Department of Electrical and Electronics Engineering

Kumaraguru College of Technology

Coimbatore - 641 006

P-128

Certificate

This is to certify that the report entitled 'PRINTER BUFFER AND MULTIPLEXER' has been submitted by

Antonisamy, M.
Vel, T.
Kalaiselvi, A.

In partial fulfilment for the award of Bachelor of Engineering in Electrical and Electronics Engineering branch of Bharathiar University, Coimbatore-6 during the academic year 1990-1991

Guide

Head of the Department

Certify that the candidate was examined by us in the Project work viva-voce examination held on __________ and the University Reg. No. was __________

Internal Examiner

External Examiner

CONTENTS

LIST OF FIGURES

CHAPTER 2

CHAPTER 3

LIST OF TABLES

ACKNOWLEDGEMENT

We sincerely thank our Principal Prof. R.Palanivelu,B.E., M.Sc(Engg.),E.I.E, M.I.S.T.E., M.I.S.P.R.S., for his kind permission to work on this project.

We express our deep sense of gratitude to Dr. K.A.Palanisamy, M.E., Ph.D., M.I.S.T.E., FIE., for having provided all necessary facilities during the execution of the project work.

We are greatly indebted to our guide Thiru S.Ramkumar,B.E., and Miss. K.Ramaprabha,B.E., who has been a source of encouragement with their valuable suggestions and timely advice during the execution of the project work.

We wish to express our heart - felt thanks to all staff members, teaching and non-techning, of Electrical and Electronics Engg. Department, without, their help this project would not have been completed successfully.

PRINTER BUFFER AND MULTIPLEXER

ABSTRACT

The functioning of a small or medium sized computer system, is inefficient in the sense that, a lot of valuable time in spent in the printing of data, during which the computer remains stagnated. The rate at which the computer can transfer data is much faster than the rate at which the printer can print; in otherwords, the printer is too slow to keep pace with the data transmission from the computer.

The factor provides room for the drawback that an optimum use of the computer, in a specific period, is not made and there is a waste of time and resources. The solution to overcome this drawback is the printer buffer.

The printer buffer is an intermediate device which is used between a personal computer and a line printer. It accepts data from the computer at a very fast pace after which the computer can be utilised for programming a different task. The buffer holds the characters and until the printer is ready to accept them and the it feeds the data to the printer at a relatively slow rate.

In this project it is proposed to design, fabricate and test a hardware printer buffer with a memory capacity of 64K.

CHAPTER I

PRINTER BUFFER AND MULTIPLEXER

INTRODUCTION

1.1 Development of Computers - A Historical Perspectice

For the past few centuries, the ability to calculate has been closely associated with the development of technology.

One of the earliest forms of a computer was the simple muscle powered device called the 'Abacus'; which was developed in ancient Chino & Egypt. The primitive device was the only prevailing model until the next major advance came in the 18th century.

in 19th century, Joseph Jacquard invented a loom that employed punched cardboard cards to automatically control the production of patterened cloth.

CHARLES BABBAGE (1792 - 1871) was probably the first to conceive the essense of general purpose computers. He was of the opinion that an error free table could be produced only by a machine that would accept the description of the computation from a human being, but, once set up, would compute without human intervention.

In 1890 Dr. HERMAN HOLLERITH used punched cards & simple machines for processing data. HAWARD AIKEN and an IBM team completed the MARK I electro-mechanical calculator in 1944. In 1946 J.P. ECKERT & J.W. MAUCHILY developed the ENIAC which has limited storage capacity and required laborious programming.

In 1951 with the appearance of UNIVAC I, the first generation of computer technology began to be mass produced. The term "first-generation" is associated with the use of vaccum-tubes as the major component of logical circuitry with the advent of transistors, the size and reliability of the computer has been considerably affected. The "second-generation" machines which appeared in late fifties using transistors saw widespread installation and use of general purpose computers.

In 1960's the computer structure became quite important and the operating systems was developed as integral to the operation of these "third-generation" computers entered the market in the 1970's with Integrated Circuits [IC's] organized in the mass structure design [Large Scale Integration or LSI] and operating systems of great complexity.

1.2 PRINTERS

A most convenient and useful method by which the computer can deliver information is by means of printing characters. This can be effectively done by printers.

Operation: The coded group of binary bits are delivered to the printer, which decodes them and then prints the correct characters.

CLASSIFICATION OF PRINTERS:

(i) Important Printers

a) Character at a time

* 5 x 7 dot matrix printer
* Daisy wheel printer
* Cylindrical printer

a) Electromagnetic:- By using magnetic recording techniques, a magnetic image of what is to be printed can be written on drum surface. Then this is passed through magnetic power which adheres to the charged areas. The power is pressed on to the paper.

b) Electro static: The paper is coated with a non-conducting dielectric materical which holds charges when voltages are applied with writing "nibs". These nibs prints dots on the paper, which passes through them. Then the paper passes through the coloured particles carrying opposite charge to that written by nibs; as a result, printed characters are formed, become visualised.

1.3 PRINTER BUFFER

Printer Buffers are used to reduce the waiting time of a computer. Suppose there is a program which produces a printer output of several kilobytes; since the printer is relatively slow,when compared to the data given as output from computer, the user has to wait for the printer to print the whole output before he can run the computer again. This may lead to a lot of waste of time. This can be avoided by using a printer buffer. Printer buffer allows the computer to dump the characters into it in a short time and get ready to do other tasks. It then holds the characters to be printed out until the printer is ready to accept them. Meanwhile if its memory capacity permits, then it can again receive data from some other computer. This process of accepting data's from more than one computer is named as multiplexing.

The data flows from computer to the printer, through the printer buffer. The computer sends a byte to printer buffer interface. The microprocessor inside the printer buffer reads the byte and stores it in the 'Dynamic RAM". This process continues until there are no more characters to be sent or if the buffer is filled up to its fully memory capacity.

While sending out the data from the buffer to the printer the character are sent from the RAM to the central processing unit of the microproçessor and then from mircroprocessor to the printer interface.

The output to printer takes place simultaneously in the same order as the input. To perform the task it works on a Z-80 microprocessor with a clock frequency of 2 MHZ. It executes the instruction stored in the EPROM.

The buffer had been 80 designed that all the input and output work in a polled environment. The use of interrupt has been avoided, as printization of the interrupts would result in unsatisfactory operation.

The RAM has a multiplexer. So, additional logic is required for its operation.

The central processing unit used in Z-80 Microprocessor.

1.4 Scope of the "Project Buffer and Multiplexer"

In this project, it is proposed to design as printer buffer with 64K bytes of RAM, which means that over 64000 characters can be stored in the printer buffer. It can also be used to get the datas from some other PC while printing datas from 1st PC, It accomplish datas in different memory location thereby avoiding any without elapse of time for the other computer. But the printer paper will be available to user only after the previous matter of the 1st PC is completed.

The simpler multiplexer can connect a single printer to 2 PC's. By turning a switch one can connect the printer cable to the printer ports of either of the 2 PC's. Electronic Switching can be used using a handful of IC's which routes the data and interface signals coming in and going out, to and from either PC's.

CHAPTER II

DESCRIPTION OF MAIN COMPONENTS

This chapter comprises of dissertions in depth about the components which play a major role in the functioning of the printer buffer unit. The Z-80 central processing unit functions as the flag-ship for the whole unit. The dynamic RAMs are preferred in place of static RAMs as they are economic for a memory capacity of 64K.

2.1 The Z-80 CPU

The Z-80, introduced by Zilog, is an one-ship microprocessor that includes essentially all hardware and software features of the 8085 with many significant advantages. The Z-80 is packaged in a standard 40 pin dual in line package (DIP) using static n MOS. technology as in the 8085. There are 16 parallel address lines which are connected directly to the Z-80. Several design changes reduces the number of control lines used. There is a single pin clock input to which the clock is fed from an external clock circuit. It has a non-maskable interrupt request line similar to 8085 TRAP lines.

The Z-80 has 2 sets of registers. One set includes the main general purpose register and the other auxillary set of registers. The main general purpose register comprises of the accumulator A, the status register F and scratchpad registers B,C,D,E,H & L. These are mirrored by primes; A', F', B', C', D', E', H', and L' Z-80 instructions normally apply to the main registers. However, by using certain exchange instructions, the contents of the main and auxillary register can be quickly swapped. Such exchange instructions provide a rapid mechanism for state saving durin subroutine and interrupt processing. The auxillary registers also effectively doubles the temporary storage spaces in the CPU. The Z-80 has a pair of 16 - bit index register 'IX' and 'IY'. These allows the Z-80 to support all the addressing modes which are immediate, direct, indirect, indexed and relative. It also contains a new 8 bit register 'I', called as interrupt page address register. 'I' register provides the higher order half of the interrupt address used in processing of certain interrupt requests. It can be used by the CPU, which is the OPCODE of the CALL instructions, followed by a byte that acts as low-order half of the interrupt address. Z-80 appends this address byte to the contents of 'I' to obtain a 16 bit address vector that can point to any location in main memory.

The Z-80 contains an 8 bit register 'R' called the memory refresh counter. Its purpose is to enable the Z-80 to sypply the signals required for periodic refreshing of external dynamic memory circuits. During each opcode fetch cycle, the Z-80 places the contents of 'R' register on the system address bys activates a special refresh control signal RFSH. This information can be formed by simple refresh circuits to control the fefresh operations for the set of memory locations whose addresses correspond to 'R' register. 'R' register is automatically incremented after each instruction fetch, allowing it to sweep through a range of memory address. The entire function of memory refresh counter is automatic and refreshing operation has no effect on CPU's operating speed.

The Z-80 has about 158 assembly language Opcodes and 696 different Machine-language Opcodes. Most of the new z-80 instructions use 2-byte Opcode at the machine language level. The first byte of these Opcode contains one of the few patterns which are not used as an Opcode by the 8085. The second byte defines the Z-80 insturctions. Because of the use of 2-byte Opcodes. Z-80 instruction range in length from 1 to 4 bytes. Besides the instructions for 8 bit data processing, Z-80 has instructions for 16-bit, 4-bit, and 1-bit data, both logic and numeric. It also has an important class of instructions for moving and searching blocks of data of arbitrary length, 16-bit data transfer and arithmetic instructions. Instructions exist for moving a block of data from one part of memory to another, or between main memory and the 'IO' port. The instructions use certain specfic CPU register as memory or 'IO' address registers and byte counters addresses are incremented or decremented automatically, and the data are transferred byte by byte , until the byte count reads zero.

One of the most important characteristics of Z-80 CPU is its compatibility with 8080 and 8085. Programs in 8080 A and 8085 can be executed in Z-80 with minor changes, but the converse is not true since it has other features that are not present on 8080 A and 8085.

Z-80 has an extra instruction code that are left unsed in 8080 A. The Z-80 uses the 'P' (or PLO) flag to indicate 2's compliment overflow after arithmetic operations on the Z-80 the 'DAA' (Decimel

Adjust Accumulator) instruction will correct decimal substraction or decimal addition unlike 8080 A which does only decimal addition.

On the Z-80, the rotate insturction can clear the AC flag unlike in 8080A where the rotate instruction does not effect the AC flag.

PIN DESCRIPTION

Machine cycle [Output active low MI] indicates the address bus holds a valid address, for a memory Read or memory Write operation. The input/output request [IORQ] indicates that the lower half of the address bus holds the valid input/output address, for an input/output read or write operation, while IORQ signal is also generated where an interrupt is being acknowledged to indicate that the interrupt response can be placed on the data bus.

When RD output of CPU is active low it points out the fact that the CPU wants to read the data from memory or an I/O device. The address I/O device or memory should use this signal to get data on to the CPU data bus. On the otherhand, when the write pin is low it indicates to be stored in the addressed memory or I/O device.

If the refresh is low then the lower of the address bus, contains, a refresh address for dynamic memories and the current MREQ signal should be used to do a refresh read to all dynamic memories. An interrupt signal is generated by I/O devices. A request will be honoured at the end of the current instructions, if the internal software controller led interrupt enable flip flop (ZFF) is enabled.

The non maskable request line has a high priority than INT and is always recognised at the end of the current instruction, independent of the status interrupt enable flip flop. NMI automatically forces the Z-80 CPU to restart to a particular location.

FEATURES OF THE Z-80 MICROPROCESSOR:

1. Single chip n-channel silicon gate CPU.
2. 158 instruction which includes all 78 of the 8080 instructions with total software capability. New instructions include 4,8 and 16 b.t operations with more useful addressing modes.
3. 17 internal registers are present.
4. 3 modes of fast interrupt response plus a non-maskable interrupt
5. Directly interfaces standard speed static and dynamic memories with virtually no external logic.
6. Instruction execution speed of 1 us.
7. Single +5V d.c supply and single +5V clock.
8. All pins are TTL compatible.
9. It has a refresh output pin and refresh register with which it can refresh the DRAMs.
10. It has 16 address lines with which it addresses 64 KB cf memory. It also has 8 bit data line for data exchange and various other functions.

The pin details & Block diagram of the Z-80 CPU has been shown in fig 1(a) & fig 1 (b).

