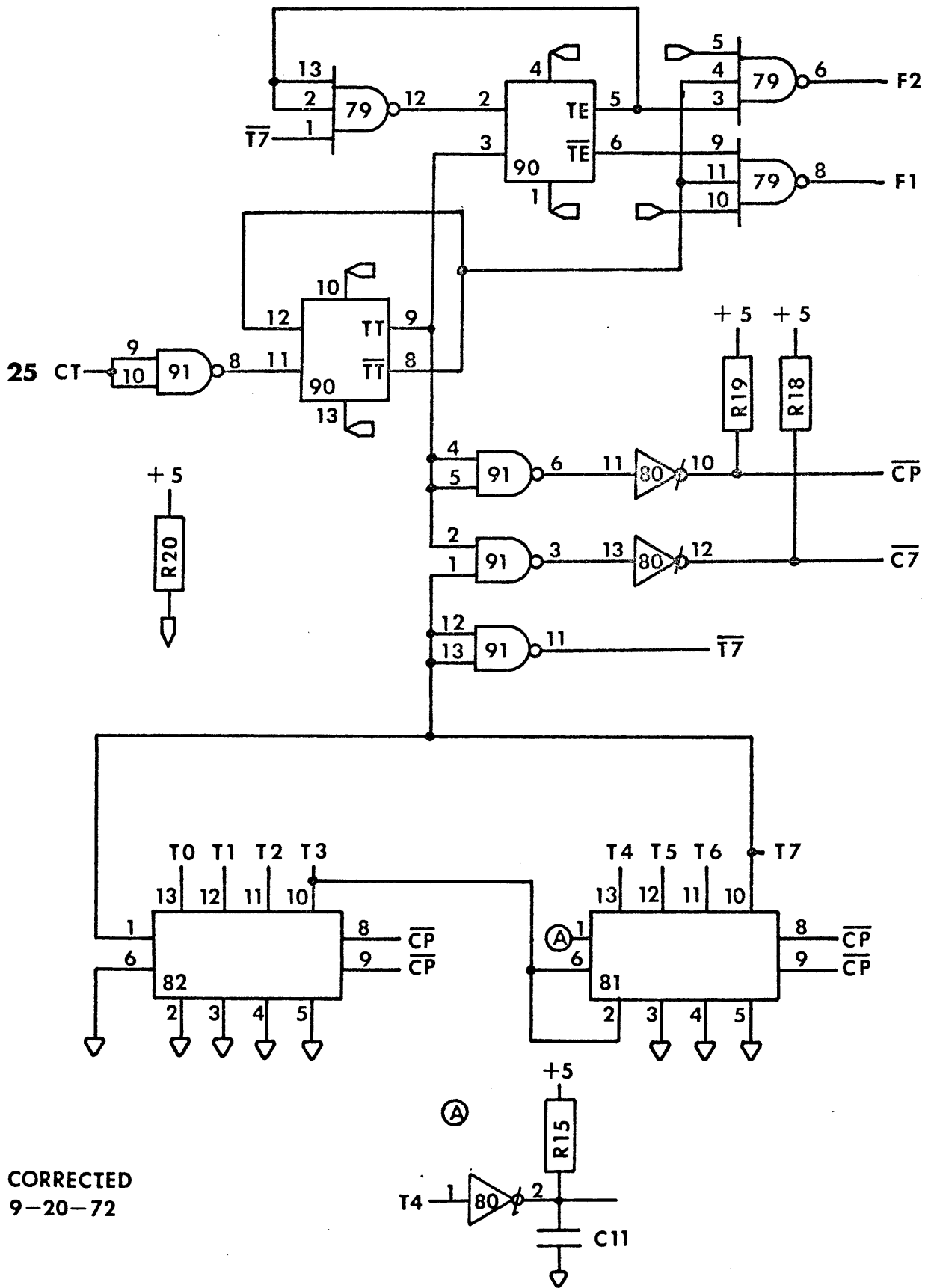
\overline{CP} , $\overline{C7}$, T0, T1, T2, T3, T4, T5, T6, T7 (sheet 01)

Original state signals, \overline{SI} and \overline{QI} (sheet 09)

Original I7, I6, I5, I4, I3, I2, I1, I0 (sheet 14)

In addition, signals which originate and which are used on the same sheet are not cross-referenced. Cross referencing is only backwards, from use to source.

Each logic element shown in the schematics contains a one, two, or three digit number which is the number of the integrated circuit package. This identity number is etched on the printed circuit board.



CORRECTED
9-20-72