2.2 Dynamic RAM (MOS Technology)

Dynamic MOS RAM cells store data as charge on small capacitors rather than in cross coupled inverter latches as the static RAM cells do. These dynamic RAM cells, therefore, need refreshing for every 2 ms or less. The advantage of dynamic cell is that they are simple and smaller very economical and need less stand by power.

Current devices use a single transistor, cell. It has a simple switch connects storage capacitor to the DATA / SENSE line when its row is selected. In a RAM array of these cells, each column has a sense amplifier to write in to a cell when it is addressed. 4464 Chip is selected to be used as RAM. The pin out details and Block diagram are shown in fig (2.1) & fig (2.2).

It is packaged in a standard 18 pin and requires only +5V supply. The device has 8 address inputs.

When 8 bit row address is applied to the device address inputs by an external multiplexer the row address strobe input [RAS] is asserted low by external circuitary to latch the row address in to internal latches.

It is to be remembered that the external hardware must supply the row address RAS, column address CAS and WR signals with proper timing. For a write operation, a row address is applied to address input and RAS is asserted low. After a hold time, the row address is removed, WR is made low if it is not already low, data is applied to the data input, and a column address is sent to the address input of the device. Then CAS is asserted low to latch in the column address. The column address must be held stable for a definite time after the CAS goes low and data to be written in must be held stable for a definite time after CAS goes low.

REFRESHING A DYNAMIC RAM

As indicated earlier, dynamic RAMs store bits as charge or no charge on a tiny capacitor. Unless the capacitor is refreshed every 2ms or else, the charge will get drained and the stored data will be lost. To do this each row in the device must be accessed atleast once in every 2 milliseconds.

The refresh operation for the dynamic RAM simply means that all the groups of memory cells (256 in all) in one row are to be refreshed or read at once, there by enabling the internal tiny capacitor cells (that stores the datas in the form of charages) to be accessed and replenished by suitable amplifiers inside. During refresh operation only the row address and RAS are required; but all the 256 address lines must be refreshed atleast once every 2ms. During RAS only, no data goes to the data pins and refreshing alone is performed.

In automatic pulse refresh cycle mode of DRAM, internal row address register is automatically incremented after each pulse, so no external counter is needed. The REFRESH pulses can be supplied in either burst or distributed mode. The device cannot be read or written into, while REFRESH is low.

In the self refresh mode, RAS is made high and REFRESH is made low. If REFRESH is low for more than 2000 ms, an internal oscillator starts up and automatically increments the internal row address register to refresh the row every 2 milliseconds. The device cannot be read or written into in this mode.

2.3 EPROM / RPROM (Erasable PROM or Reprogrammable PRCM)

The EPROM and RPROM are read only user programmable memories that can be reprogrammed a number of times. There are two main types,

1. UV - erasable PROM.
2. Electrically - erasable PROM

A typical EPROM is erased by exposing it to hard (high - frequency) ultra - violet light for five to ten minutes, thus returning the contents of all the memory cells to zero by discharging them. An EPROM package has a characteristic asper: the seal on top of the chip is not opaque; it is a quartz window that allows ultra - violet light through. Once zeroed, the EPROM can be programmed with a special EPROM programmer. Selected locations within the EPROM can then be programmed and with in a few minutes a bit pattern can be installed in the EPROM. The component can then be inserted in the application board. If errors are detected or changes are desired, the EPROM can be unplugged and reprogrammed with in minutes.

EPROMs, however are expensive. In addition, EPROMs are often not pin-for-pin compatible with the final ROM or PROM that will be installed on the board. The speed and density of the EPROM is also significantly different from that of ROM or PROM.

Techniques to implement EPROMs

The "floating gate" technique is one of the best used.

A charge is accumulated in a silicon gate "floating" above the silicon substrate but isolated from it by a silicon-dioxide layer. The charge is induced in the silicon gate by trains of pulses. Once programmed, as EPROM is expected to retain its charges for 10 years. Erasure of the charge is accomplished with hard ultra violet light. When the photons the light hits the floating silicon gate, electrons from snallow energy levels are released and couse them to migrate to silicon substrate where their charges are neutralized. When the charge is neutralized, the corresponding bit reverts to ' O '.

The other important components used in the hardware circuit is shown from fig (2.4) to fig (2.9).

CHAPTER III

HARDWARE DESCRIPTION OF PRINTER BUFFER

3.1 Circuit Description

In order to accomodate the memory buffer a 8 bit microprocessor is used. For this Z-80 is selected due to following reasons

1. Efficient memory transfer instructions.
2. It has provision for refreshing dynamic RAM CHIPS.

Additional memory can be obtained in buffer by using dynamic RAM chips. In addition to RAM and the Z-80, an EPROM is needed to house the software which deals with the buffer storing, printing and multiplexing jobs. Input and output ports are used to receive the printing data from the printer port of PCs as well as to route the buffer data to the printer itself. Input port is also used for handshaking signals and an output port for audible output and LED indicators. The buffer unit needs only 5V supply, which can be obtained from one of the PCs power supply connectors.

Input port IC is 74LS373

Output port is 74LS373

Description of I/O Ports:

These 8 bit register feature 3 state outputs designed specifically for driving highly capacitive or relatively low impedance loads. The high impedance 3rd state and increased high logic level drive provides these registers with the capability of being connected directly to and driving the bus lines in a bus - organized system without need for interface or pull-up components.

The 8 latches of LS373 are transparent D-type latches, meaning that when enable [C] is high the Q output will follow the data (D) i/ps. When enable is taken low the output will be latched at the level of the data that was already set up.

Clock Circuit:

IC 74LS373 (ICI) is used for generating an 8 MHZ Clock signal with 2 TTL inverters and a crystal. The buffered clock of 8 MHZ comes from a third inverter and is used for refresh timing logic. This is further divided by four in two flip-flops using a single IC 74LS373 (IC2). This 2 MHZ is buffered with an inverter again, pulled up by a 330 ohm registor and passed on to the Z-80 clock pin. This pull-up is required for the Z-80 clock, when driven by a TTL logic.

The Z-80 Chip has 16 address lines, 8 data lines, two interrupt inputs (NMI and INT) five read write control signals, and one clock input. The circuit of the Z-80 & buffer memory is shown in fig (3). The reset pin 26 of Z-80 (IC3) is connected with a pull-up resistor and a capacitor to ground so that on power up the Z-80 resets itself. Since the bus control is not employed in this circuit, the BUSRQ and wait pins are tied up to the positive 5V supply as well as NMI interrupt pin. The INT pin is connected through a resistor to positive and if needed for any expansion, this interrupt pin can be used.

Address Decoding

The total 64K of memory which a Z80 can address is divided into 2 parts.

a) 4K of EPROM (0000 - OFFF)

b) 60K of dynamic RAM (Through a 64K RAM is fitted) (1000 - FFFF)

The address for EPROM is fixed in the lowest 4K space. For this an IC 7427 (IC4) with its triple 3-input NOR gate and two inverters is used. IC4a (7427) combine A15 and A14 address lines; IC4b combines the address lines A13 and A12, all of which would have to be low for EPROM selection. The NOR gate outputs are inverted in ICs 5a and Id. IC4c gates a low signal at pin 9 if all A12 - A15 are low.

The MREQ signal goes to its pin 10. The inverted RFSH signal from Z80 also goes to pin 11. Thus gate 'C' of IC4 generates a high-going signal for memory requests in the address range (0000-OFFF) This is inverted in IC le and is used as the chip-select signal for the EPROM. Note that the EPROM is not selected for refresh operations by the Z80 since it does not need it.

The Dynamic RAM Refresh Circuit

Fig (2.3) explains the working of the DRAM interface to the Z80. Fig (2.3) shown is the timing diagram for the Z80 and the DRAM. During each Opcode fetching or M1 cycle, the Z80 generates a refresh signal. During T3 and T4 timing states, when the Z80 is decoding the fetched instruction, it puts a refresh address on the address bus and pulses of pin 28 goes low. Only A0-A6 address lines are sequenced from one M1 cycle to the other, so that all possible amoung 128 addresses are addressed one by one in 128 fetch cycles.

The MREQ signal also goes slow during this time, because the refresh address is generated to be stable only when MREQ is active low. The Z80 has merely a modulo 7 counter for refresh addresses; it uses lines A0-A6 only when it puts the refresh address on the bus.

In this circuit, a dynamic RAM of the 64K, 4 bit variety is employed. These are organised in 8 rows and 8 columns. With 8 lines, 256 memory cells can be accessed and they are all in eight column for each row, the memory is totally 256 x 256 or 64K. There are 4 bits per memory and so 2 such 4464 chips (IC 6 & 7) make up all of the Z80's memory.

Dynamic RAM for 64K have only 8 address lines.

Principle of picking up a memory cell for reading or writing is:

Row address is fed to the eight input lines A0-A7. This enables the row buffers, choosing one amoung 256 rows, A little later, the required column address is fed to the same eight pins, and the column address strobe is pulsed low. This causes the choice the selected columns from among the 256.

The Z80 does not provide an address in this manner (ie) first choosing the row address and then a column address. All it does it to output an 16 bit address making use of lines A0-A15.

Thus we have to split these 16 address lines into 2 groups of 8 and supply them one after another to the DRAM chip. For this a pair of multiplexer IC8 the row address is formed with A0-A3 and the column address with A8-A11. The multiplexing signal is one that switches the column address after the row address has first been given to the RAM. Fig(3) shows how the timing of this signal is midway between the RAS and CAS signals that strobe the row address and column address inside the DRAM chips. IC9 is another multiplexer (74LS157) which groups the remaining eight address lines A4-A7 and A12-A15. The former is selected for rows and the latter for columns.

Refresh Operation of dynamic RAM

It means that all the groups of memory cells (256 in all) in one row are to be refreshed or read at once, thereby enabling the internal tiny capacitor cells to be accessed and replenished by suitable amplifiers inside. During refresh operation, therefore only the row address and RAS are required; but all the 256 address lines must be refreshed atleast once in every 2ms. During RAS only, no data goes to the data pins and refreshing alone is performed. The block diagram and pinout details of the 64K x 4 bit memory chip are shown in fig(2).

Z80 provides only 128 row refresh addresses. Thus the A7 line will not get affected. However, the RAM chips may have to be accessed on their 256 rows. To accomodate this we have to generate the A7 signal externally. Thus if we divide the refresh signal from Z80 by a counter 256 times, its output will be high for half the time and low for half the time. That output will thus be able to give an equivalent A7 signal, called RA7 or row A7 signal.

In IC10 (74393), there are 2 divide by 16 counters; its output therefore gives an equivalent A7 signal for refresh operations. IC11 (74157) is another multiplexer that chooses this generated A7 signal for refresh operations and the normal A7 signal (from Z80) for memory read / write operations. IC11 (74LS157) is a switch that is controlled by the RFSH signal from Z80 at its pin 1. Only one switch (among 4) is used in IC11.

To explain how RAS, MUX and CAS signals are generated, following every memory reqest signal. For this ICs 12 - 14 are needed. [IC 74LS32, IC74LS00, 74LS175]

Pin 8 of IC 4C [74LS27] goes low for refresh operations and for memory selections other than the EPROM. Thus it selects the DRAM both for refresh and actual read/write operations. MRAs or row-address strobe signal is got by combining MREQ and RFSH in IX12b (74LS32), which is an OR gate. This is given to the DRAM chips via a current limiting 33 ohm resistor.

MRAS is combined with the memory access signal of IC4c

[74LS27] (pin 8) in another OR gate [IC12a] and inverted in IC5 [7404]. The positive edge of this signal is time delayed in a series of 3-D flip flops to generate MUX and CAS signals. IC14 [74LS175] is a quad flop-flop. The clock has a speed of 8MHZ which is 4 times faster than the Z80's timing clocks T1, T2 etc. Thus 2 clocks after the RAS, we get the MUX signal and after three clocks the CAS signal are produced one after another. The MUX signal provides the cloumn address which is set ready for the column access by the time, when CAS becomes low.

In this type of 4 bit DRAM, there are 4 data pins which are controlled by the output-enable pin 1 to read the data. Data is enabled whenever memory read and CAS are active, by combining them in another OR-gate of IC12d.

INPUT OUTPUT SELECTION LOGIC

IC15 is the address decoder [IC74138] which selects I/O addresses 80-87H by using A7 to select the chip when high. These port adresses are used for the printer output and input ports from the PC's, printer ports, which are shown in fig (3).

The printer ports of the PC's are connected to the multiplexer board via two D25 to D25 cables. The PC printer port handles all the printer signals, but now the only signals needed for the multiplexer-buffer are the eight data, one strobe and the two BUSY and ACK signals. Two input ports are configured for reading the data from the two PCs.

IC16 [74LS373] & IC17 [74LS373] chips are wired for input. The data can be read by reading ports 82 and 86. Pin 1 of the chips are the output enables pins for the latches in these chips.

The data from the PC meant for the printer, is strobed by the 'strobe' signal coming from pin 1 of the D25 connector. This is given, after an inversion in one of the IC5's inverters to latch pin 11 of IC16. The data meant for the printer is latched in IC's 16,17 and the Z80 program will read it subsequently.

HANDSHAKING WITH THE PERSONAL COMPUTER

When data meant for the printer comes from the PC's printer port and is taken and filled in the RAM buffer, there should be a proper handshaking, so that the PC should think that a printer is receiving the data. Otherwise the PC's printing software will get out of function and display an error message on the screen; and hence it may not send further data at all. The 2 signals used are BUSY and ACK. After a data byte is latched, BUSY is generated and kept active high until the latched data is read. Thereafter, the ACK, signal is pulsed low to inform the PC that the data has been received by the printer.

IC13 [74LS00] and IC18 [74LS00] are used to form the two set-reset flip flops, one for each port. The negative edge of the strobe signal sets the flip flop andgenerates the BUSY signal that is fed to pin 11 of D25 connector. This tells the printer port of the PC not to dump with any further data.

Now software looks at the BUSY signal via another input port IC19 in bit D3 (bit do for the other PC). Having learnt the printing matter is comming from either PC, the input port latched data is read from IC16 [74LS373] and saved. After saving the data, the ACK signal is generated on port 84 [or port 87]. This generates a deviec-select pulse that goes to reset the flip flop formed by IC13 [74LS00] gates or IC18. SO the BUSY signal goes low again; the same pulse also serves as the ACK signal for the PC's printer port.

After saving data from either PC it is stored in the buffer. The buffer output must be routed to the printer. The printer is connected via an output port and another D25 output connector. To handshake with the actual printer, the strobe signal is to be generated. Port 80 is used for the printer port. When port 80 is written, data from memory gets latched and is availaible for output from IC20 [IC74LS373].

The device select port 80 pulse also serves as the strobe signal. A small capacitor (330 PF) is required to stretch this pulse as per printer requirements. The BUSY signal from the printer port 83. Furter data is transferred if this goes low. The printer generates this signal after receiving the byte sent to it. Thus the buffer memory is fully transferred to the printer until the end of the printer matter.

Management Printer Errors:

Tghe printer is initialised automatically on power up. There are occasions when the printer get stuck for some mechanical problem like with its ribbon and head, in which case, the printer sends an error signals. This will normally be recognised by the PC and would cause, while printing a matter, an error message to appear on the screen. When this printer multiplexer is employed, the error from the printer is received by the unit and will cause a beeping sound. If there is no paper, the printer itself sounds an intermittent beep and so this eventually is not handled by this unit. As far as PC's are concerned the printer error will not be received by it as the error pin connection is not made to the PC's printer port. Even if it is connected, it will serve no purpose at the time error appears, the PC will not be in a position to recognise it, as it would have dumped with all the printing matter into the buffer and may be engaged with some other function. If any error such as "PRINTER NOT READY" appears on the screen, it may be only due to the buffer is not receiving the datas.

A simple peizo-sounder which is used in toys is fitted and connected to bit Do of the output port IC21 [74LS373] whose address is 81H. A software routine, under such conditions, sounds a continous beep.

Three LED's are provided on this output port to indicate data flow from the PC's to the printer. Two switches are provided on IC19 [74LS373] which are read by software to know how many PC's are shared by the single printer. Infact, one could add an additional PC by adding one more connector and an input port.

3.2 CONSTRUCTION AND TESTING

The PCB is a double - sided PTH type as shown in fig(4) and fig(5). The component layout is given in fig(6). There are three 26-pin PCB connectors, in two rows of 13 pins. These have to be fixed with Berg strip when have 0.6 mm square posts spaced 2.54mm apart. Such Berg strips with gold plated posts are available with 36 pins and are very useful in these connections.

The Berg strip can be broken to select 13 posts and fixed into 2 rows on the PCB, in the holes provided. Plug connectors are used according to the posts on these. A 26 untack insulation displacement type plug connector is required for each and hence totally 3 in this case. A flat 26 wire ribbon cable will have to be fixed on the plug connector for making a DP plug jumper cable. Several resistors few capacitors (one for reset, one for printer strobe and the remaining for decoupling) and the IC bases are all mounted (Taking care of pin1, with reference to fig(4), which shows the components layout). The crystal is fixed at the board end, near IC1[74LS04].

The one output port is connected to a buzzer and 3 LEDs. They indicate printer erroror buffer full sound and the data flow on to the multiplexer. These can be wired outside the board taking the connections off the holes provided in the PCB.

There are three ports on the right side of the board; the top one is for printer and the two on the side are the inputs from the printer ports of the 2 PCs. Berg strips are fixed on the two rows of holes and socket connectors are used for making the connection to the PC's printer port.

Now the board is ready for testing.

First a +5V supply is connected to the board. IC 1,2,3 and 4 are fixed. The 8MHZ clock would be noted on a scope at pin 4 of IC 1 , IC 2 would divide it by 4 and the 2 MHZ clock should be seen on pin 5 of 74LS74 and thereafter at pin 6 of the Z80. Since power on reset is provided, the Z80 would now generate address pulses on the address lines.

After switching off, one has to check whether there are any shorts in the address and data lines, between themselves or with the power supply. All the IC's should be getting the power supply of 5V. Dynamic RAM chips configuration is as follows. Pin 9 is +5V and pin18 is ground. There must be 1 or 2 monolithic decoupling. 1 uf capacitors near the power supply pin of RAM. Also 33 ohm resistors are connector to each address lines and the RAS, CAS and OE pins.

In order to test the I/O port and just to check if the Z80 is functioning properly in the EPROM. This will operate if Pin21 of the EPROM is tied high. By opening out that pin while inserting it in the socket and soldering a wire to that pin and taking it to +5V. For normal working with all its pins in the base.

After finding up all the ICs and checking for clock and address lines, one can observe the output pins of the printer port on pins 2-9. This would show the output on port 80. Since we have given a looping program (Table-1) there should be pulses on the strobe pin 11 of IC20. The data byte AA should be noted on these pins.

TABLE I

```
800     3EAA       beg; LD A, AAH     Check AA on Print port
802     D380       OUT 80 H           To look for Pulses.
504     DB81       IN 81              On port ICs
806     DB82       IN 82
808     DB 83      IN 83
80A     3EOF       LD A, OF           Sound and LEDs on
80C     D381       OUT 81
80E     C300 00    JP BEGIN
```

The address for the program listing is given as 800, because that is for use when pin 21 is held up. In use upon reset, this program will read from the ports and output AA continuously on the printer port. One can check for pulses on the respective pins of the 373 ICs. The data on the output should show AA, by looking at pins 2 - 9 of the printer port. Also if a buzzer is connected to pin 2 of port 81, that should sound and all 3 LED's will glow. After this test the board is almost free from hardware errors. But dynamic RAM is not tested by this method.

EPROM pin 21 should be properly fixed on the base. When power-up, the regular printer multiplexer program will be functioning. That would accept data from the PC's initially. Hence, pulses should be noted on the input ports 82 and 83 at pin 1 of the 74373 chips. Now the board has gone through the hardware checkup, it now needs to fix it up properly and interconnect the printer to the 2 PC's. A sheet metal box with 3 connectors (D25) is preferred. The box should have the D25 connectors in the front and back. So that the cables are coming from the two PC's at the front and going to the printer at the back.

Since printer errors will now show up only by the sound from the multiplexer, it should be positioned in such a way that both PC users are able to hear it.

Once the board has been tested with pin 21 of EPROM is opened out and connected to positive and if the I/O functions are working well then, it is almost ready for use. The EPROM can now be fitted with pin 21 inside. That would cause the program to be executed from the 0000H address in that EPROM, where the print-up-mux software is programmed.

In order to test the unit with the PCs and the printer, it is required to connect the input ports to the printer ports of PCs and the printer cable to the output port of this unit. It is enough to have one PC for testing the unit first. To simulate data for printing from the second PC, one can remove the input cable 1 and fix the second input cable to the same PC's printer port connector. In actual use, however, the 2 input cables would be permanently connected to the printer ports of 2 PC's.

From the following PC enter the following to run a buffer testing procedure.

Enter " Copy con Prn " ; press return

Then enter any short message for printing. Enter F6 and return

```
  COPY CON PRN
This is a short printer message from the first PC.
  Z
```

The power is made high for the printer buffer multiplexer unit. Pressing the return key it should pass the message to the buffer by turning LED 1 on. The message ¢ 1 file(s) Copied ¢ is seen on the screen.

Ready with the insertion of the paper in the printer, and with ONLINE light glowing, the printer would now print the above short messge, after a few seconds. Pressing F6 inserts a Ctrl-Z to show on the screen, but that would not be printed.

If the message does not go to the buffer, the "1file(s) copied" message would not come on the screen instead, the PC, after waiting 30 seconds, will display a "Printer not ready" message on screen. This means that is something wrong in the R-S flip flop which sends the

BUST signal to the PC. If this is stuck high always, such an error may appear. Therefore, ICs 13 and 18, used for the Flip flop (BUSY _ ACK) should be checked and replaced.

After the buffer and printer form one PC is tested, the same can be done with the second, by connecting to the second input cable from the PC's printer port.

The word star program is now run. A page of text is typed and saved. The same can be now printed by Ctrl-K-P command and pressing Esc Key. The buffer would then receive this text and start printing. While printing from word star, the " Printer not ready " message does not come and the program simply waits. Like this, printer management routines may vary from one application of software to another, but the buffer will handle to all of them just as if a printer was directly connected to either PC.

Thus after sending the text the buffer, in a few seconds, it starts to print. While it is printing, try entering another matter for printing. This is done by going back to DOS first and then typing COPY CON PRN message program as before. The " 1 file(s) copied" message on the screen in forms the user that the buffer has received it, while previous text is being routed to the printer.

After printing the first text, the second matter goes to the printer. Try entering COPY CON PRN again and again while printing the next text matter. The beeb sounds is produced indicating that both buffers are full. Thus among the 2 buffers allocated, at least one should be free for further entry. " Printer not Ready " message would then ppear on the screen but soon after the buffer is emptied into the printer, it will be accepted.

.3.3 OPERATION OF PRINTER BUFFER

Printer buffer gets data from PC at the same rate as computer output and sends the data to the printer at slow rate. There shouid be proper handshaking, so that the PC should think that a printer is receiving the data. Otherwise the PC's printing software will get out of function and display an error message on the screen; it may not send further data at all. The two signals used are BUSY and ACK.

After a data byte is latched BUSY is generated and kept active high until the latched data is read. Thereafter, the ACK signal is pulsed low to inform the PC that the data has been received by the printer. ICs 13 and 18 (74LS00) are used to form the 2 set - reset flip flops, one for each port. The negative edge of strobe signal sets the flip-flop and generate the BUSY signal that is fed to pin 11 of the D25 connector. This tells the printer port of the PC not to dump any further data.

Now our software looks at the BUSY signal via another input port IC19 (74LS373) in bit D3. Having thus learnt that printing matter is coming from either PC, the input port latched data is read from IC16 (74LS373) and saved.

After saving that data, the ACK signal is generated on port 84 or port 87. This generates a device - select pulse that goes to reset the flip flop formed by IC13 gates (74LS00) or IC18. So that BUSY signal goes low again; the same pulse also serves as the ACK signal for the PC's printer port.

Having saved the data from either PC in the buffer, the output must be routed to the printer. The printer is connected via an output port and another D25 output connector. To handshake with tne actual printer, the strobe signal is to be generated Port 80 is used for written, data from memory gets latched and is available for output from IC20.

The device select port 80 pulse also serves as the strobe signal. The BUSY signal from the printer is read via bit D2 on the input port 83.

Further data is transferred if this goes low. The printer generates this signal after receiving the byte sent to it. Thus the buffer memory is fully transferred to the printer until the end of the printed matter.

CHAPTER IV

SOFTWARE DETAILS

The software which is to be programmed into the EPROM on the print-buffer-mux PCB is in Z80 code and is based on a certain logic or flow chart illustrated in fig(4).

Two buffer flag are used, one for each buffer, to indicate when the buffer is filled with printing matter. In the flow chart shown in fig(5), at point A, the program checks if any buffer is full. If not, which is true initially, the program enters point c, where in the address at which the new data from any PC has to be stored is evaluated.

For instance, if buffer, is full it would be the other buffer starting at address 8000H. The 2 buffers use 1200H to 7FFF and 8000H to FFFF. Stack is used is 13FF down wards. Thus about 28K is reserved for each buffer. Then the input routine is called, which searches for data from the PC and waits for it. If no data is forth coming, it returns, after waiting a few seconds. This timing permits the wordstar program to dump matter after reading diskette, pages after pages. Then the program jumps to point A. At this point, if there is data that has been stored in the buffer, it would tell that the buffer flag is full for buffer 1. So it branches to point B. There itcalls a routine to find the buffer that is full and needs printing which at present, would be 1200H. This is first buffer.

Then it calls printing routine(Table - II) which handshakes with the printer. This print routine is only for printing one byte from the buffer and is repeatedly called until there is no more buffer data, which is known by finding 00 in the memory which is cleared before filling itfrom the PC. The address 1010 is an indirect address, where the buffer current address is stored. If, after exit from this routine, there is no carry, then it indicates that the buffer flag is cleared by going to point D and the routine jumps point A, after clearing the current buffer.

If still there is data to be printed, then a check on inputs is made which will check if data is awaiting from any PC. Any such call from the PC is attended to, by storing in the alternate buffer and its flag is also set. After servicing that PC, the print routine is entered again.

At point E where there is data from the PC's, the store address is found for that fresh data. If both buffers are full, the data has to wait. So a beep sound is called. This tells the PC user to wait soon after finishing the previous print out the data will be received, because then the program will clear the buffer flag of the buffer that was printed out, clear that buffer and reach point A, where input data will be acquired again.

TABLE II

PRINT SUBROUTINE TO PRINT THE DATA IS

```
LD HL, (1010)              -  Load the HL registers with buffer
                              start address
LD DE, 0000H               -  DE registers for timing count.
P1:IN83                    -  Read the busy bit of printer.
AND04                      -  Mask the D2 bit
JPNZ TIME OUT CHK          _  If bit shows busy, wait for while
LD A, (HL)                 -  Get buffer data
OR A                       -  OR operator of A with A to set flags
JZ PRTOVR
OUT80                      -  Output to printer port
INC HL                     -  Point to next data
CALL DELAY                 -  To allow printer to ack data.
LD(1010), HL               -  Save back current buffer.
SCF                        -  Carry is set 17 buffer stil has data
RET
PRTOVR: LD(1010), HL
XOR A                      -  CArry is cleared if printing over
RET
TIMEOUT CHK: DEC DE        _  If printer is busy too long, sound
                              beeb.
```

```
LD A,D
OR A,E
JP NZ PI
BEEP: LD A 01 H            -  Sound beep until printer ready.

OUT 81
IN 83
AND 04 H
JP NZ BEEP
LD A, 00 H
OUT 81
RET

DELAY: IN 83
AND 02
RET Z                      -  If printer acknowledges, Return
IN 83
AND 80                     -  Check for printer error;
CALL NZ BEEP - Z
JP DELAY                   -  If not delay until ack goes low.
BEEP-2 A                   -  Routine to should 2 beeps indicating
                              printing error.
```

PRINTER MULTIPLEXER

```
You chose the type EPROM =2732A( Programming @ 21.5V)______________________
---------------------------------------------------------------------------

  ØØØØ     C3 79 ØF FF FF FF FF FF  C3 Ø8 Ø5 FF FF FF FF FF  Cy.C..
           FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ2Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ3Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ4Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ5Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ6Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ7Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ8Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØ9Ø     FF FF FF FF FF C3 95 ØF  FF FF FF FF FF FF FF FF  C..
  ØØAØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  ØØBØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF F5  u
  ØØCØ     D5 11 3Ø Ø9 1B 7A B3 C2  C4 ØØ D1 F1 C9 FF BB ØØ  U.Ø..z3BD.QqI;.
  ØØDØ     C1 C9 FF FF FE 43 C2 59  ØØ F1 E3 CD DF ØØ C9 F5  AI~CBY.qcM_.Iu
  ØØEØ     3A 1Ø 13 47 3A 11 13 4F  3A 12 13 57 3A 13 13 5F  :..G:..O:..W:.._
  ØØFØ     3A 14 13 67 3A 15 13 6F  F1 C9 FF FF FF FF FF FF  :..g:..oqI

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the typo EPROM =2732A( Programming @ 21.5V)
---------------------------------------------------------------------------

  Ø1ØØ     31 FF 13 21 ØØ 1Ø 4E CD  5Ø Ø1 FE 4Ø D2 1C Ø1 47  1.!..NMP.~@R..G
  Ø11Ø     79 17 17 17 17 E6 FØ BØ  4F C3 Ø7 Ø1 FE 42 C2 25  y....fpØOC..~BB%
  Ø12Ø     Ø1 69 C3 Ø6 Ø1 FE 41 C2  2E Ø1 61 C3 Ø6 Ø1 FE 44  .iC..~AB..aC..~D
  Ø13Ø     C2 37 Ø1 2B C3 Ø6 Ø1 FE  47 C2 41 Ø1 71 23 C3 Ø6  B7.+C..~GBA.q#C.
  Ø14Ø     Ø1 FE 43 C2 Ø6 Ø1 3E F6  D3 Ø2 3E ØØ D3 Ø1 E9 ØØ  .~CB..>vS.>.S.i.
  Ø15Ø     DB Ø2 B7 FA 5Ø Ø1 CD BF  ØØ CD 7Ø Ø1 DB Ø2 B7 F2  [.7zP.M?.Mp.[.7r
  Ø16Ø     59 Ø1 CD BF ØØ DB Ø2 B7  F2 59 Ø1 EE CØ E6 4F C9  Y.M?.[.7rY.n@fOI
  Ø17Ø     C5 F5 7C E6 FØ ØF ØF ØF  ØF 5F 3E 7F D3 Ø1 CD C3  Eu¦fp...._>S.MC
  Ø18Ø     Ø1 7C E6 ØF 5F 3E BF D3  Ø1 CD C3 Ø1 7D E6 FØ ØF  .¦f._>?S.MC.}fp.
  Ø19Ø     ØF ØF ØF 5F 3E EF D3 Ø1  CD C3 Ø1 7D E6 ØF 5F 3E  ..._>oS.MC.}f._>
  Ø1AØ     77 D3 Ø1 CD C3 Ø1 79 E6  FØ ØF ØF ØF ØF 5F 3E FD  wS.MC.yfp...._>}
  Ø1BØ     D3 Ø1 CD C3 Ø1 79 E6 ØF  5F 3E FE D3 Ø1 CD C3 Ø1  S.MC.yf._>~S.MC.
  Ø1CØ     F1 C1 C9 16 Ø1 3E FØ B3  5F 1A D3 Ø2 C3 DB Ø1 C9  qAI..>p3_.S.C[.I
  Ø1DØ     F5 D5 1E ØE 1D C2 D4 Ø1  D1 F1 C9 CD DØ Ø1 AF D3  uU...BT.QqIMP./S
  Ø1EØ     Ø2 C9 FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  .I
  Ø1FØ     7E ØC B6 9E CC DA FA ØE  FE CE EE F8 72 BC F6 E2  ~.6.LZz.~Nnxr<vb
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
--------------------------------------------------------------------------------

 Ø2ØØ     21 ØØ 2Ø DB 83 E6 Ø8 CA  Ø3 Ø2 DB 82 77 23 D3 84  !. [.f.J..[.w#S.
 Ø21Ø     D3 Ø4 C3 Ø3 Ø2 FF FF FF  FF FF FF FF FF FF FF FF  S.C..
 Ø22Ø     21 ØØ 2Ø DB 83 E6 Ø4 C2  23 Ø2 7E B7 CA 3A Ø2 D3  !. [.f.B#.~7J:.S
 Ø23Ø     8Ø D3 Ø4 23 CD DØ Ø1 C3  23 Ø2 C7 FF FF FF FF FF  .S.#MP.C#.G
 Ø24Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø25Ø     21 ØØ 2Ø 3E ØØ 77 23 7C  FE 8Ø C2 53 Ø2 C9 FF FF  !. >.w#¦~.BS.I
 Ø26Ø     21 ØØ 8Ø 3E ØØ 77 23 7C  FE FF C2 63 Ø2 C9 FF FF  !..>.w#¦~Bc.I
 Ø27Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø28Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø29Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø2AØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø2BØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø2CØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø2DØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø2EØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø2FØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
--------------------------------------------------------------------------------

 Ø3ØØ     CD 5Ø Ø2 CD 6Ø Ø2 3E ØØ  D3 81 Ø1 ØØ ØØ ØØ ØØ ØØ  MP.M'.>.S.......
 Ø31Ø     ØØ ØØ CD 8Ø ØE CD CØ ØD  D2 15 Ø3 CD EØ Ø5 C7 FF  ..M^.M@.R..M'.G
 Ø32Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø33Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø34Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø35Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø36Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø37Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø38Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø39Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø3AØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø3BØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø3CØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 Ø3DØ     CD 5Ø Ø2 Ø1 ØØ ØØ 78 CD  8Ø ØE CD 8Ø ØD D2 DA Ø3  MP....xM..M..RZ.
 Ø3EØ     CD EØ Ø4 C7 FF FF FF FF  FF FF FF FF FF FF FF FF  M'.G
 Ø3FØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
----------------------------------------------------------------------------------


Ø4ØØ    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø41Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø42Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø43Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø44Ø    FF FF FF FF FF DB 83 E6  Ø1 C3 49 Ø5 FF FF FF FF   [.f.CI.
Ø45Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø46Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø47Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø48Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø49Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø4AØ    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø4BØ    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø4CØ    3E 3F 3D C2 C2 Ø4 C3 45  Ø5 FF FF FF FF FF FF FF   >?=BB.CE.
Ø4DØ    3E 3F 3D C2 D2 Ø4 C3 ED  Ø4 FF FF FF FF FF FF FF   >?=BR.Cm.
Ø4EØ    78 E7 F8 ØØ 11 ØØ ØØ 79  FE Ø1 C2 45 Ø5 DB 83 E6   x7x....y~.BE.[.f
Ø4FØ    Ø8 CA FF Ø4 DB 82 77 CD  6Ø Ø5 D3 84 C3 ED Ø4 1B   .J.[.wM'.S.Cm..

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
----------------------------------------------------------------------------------


Ø5ØØ    7A B3 CA Ø8 ØØ C3 DØ Ø4  7C FE 8Ø D2 1F Ø5 7C D6   z3J..CP.¦~.R..¦V
Ø51Ø    2Ø 57 5D EB 22 FØ 1Ø EB  3E Ø1 BØ 47 C3 2C Ø5 D6    W]k"p.k>.ØGC,.V
Ø52Ø    8Ø 57 5D EB 22 F2 1Ø EB  3E Ø2 BØ 47 7A B3 CA 33   .W]k"r.k>.ØGz3J3
Ø53Ø    Ø5 ØØ C9 3E 8Ø BC DA 3F  Ø5 3E FE AØ 47 ØØ C9 3E   ..I>.<Z?.>~ G.I>
Ø54Ø    FD AØ 47 ØØ C9 C3 45 Ø4  ØØ CA 57 Ø5 DB 86 77 CD   } G.ICE..JW.[.wM
Ø55Ø    6Ø Ø5 D3 87 C3 45 Ø5 1B  7A B3 CA Ø8 Ø5 C3 45 Ø5   '.S.CE..z3J..CE.
Ø56Ø    7C FE 8Ø 23 DA 72 Ø5 7C  FE ØØ CA 7D Ø5 3E Ø4 D3   ¦~.#Zr.¦~.J}.>.S
Ø57Ø    81 C9 7C FE 8Ø CA 7D Ø5  3E Ø2 D3 81 C9 2B F1 C3   .I¦~.J}.>.S.I+qC
Ø58Ø    ØE Ø5 FF FF FF FF FF FF  FF FF FF FF FF FF FF FF   ..
Ø59Ø    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø5AØ    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø5BØ    FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
Ø5CØ    21 ØØ 2Ø DB 83 E6 Ø8 CA  C3 Ø5 DB 82 77 23 ØØ ØØ   !. [.f.JC.[.w#..
Ø5DØ    ØØ D3 84 C3 C3 Ø5 FF FF  FF FF FF FF FF FF FF FF   .S.CC.
Ø5EØ    ØØ ØØ ØØ ØØ 11 ØØ ØØ 79  FE Ø1 C2 45 Ø5 DB 83 E6   .......y~.BE.[.f
Ø5FØ    Ø8 CA FF Ø5 DB 82 77 CD  6Ø Ø6 D3 84 C3 ED Ø5 1B   .J.[.wM'.S.Cm..

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
-----------------------------------------------------------------------------


  Ø600     7A B3 CA Ø8 Ø6 C3 ED Ø5  7C FE 8Ø D2 1F Ø6 7C D6   z3J..Cm.¦~.R..¦V
  Ø61Ø     2Ø 57 5D EB 22 FØ 1Ø EB  3E Ø1 BØ 47 C3 2C Ø6 D6    W]k"p.k>.ØGC,.V
  Ø62Ø     8Ø 57 5D EB 22 F2 1Ø EB  3E Ø2 BØ 47 7A B3 CA 33   .W]k"r.k>.ØGz3J3
  Ø63Ø     Ø6 ØØ C9 3E 8Ø BC DA 3F  Ø6 3E FE AØ 47 ØØ C9 3E   ..I>.<Z?.>~ G.I>
  Ø64Ø     FD AØ 47 ØØ C9 DB 83 E6  Ø1 CA 57 Ø6 DB 86 77 CD   } G.I[.f.JW.[.wM
  Ø65Ø     6Ø Ø6 D3 87 C3 45 Ø6 1B  7A B3 CA Ø8 Ø6 C3 45 Ø6   '.S.CE..z3J..CE.
  Ø66Ø     7C FE 8Ø 23 DA 72 Ø6 7C  FE ØØ CA 7D Ø6 3E Ø4 D3   ¦~.#Zr.¦~.J}.>.S
  Ø67Ø     81 C9 7C FE 8Ø CA 7D Ø6  3E Ø2 D3 81 C9 C3 AB ØE   .I¦~.J}.>.S.IC+.
  Ø680     78 FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF   x
  Ø69Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø6AØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø6BØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø6CØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø6DØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø6EØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø6FØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

0700 Page free

```
You chose the type EPROM =2732A( Programming @ 21.5V)
-----------------------------------------------------------------------------


  Ø8ØØ     3E AA D3 8Ø DB 81 DB 82  DB 83 3E ØF D3 81 C3 ØØ   >*S.[.[.[.>.S.C.
  Ø81Ø     ØØ FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF   .
  Ø82Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø83Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø84Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø85Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø86Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø87Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø88Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø89Ø     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø8AØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø8BØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø8CØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø8DØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø8EØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
  Ø8FØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

[illegible] 09, 0A free

```
You chose the type EPROM =2732A( Programming @ 21.5V)
----------------------------------------------------------------------------------


 ØBØØ    CD 5Ø Ø2 CD 6Ø Ø2 3E ØØ   D3 81 Ø1 ØØ ØØ ØØ ØØ ØØ   MP.M'.>.S.......
 ØB1Ø    ØØ ØC CD 8Ø ØE CD CØ ØD   D2 15 ØB CD EØ ØD C7 FF   ..M..M@.R..M'.G
 ØB2Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB3Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB4Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB5Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB6Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB7Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB8Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØB9Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØBAØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØBBØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØBCØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØBDØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØBEØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØBFØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF


^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
You chose the type EPROM =2732A( Programming @ 21.5V)
----------------------------------------------------------------------------------


 ØCØØ    CD 5Ø Ø2 CD 6Ø Ø2 3E ØØ   D3 81 Ø1 ØØ ØØ 78 B7 C2   MP.M'.>.S....x7B
 ØC1Ø    21 ØC CD 8Ø ØE CD CØ ØD   D2 15 ØC CD EØ ØD C3 ØD   !.M..M@.R..M'.C.
 ØC2Ø    ØC CD BØ ØE CD ØØ ØF D2   39 ØC CD CØ ØD D2 24 ØC   .MØ.M..R9.M@.R$.
 ØC3Ø    CD 8Ø ØE CD EØ ØD C3 24   ØC 79 FE Ø1 CA 44 ØC FE   M..M'.C$.y~.JD.~
 ØC4Ø    Ø2 CA 4E ØC 3E 7E AØ 47   CD 5Ø Ø2 C3 ØD ØC 3E 7D   .JN.>~ GMP.C..>}
 ØC5Ø    AØ 47 CD 6Ø Ø2 C3 ØD ØØ   C3 57 ØD FF FF FF FF FF    GM'.Ç..CW.
 ØC6Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØC7Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
 ØC8Ø    CD 5Ø Ø2 CD 6Ø Ø2 3E ØØ   D3 81 Ø1 ØØ ØØ 78 B7 C2   MP.M'.>.S....x7B
 ØC9Ø    A1 ØC CD 8Ø ØE CD CØ ØD   D2 95 ØC CD EØ ØC C3 8D   !.M..M@.R..M'.C.
 ØCAØ    ØC CD BØ ØE CD ØØ ØF D2   B9 ØC CD CØ ØD D2 A4 ØC   .MØ.M..R9.M@.R$.
 ØCBØ    CD 8Ø ØE CD EØ ØC C3 A4   ØC 79 FE Ø1 CA C4 ØC FE   M..M'.C$.y~.JD.~
 ØCCØ    Ø2 CA CE ØC 3E 7E AØ 47   CD 5Ø Ø2 C3 8D ØC 3E 7D   .JN.>~ GMP.C..>}
 ØCDØ    AØ 47 CD 6Ø Ø2 C3 8D ØC   FF FF FF FF FF FF FF FF    GM'.C..
 ØCEØ    D5 CD EØ ØD D1 C9 FF FF   FF FF FF FF FF FF FF FF   UM'.QI
 ØCFØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF


^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
---------------------------------------------------------------------------

ØDØØ    78 1F D2 15 ØD 21 ØØ 2Ø   22 1Ø 1Ø 3E Ø1 32 2Ø 1Ø   x.R..!. "..>.2 .
ØD1Ø    2A FØ 1Ø EB C9 1F D2 29   ØD 21 ØØ 8Ø 22 1Ø 1Ø 3E   *p.kI.R).!.."..>
ØD2Ø    Ø2 32 2Ø 1Ø 2A F2 1Ø EB   C9 11 ØØ ØØ C9 FF FF FF   .2 .*r.kI...I
ØD3Ø    3E ØF ØØ ØØ EB EB 3D ØØ   ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ   >...kk=........
ØD4Ø    ØØ ØØ ØØ C2 32 ØD 1B 7A   B3 CA Ø8 ØE C3 ED ØD FF   ...B2..z3J..Cm.
ØD5Ø    FF FF FF FF FF FF FF 3E   7F ØØ ØØ EB EB 3D C2 59   >..kk=BY
ØD6Ø    ØD 1B 7A B3 CA Ø8 ØE C3   45 ØE FF FF FF FF FF FF   ..z3J..CE.
ØD7Ø    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
ØD8Ø    DB 83 E6 Ø9 CA 9Ø ØD FE   Ø8 C2 94 ØD ØE Ø1 37 C9   [.f.J..~.B....7I
ØD9Ø    AF ØE ØØ C9 ØE Ø2 37 C9   FF FF FF FF FF FF FF FF   /..I..7I
ØDAØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
ØDBØ    3E 7F ØØ ØØ EB EB 3D C2   B2 ØD 1B 7A B3 CA Ø8 ØE   >..kk=B2..z3J..
ØDCØ    1A Ø3 ØØ ØØ CA CØ ØD C2   ØØ ØØ 1Ø Ø8 Ø2 ØØ ØØ Ø8   ....J@.B........
ØDDØ    83 ØC ØØ C9 ØE Ø2 37 C9   FF FF FF FF FF FF FF FF   ...I..7I
ØDEØ    78 B7 F8 ØØ 11 ØØ ØØ 79   FE Ø1 C2 45 ØE DB 83 E6   x7x....y~.BE.[.f
ØDFØ    Ø8 CA 3Ø ØD DB 82 77 CD   6Ø ØE D3 84 C3 ED ØD 1B   .JØ.[.wM'.S.Cm..

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
You chose the type EPROM =2732A( Programming @ 21.5V)
---------------------------------------------------------------------------

ØEØØ    7A B3 CA Ø8 ØE C3 ED ØD   7C FE 8Ø D2 1F ØE 7C D6   z3J..Cm.¦~.R..¦V
ØE1Ø    2Ø 57 59 EB 22 FØ 1Ø EB   3E Ø1 BØ 47 C3 2C ØE D6    WYk"p.k>.ØGC,.V
ØE2Ø    8Ø 57 5D EB 22 F2 1Ø EB   3E Ø2 BØ 47 7A B3 CA 33   .W]k"r.k>.ØGz3J3
ØE3Ø    ØE ØØ C9 3E 8Ø BC DA 3F   ØE 3E FE AØ 47 ØØ C9 3E   ..I>.<Z?.>~ G.I>
ØE4Ø    FD AØ 47 ØØ C9 DB 83 E6   Ø1 CA 57 ØC DB 86 77 CD   } G.I[.f.JW.[.wM
ØE5Ø    6Ø ØE D3 87 C3 45 ØE 1B   7A B3 CA Ø8 ØE C3 45 ØE   '.S.CE..z3J..CE.
ØE6Ø    7C FE 8Ø 23 DA 72 ØE 7C   FE ØØ CA 7D ØE 3E Ø4 D3   ¦~.#Zr.¦~.J}.>.S
ØE7Ø    81 C9 7C FE 8Ø CA 7D ØE   3E Ø2 D3 81 C9 C3 AB ØE   .I¦~.J}.>.S.IC+.
ØE8Ø    78 1F DA 8B ØE 21 ØØ 2Ø   C3 9E ØE 1F DA 95 ØE 21   x.Z..!. C...Z..!
ØE9Ø    ØØ 8Ø C3 9E ØE 3E Ø1 D3   81 3E 8Ø BØ 47 C9 78 E6   ..C..>.S.>.ØGIxf
ØEAØ    7F 47 3E ØØ D3 81 C9 FF   FF FF FF 2B F1 C3 Ø8 Ø6   G>.S.I+qC..
ØEBØ    78 1F D2 C2 ØE 21 ØØ 2Ø   22 1Ø 1Ø ØE Ø1 2A FØ 1Ø   x.RB.!. "....*p.
ØECØ    EB C9 1F D2 D3 ØE 21 ØØ   8Ø 22 1Ø 1Ø ØE Ø2 2A F2   kI.RS.!.."....*r
ØEDØ    1Ø EB C9 11 ØØ ØØ C9 FF   FF FF FF FF FF FF FF FF   .kI...I
ØEEØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF
ØEFØ    FF FF FF FF FF FF FF FF   FF FF FF FF FF FF FF FF

^ = PAGE UP  ESC = EXIT  ANY OTHER KEY = PAGE DOWN
```

```
Yo[illegible]e the type EPROM =2732A( Programming @ 21.5V)
------- ----------------------------------------------------------------------

 ØFØØ     2A 1Ø 1Ø DB 83 E6 Ø4 C2  Ø3 ØF 7A B3 CA 24 ØF 7E  *..[.f.B..z3J$.~
 ØF1Ø     D3 8Ø 3E Ø8 D3 81 23 CD  DØ Ø1 22 1Ø 1Ø 1B 37 3E  S.>.S.#MP."...7>
 ØF2Ø     ØØ D3 81 C9 AF C9 DB 83  E6 8Ø CA 37 ØF DB 83 E6  .S.I/I[.f.J7.[.f
 ØF3Ø     4Ø C2 53 ØF C3 Ø3 ØF 3E  Ø1 D3 81 3E 32 CD BF ØØ  @BS.C..>.S.>2M?.
 ØF4Ø     3D C2 3D ØF 97 D3 81 3E  32 CD BF ØØ 3D C2 49 ØF  =B=..S.>2M?.=BI.
 ØF5Ø     C3 26 ØF 3E Ø1 D3 81 3E  64 CD BF ØØ 3D C2 59 ØF  C&.>.S.>dM?.=BY.
 ØF6Ø     97 D3 81 3E 64 CD BF ØØ  3D C2 65 ØF 97 D3 81 C3  .S.>dM?.=Be..S.C
 ØF7Ø     26 ØF FF FF FF FF FF FF  FF 31 FF 13 D3 84 D3 87  &.1.S.S.
 ØF8Ø     CD 5Ø Ø2 CD 6Ø Ø2 3E ØØ  D3 81 Ø1 ØØ ØØ 78 B7 C2  MP.M'.>.S....x7B
 ØF9Ø     A1 ØF CD 8Ø ØE CD 8Ø ØD  D2 95 ØØ CD EØ Ø4 C3 8D  !.M..M..R..M'.C.
 ØFAØ     ØF CD ØØ ØD CD ØØ ØF D2  B9 ØF CD 8Ø ØD D2 A4 ØF  .M..M..R9.M..R$.
 ØFBØ     CD 8Ø ØE CD EØ Ø4 C3 A4  ØF 3A 2Ø 1Ø FE Ø1 CA C6  M..M'.C$.: .~.JF
 ØFCØ     ØF FE Ø2 CA DØ ØF 3E 7E  AØ 47 CD 5Ø Ø2 C3 8D ØF  .~.JP.>~ GMP.C..
 ØFDØ     3E 7D AØ 47 CD 6Ø Ø2 C3  8D ØF FF FF FF FF FF FF  >} GM'.C..
 ØFEØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
 ØFFØ     FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF

^ = PAGE
```

SOURCE CODE LISTING

PRINTER BUFFER PROGRAM: (For two PCs to one Printer)

```
00 00 C3 79 OF JMP OF 79

00 08 C3 08 05 JMP 08
```

0.5m Sec delay Routine:

```
01   D0   F5               PUSH PSW
     D1   D5               PUSH D
     D2   IE 0E            MVIE OE
     D4   ID           D2: DCRE
     D5   C2 D4 01         JNZ D2
     D8   D1               POP D
     D9   F1               POP PSW
     DA   C9               RET
```

10m Sec delay Routine

```
00   BF   F5               PUSH PSW
     C0   D5               PUSH D
     C1   11 30 09         LXID 09 30
     C4   1B           D1: DCXD
     C5   7A               MOV D,A
     C6   B3               ORAE
     C7   C2 C4 00         JNZ D1
     CA   D1               POP D
     CB   F1               POP PSW
     CC   C9               RET
```

```
OF  79   31 PF 13       LXIP 13 FF

OF  7C   D3 84          OUT 84

    7E   D3 87          OUT 87

    80   CD 50 02       CALL CLEAR BUFFER 1

    83   CD 60 02       CALL CLEAR BUFFER 2

    86   3E 00          MVIA 00
                                  CLEAR LIGHTS AND SOUNDS
    88   D3 81          OUT 81

    8A   01 00 00       LXIB 00 00  Flags  =  0,  C  =  0, B = 0

                                    C=Printing Flag
                                    C=0 0.1 for printing buffer 2
                                    B=Buffer full flag
                                    B,01=Buffer 1 full
                                    B=02, Buffer 2 full
                                    B=03, Both Buffers full


     8D   78                A: MOV A,B

     3E   B7                ORAA

     8F   C2 A1 OF          JNZ B:

     92   CD 80 OE          CALL set store Addr.
                            Find either 2000 or 8000 as address

     95   CD 80 OD A1       CALL I/P Check

     98   D2 95 OF          JNC A1

     9B   CD E0 04          CALL I/P

     9E   C3 8D OF          JMP A:

      A1   CD 00 OD      B:   CALL Set printing Addr.

      A4   CD 00 OF      B1:  CALL PRINT

      A7   D2 B9 OF      JNC D:          Go  to  clear  buffer  and
                                         after  printing  is  over.

      AA   CD 80 OD      CALL I/P Check

           AD D2 A4 OF   JNC B1

           B0 CD 80 OE   CALL  Set store addr.
                                         Set  buffer  address  for
                                         I/P, if buffer isa available
                                         otherwise beeps go to B1
```

```
Cf   B3   CD E0 04          CALL I/P data
     B6   C3 A4 OF          JMP B1

     B9   3A 20 10          LDA 10 20 Check Printing flag

     BB   FE 01 C6 OF       CPI 01

     BE   CA C6 OF          JZ D1

     C1   FE 02             CPI 02

     C3   CA DO OF          JZ D2

     C6   3E 7E        D1   MVIA 7E

     C8   AO                ANA  B

     C9   A7                MOV B,A
                                                   Clears do bit for B
     CA   CD 50 02          CALL Clear Buffer 1     Register

     CD   C3 8D OF          JMP A:

     D0   3E 7D        D2   MVIA 7D

     D2   A0                ANA B

     D3   A7                MOV B,A           Clears do bit for B'Register

     D4   CD 60 02          CALL Clear buffer 2

     D7   C3 8D OF          JMP A
```

I/F Check Routine:

```
OD   80   DB 83             IN 83

     82   E6 09             ANI 09

     84   CA 90 OD          JZ  no data

     87   FE 08             CPI 08

     89   C2 94 OD          JNZ  PC2

     8C   OE 01        PCI  MVIC 01      (PC1 requests data for printing)

     8E   37                STC

     8F   C9                RET
```

```
OD  90   AF              No data  XRAA      Clear  carry  if  no  data
                                            from  either  PC.

         OE 00           MVIC 00

         C9              RET

         OE 02 PC2:      MVIC 02        Let  C  =  2  PC2  sends  data.

         37              STC

         C9              RET
```

INPUT ROUTINE:

```
04   E0   78               MOV A,B

     E1   B7               ORAA

     E2   F8               RM

     E3   OO               NOP

     E4   11 00 00         LXID 00 00  This is time limit counter.

     E7   79               MOV A,C

     E8   FE 01            CPI 01      PC1 need I/P ?

     EA   C2  45 05        JNZ PC2     no, go to Pt. PC2:

     ED   DB 83      PC1:  IN 83       Reads  I/P  port  strobe  signal

     EF   E6 08            ANI 08

     F1   CA FF 04         JZ  Wait

     F4   DB 82            IN 82       Read  data

     F6   77               MOV M,A     Read  data  and  store  in  (HL)

     F7   CD 60 05         CALL memory Increment
```

```
04  FA  D3 84          OUT 84
    FC  C3 ED 04       JMP PC1
    FF  1B             Wait:  DCXD
    00  7A             MOV A,D
    01  B3             ORAE
    02  CA 08 00       JZ EXIT      Can't wait too long. no more
                                    data, therefore exit.
    05  C3 D0 04       JMP PC1
    08  7C             EXIT:  MOV A,H
    09  FE 80          CPI 80
    OB  D2 IF 05       JNC Buffer 2
05  OE  7C             Buffer  MOV A,H
    OF  D6 20          SUI 20
    11  57             MOV D,A
    12  5D             MOV E,L     Save no. of bytes in D-E
    13  EB             XCHG
    14  22 FO 10       SHLD 10 FO Store  no.  of  bytes  in  10FO,
                                  10F1
    17  EB             XCHG
    18  3E 01          MVIA 01
        B0             ORAB
        47             MOV B,A
        C3 2C 05       JMP no data clock
    1F  D6 80          Buffer 2:  SUI 80
    21  57             MOV D,A
    22  5D             MOV E,L
    23  EB             XCGH
```

```
05   24   22 F2 10          SHLD  10 F2     Store   data   bytes   for
                                            buffer  2  in  10F2,  10F3

     27   EB                XCHG

     28   3E 02             MVIA 02

     2A   B0                ORA B

     2B   47                MOV B,A

     2C   7A                MOV A,D   No data clock

     2D   B3                ORAE

     2E   CA 33 05          JZ to clear flag

     31   00                NOP

     32   C9                RET

     33   3E 80             MVIA 80   Clear   the   buffer   according
                                      as buffer 1 or 2

     35   BC                CHPH

     36   DA 3F 05          JC

     39   3E FE             MVIA FE

     3B   A0                ANA B     Do bit is cleared

     3C   47                MOV B,A

     3D   00                NOP

     3E   C9                RET

     3F   3E FD         N:  MVIA F D

     41   A0                ANA  B    D1 bit is cleared

     42   47                MOV  B,A

     43   00                NOP

     44   C9                RET

     45   DB 83       PC2:  IN 83

     47   E6 01             ANI 01

     49   CA 57 05          JZ   wait 2:
```

```
05   4C   DB 86          IN 86
     4E   77             MOV M,A
     4F   CD 60 05       CALL  Memory increment.
     52   D3 87          OUT 87
     54   C3 45 05       JMP PC2
     57   1B             wait 2:  DCX D
     58   7A             MOV A,D
     59   B3             ORAE
     5A   CA 08 05       JZ EXIT
     5D   C3 45 05       JMP PC2
     60   7C             MOV A,H
          PE 80          CPI 80
          23             INX H
          DA 72 05       JC Buffer 1
          7C             MOV A,H
     68   FE 00          CPI 00
          CA 7 D 05      JZ Buffer full
          3E 04          MVIA 4      Light LED 2
     6F   D3 81          OUT 81
     71   C9             RET
     72   7C             Buffer 1  MOV A,H
     73   FE 80          CPI 80
     75   CA 7D 05       JZ Buffer full Light LED 1
     78   3E 02          MVIA 02
     7A   D3 81          OUT 81
     7C   C9             RET
     7D   2B             DCXH         Buffer full
     7E   F1             POP PSW
     7F   C3 0E 05       JMP EXIT
```

SET STORE ADDRESS

```
OE  80   78              MOV A,B
    81   1F               RAR
    82   DA 8B 0E        JC  Buffer full
    86   21 00 20        LXIH 2000
    88   C3 9E 0E        JMP          Clear overflow & Beep

    8B   1F              RAR

    8C   DA 95 0E        JC   B:

    8F   21 00 80        LXIH 8000

    92   C3 9E 0E        JMP Clear overflow beep

         3E 01           MVIA 01

    97   D3 81           OUT 81       Beep enable

    99   3E 80           MVIA  80

         B0              ORAB         Set D7 bit high if both buffers
                                      are full.
         47              MOV B,A

         C9              RET
    9E   78              MOV A,B      Clear overflow bit & beep

    9F   E6 7F           ANI 7F

    A1   47              MOV B,A

    A2   3E  00          MVIA 00      Clear the sound

    A4   D3 81           OUT 81

    A6   C9              RET
```

SET PRINT ADDRESS:

```
OD  00  78          MOV A,B
        1F          RAR
        D2 15 OD    JNC  Buffer 2
        21  00 20   LXIH 2000
        22 10 10    SHLD 10 10 1010 holds buffer start address.
        3E 01       MVIA 01
        32 20 10    STA 10 20
    10  2A F0 10    LHLD 10 F0
    13  EB          XCHG              Gets the bytes store into
                                      D, E registers
    14  C9          RET
    15  1F          RAR           Buffer 2
        D2 29 OD    JNC           Clock
        21 00 80    LXID 8000
        22 10 10    SHLD 10 10
    1F  3E 02       MVIA 02
        32 20 10    STA 10 20
        2A F2 10    LHLD 10 F2
        EB          XCHG
    28  C9          RET
    29  11 00 00    LXID 00 00     Clock:   Bytes = 0
                                            if no buffer is full
        C9          RET
```

PRINT ROUTINE:

D-E on entry - Number of bytes to print on exit is set, if furter data is to be printed. Prints are byte in one pass. C=1 or 2 tells which buffer printer clock routine can be added if necessary.

```
OF  00  2A 10 10    LHLD 10 10
        DB 83       IN 83        PR. BUSY
        E6 04       ANI 04
    07  C2 03 OF    JNZ          PR. BUSY  OR  JNZ to printer clock
    0A  7A          MOV A,D      By clock
        B3          ORAE
        CA 24 OF    JZ  End of print
    OF  7E          MOV A,M
        D3 80       OUT 80    is printer port
        3E 08       MVIA 08
        D3 81       OUT 81       Let LEd 3 for printer be ON.
    16  23          INX H
    17  CD DO 01    CALL 0.5ms Delay
        22 10 10    SHLD  10 10
        1B          DCXD         Decrements byte count
        37          STC
    1F  3E 00       MVIA 00      Clear LED
    21  D3 81       OUT  81
    23  C9          RET
```

```
OF   24   AF              XRAA         End of print
          C9              RET

PRINTER CHECK:

OF   26   DB 83           IN 83
     28   E6 80           ANI 80
     2A   CA 37 OF        JZ           Printer Error
     2D   DB 83           DB 83
     2F   E6 40           ANI 40
     31   C2 53 OF        JNZ          No Paper
     34   C3 03 OF        JMP          Printer BUSY
     37   3E 01           MVIA         Printer error
     39   D3 81           OUT 81
     3B   3E 32           MVIA 32
     3D   CDBF 00         CALL DLY (10m Sec)
          3D              DCRA         BPH
          C2 3D OF        JNZ          BPH
          97              SUB A
          D3 81           OUT 81
     47   3E 32           MVIA 32
     49   CD BF 00        CALL DLY     BPL
     4C   3D              DCR A
     4D   C2 49 OF        JNZ          BPL
     50   C3 26 DP        JMP Printer Ch.
```

```
OF   53   3E 01          MVIA 01
          D3 81          OUT 81
     57   3E 64          MVIA 64
     59   CD BF 00       CALL  10 ms DLY    Beep H
          3D             DCR A
          C2 59 OF       JNZ Bp. H
     60   97             SUB A
     61   D3 81          OUT 81
     63   3E 64          MVIA 64
     65   CD BF 00       CALL DLY   BPL 2
     68   3D             DCR A
          C2 65 OF       JNZ        BPI 2
          97             SUB A
          D3 81          OUT 81
     6F   C3 26 OF       JMP    Printer check.
```

CLEAR BUFFER MEMORY - 1 ROUTINE:

```
02  50  21 00 20     LHIX 2000
    53  3E 00        MVIA 00
    55  77           MOV M,A
    56  23           INXH
    57  7C           MOV A,H
    58  FE 80        CPI 80
    5A  C2 53 02     JNZ Y
    5D  C9           RET
```

CLEAR BUFFER MEMORY - 2 ROUTINE:

```
02  60  21 00 81     LXIH 8000
    63  3E 00        MVIA 00
    65  77           MOV M,A
    66  23           INXH
    67  7C           MOV A,H
    68  FE FF        CPI FF
    6A  C2 63 02     JNZ Y1
    6D  C9           RET
```

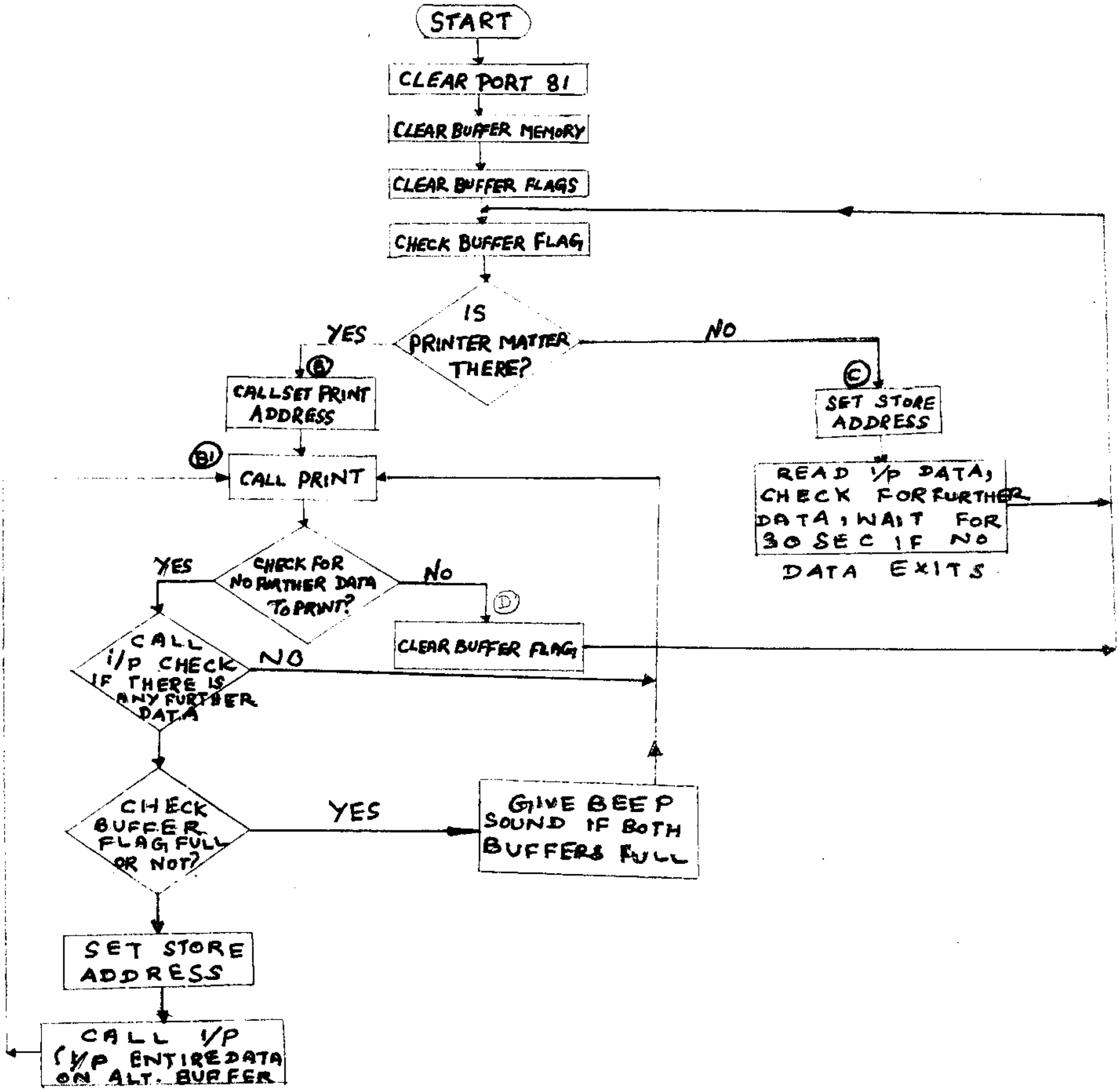

START
CLEAR PORT 81
CLEAR BUFFER MEMORY
CLEAR BUFFER FLAGS
CHECK BUFFER FLAG
IS PRINTER MATTER THERE?
YES
NO
B
CALLSET PRINT ADDRESS
C
SET STORE ADDRESS
READ I/P DATA, CHECK FOR FURTHER DATA, WAIT FOR 30 SEC IF NO DATA EXITS
B1
CALL PRINT
CHECK FOR NO FURTHER DATA TO PRINT?
YES
NO
D
CLEAR BUFFER FLAG
CALL I/P CHECK IF THERE IS ANY FURTHER DATA
NO
CHECK BUFFER FLAG FULL OR NOT?
YES
GIVE BEEP SOUND IF BOTH BUFFERS FULL
SET STORE ADDRESS
CALL I/P (I/P ENTIREDATA ON ALT. BUFFER

FIG (4) FLOW CHART FOR PRINTER BUFFER AND MULTIPLEXER

CHAPTER V

USES AND ENHANCEMENT OF PRINTER BUFFER

5.1 Uses of the Printer Buffer

The most singular and striking advandage of the printer buffer is that it saves computer time, in the sense that, the computer is not blocked with feeding of data necessary to be printed to the printer. Instead, as the data are stored temoorarily in the memory of the buffer and the computer is freed to ooerate on a different task.

The printer buffer has the additional facility of making as many cooies of a file as required by the user.

It may be thought that since the printer ooerates independent of the host comouter after initial storage of the data in to the printer are lost. This is however not have true as printer can be controlled through the buffer by making use of keys suchas Stop, Pause etc..

5.2 Enhancement

The printer buffer is software programmable and hence is operation can be greatly enhanced. There are provisions to add a stop-on-form feed switch for a single sheet printer.

The printer buffer can also be design with Z80 PIO, which enables multiple input and outputs. Interruots can also be used for the same purpose. In such a case the CPU has to run on a faster clock and RAM capacity may be increased. The 8088 CPU is the ideal choice in place of the Z80 CPU under above circumstances.

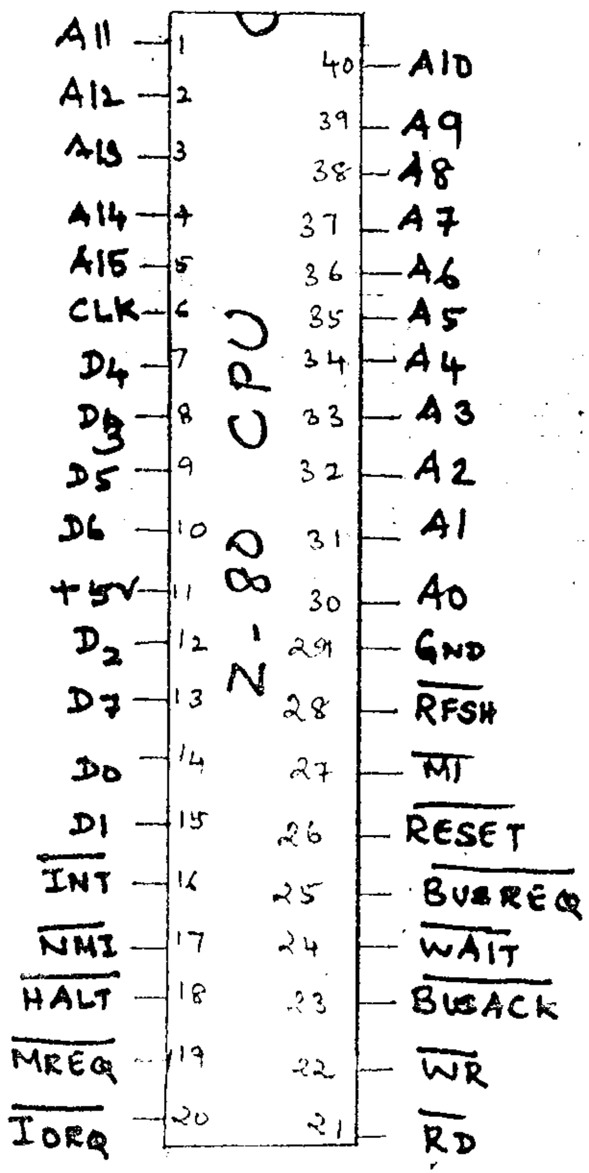
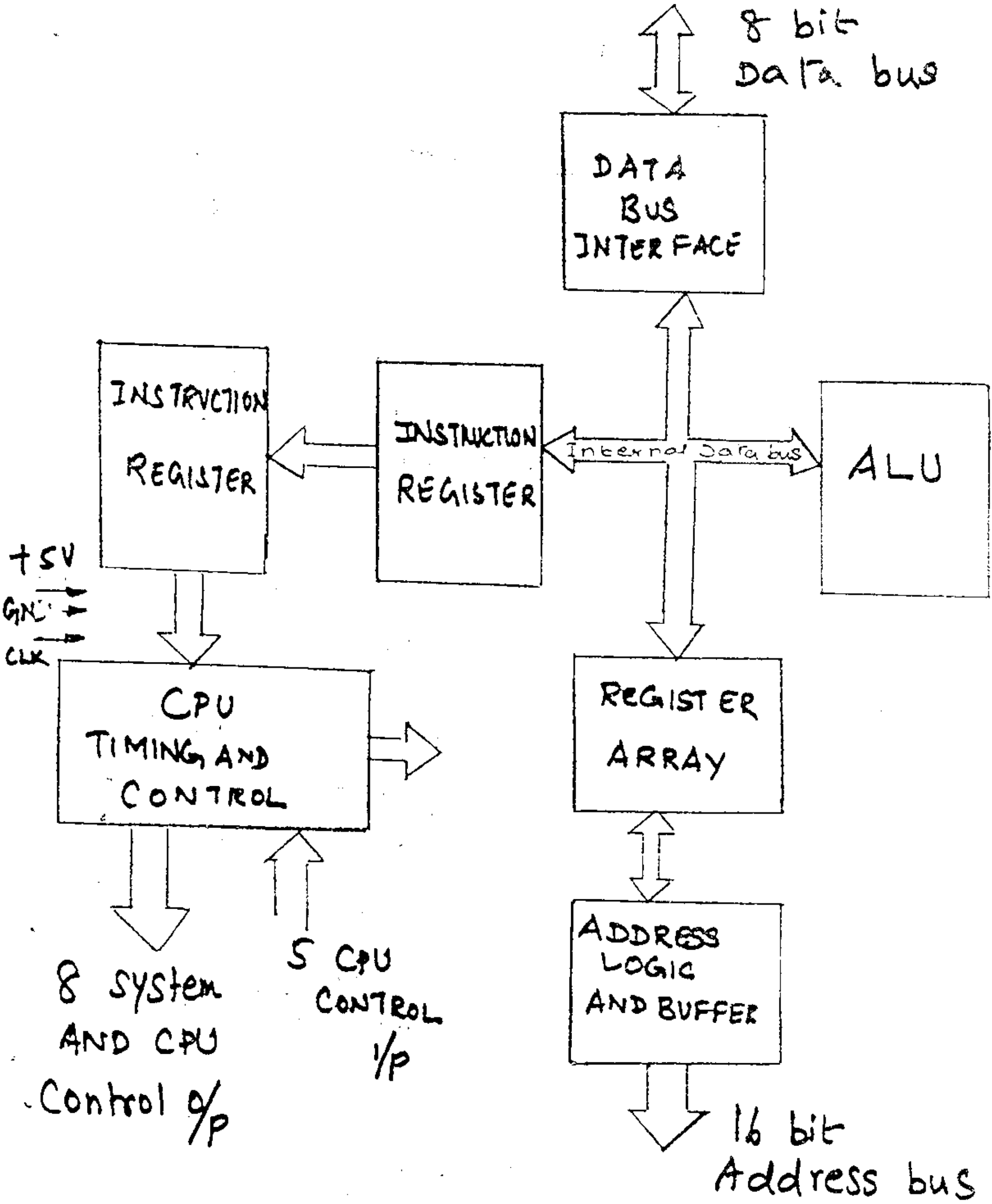


Fig (1.1) PINOUT DETAILS OF Z-80



Fig(2) Z-80 MICROPROCESSOR BLOCK DIAGRAM

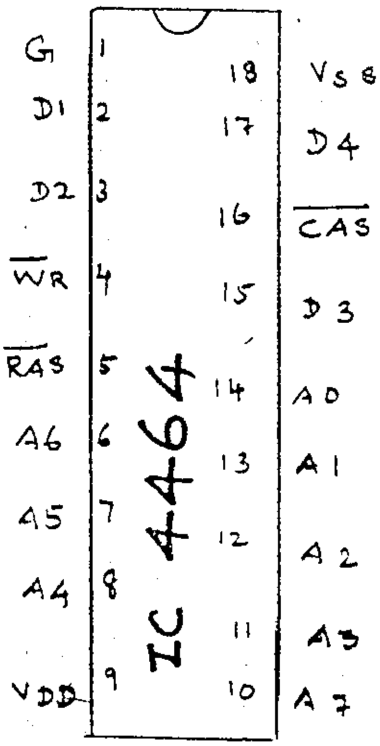


Fig (2.1) PIN OUT DETAILS OF DYNAMIC RAM

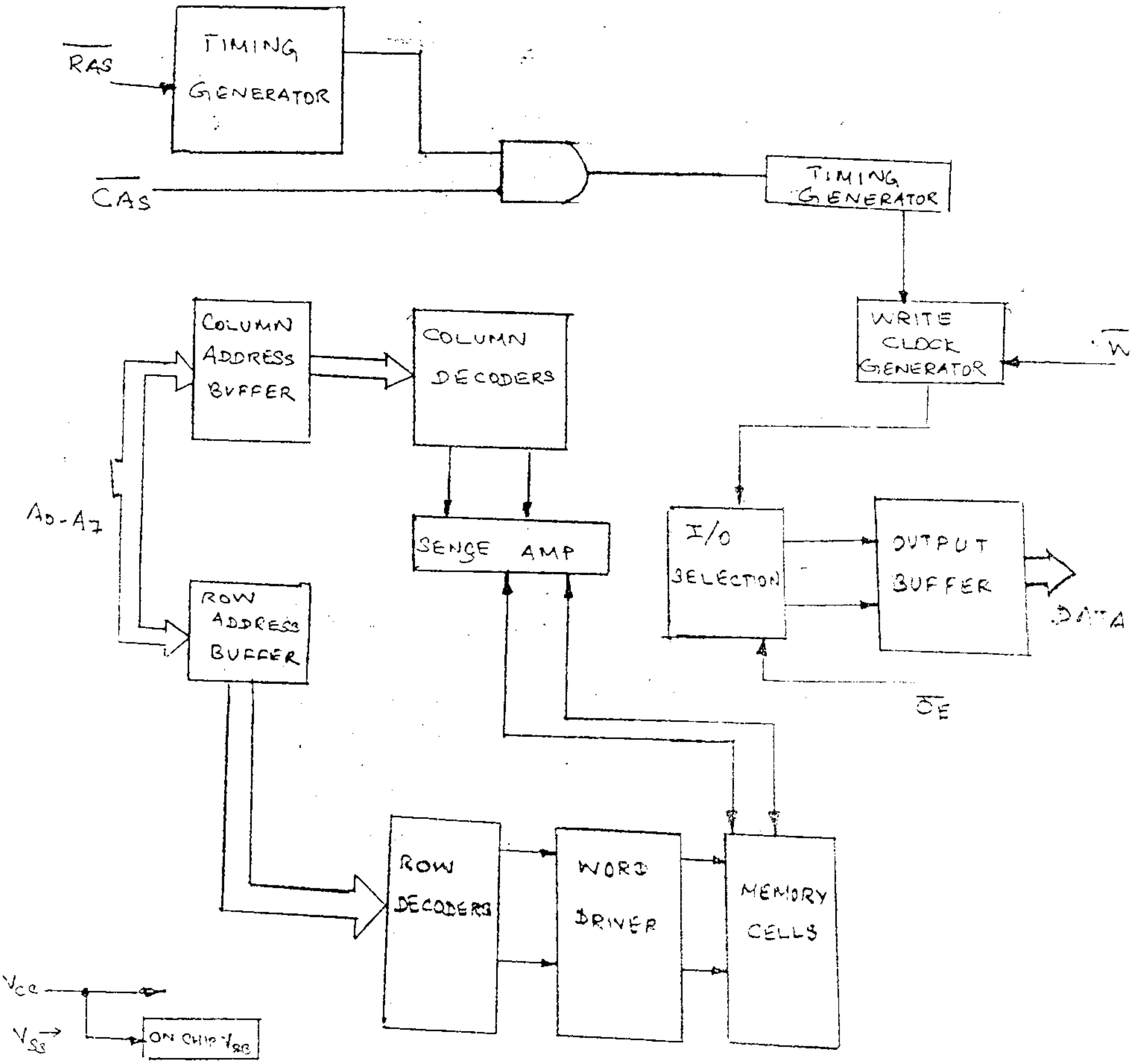


Fig (2.2) BLOCK DIAGRAM OF DYNAMIC RAM 4464

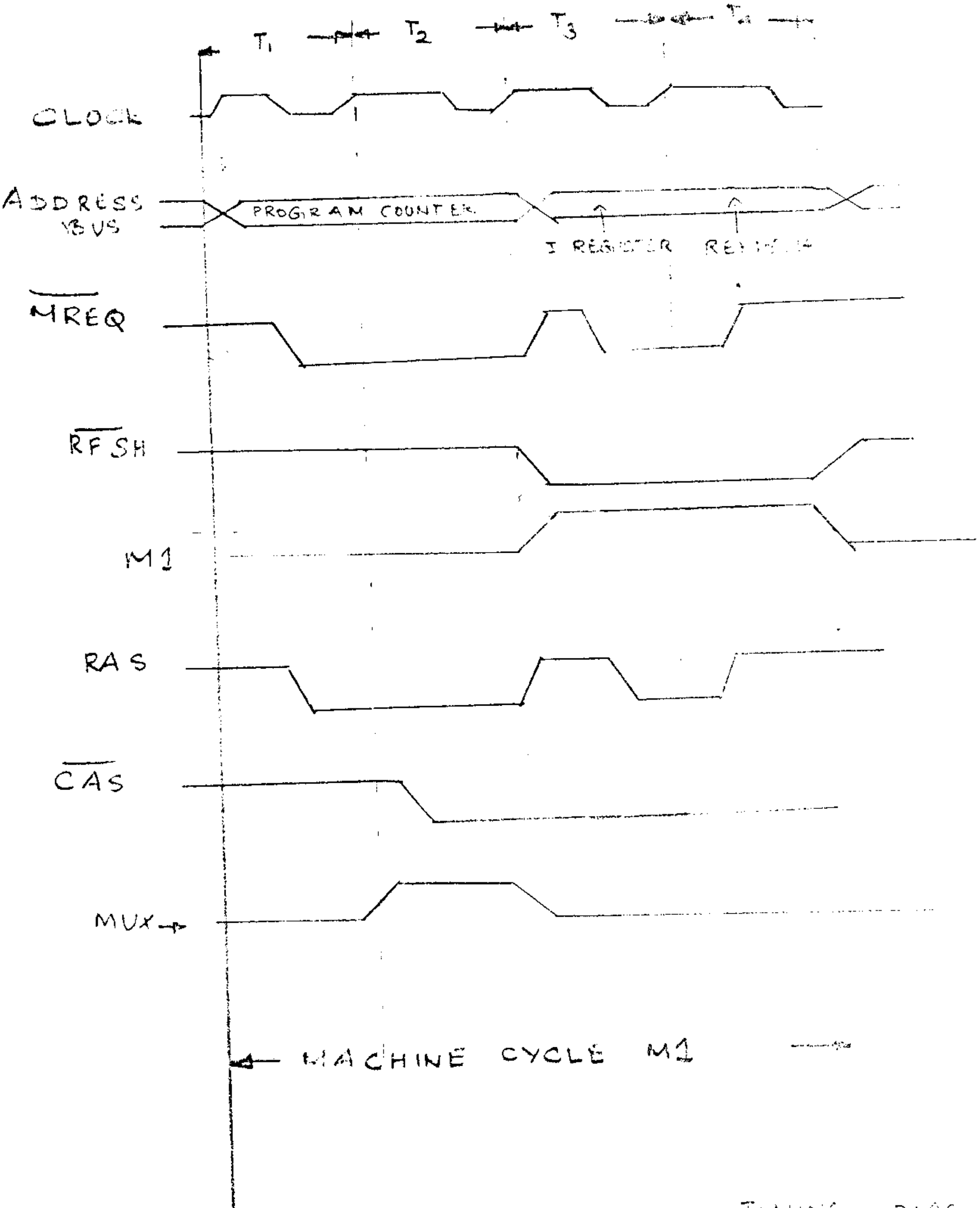


Fig (2.3) DYNAMIC RAM TIMING DIAGRAM

SHOWING Z-80 and RAM SIGNALS

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374
OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

OCTOBER 1975 – REVISED APRIL 1985

- Choice of 8 Latches or 8 D-Type Flip-Flops In a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)

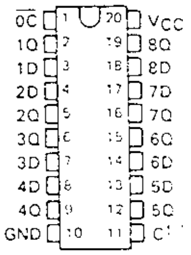
'LS373, 'S373
FUNCTION TABLE

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

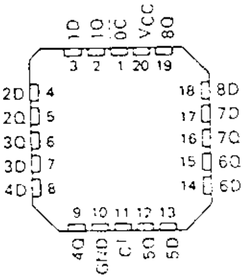
'LS374, 'S374
FUNCTION TABLE

OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

SN54LS373, SN54LS374, SN54S373,
SN54S374 . . . J PACKAGE
SN74LS373, SN74LS374, SN74S373,
SN74S374 . . . DW, J OR N PACKAGE
(TOP VIEW)



SN54LS373, SN54LS374, SN54S373,
SN54S374 . . . FK PACKAGE
SN74LS373, SN74LS374, SN74S373,
SN74S374 . . . FN PACKAGE
(TOP VIEW)



†C for 'LS373 and 'S373 CLK for 'LS374 and 'S374

description

These 8-bit registers feature three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (C) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

3

TTL DEVICES

TYPES SN54390, SN54LS390, SN54393, SN54LS393, SN74390, SN74LS390, SN74393, SN74LS393
DUAL 4-BIT DECADE AND BINARY COUNTERS

OCTOBER 1976 – REVISED DECEMBER 1983

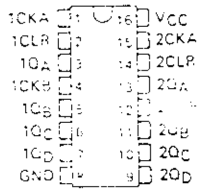
- Dual Versions of the Popular '90A, 'LS90 and '93A, 'LS93
- '390, 'LS390 . . . Individual Clocks for A and B Flip-Flops Provide Dual ÷ 2 and ÷ 5 Counters
- '393, 'LS393 . . . Dual 4-Bit Binary Counter with Individual Clocks
- All Have Direct Clear for Each 4-Bit Counter
- Dual 4-Bit Versions Can Significantly Improve System Densities by Reducing Counter Package Count by 50%
- Typical Maximum Count Frequency . . . 35 MHz
- Buffered Outputs Reduce Possibility of Collector Commutation

description

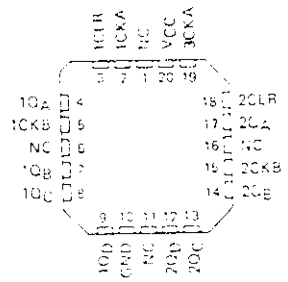
Each of these monolithic circuits contains eight master-slave flip-flops and additional gating to implement two individual four-bit counters in a single package. The '390 and 'LS390 incorporate dual divide-by-two and divide-by-five counters, which can be used to implement cycle lengths equal to any whole and/or cumulative multiples of 2 and/or 5 up to divide-by-100. When connected as a bi-quinary counter, the separate divide-by-two circuit can be used to provide symmetry (a square wave) at the final output stage. The '393 and 'LS393 each comprise two independent four-bit binary counters each having a clear and a clock input. N-bit binary counters can be implemented with each package providing the capability of divide-by-256. The '390, 'LS390, '393, and 'LS393 have parallel outputs from each counter stage so that any submultiple of the input count frequency is available for system-timing signals.

Series 54 and Series 54LS circuits are characterized for operation over the full military temperature range of −55°C to 125°C; Series 74 and Series 74LS circuits are characterized for operation from 0°C to 70°C.

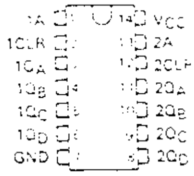
SN54390, SN54LS390 . . . J OR W PACKAGE
SN74390 . . . J OR N PACKAGE
SN74LS390 . . . D, J OR N PACKAGE
(TOP VIEW)



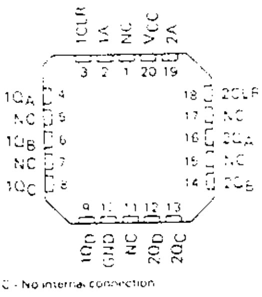
SN54LS390 . . . FK PACKAGE
SN74LS390 . . . FN PACKAGE
(TOP VIEW)



SN54393, SN54LS393 . . . J OR W PACKAGE
SN74393 . . . J OR N PACKAGE
SN74LS393 . . . D, J OR N PACKAGE
(TOP VIEW)



SN54LS393 . . . FK PACKAGE
SN74LS393 . . . FN PACKAGE
(TOP VIEW)



NC – No internal connection

TYPES SN5400, SN54H00, SN54L00, SN54LS00, SN54S00, SN7400, SN74H00, SN74LS00, SN74S00 QUADRUPLE 2-INPUT POSITIVE-NAND GATES

REVISED DECEMBER 1983

- Package Options Include Both Plastic and Ceramic Chip Carriers in Addition to Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

description

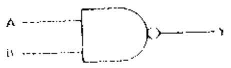
These devices contain four independent 2-input NAND gates.

The SN5400, SN54H00, SN54L00, and SN54LS00, and SN54S00 are characterized for operation over the full military temperature range of − 55°C to 125°C. The SN7400, SN74H00, SN74LS00, and SN74S00 are characterized for operation from 0°C to 70°C.

FUNCTION TABLE (each gate)

INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

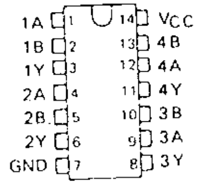
logic diagram (each gate)



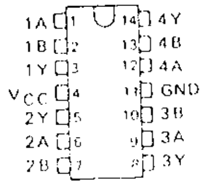
positive logic

Y = $\overline{A \cdot B}$ or Y = $\overline{A} + \overline{B}$

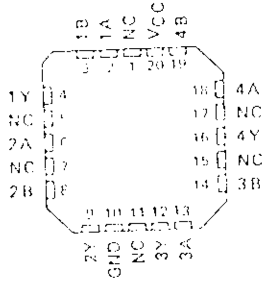
SN5400, SN54H00, SN54L00 . . . J PACKAGE
SN54LS00, SN54S00 . . . J OR W PACKAGE
SN7400, SN74H00 . . . J OR N PACKAGE
SN74LS00, SN74S00 . . . D, J OR N PACKAGE
(TOP VIEW)



SN5400, SN54H00 . . . W PACKAGE
(TOP VIEW)



SN54LS00, SN54S00 . . . FK PACKAGE
SN74LS00, SN74S00 . . . FN PACKAGE
(TOP VIEW)



TYPES SN5404, SN54H04, SN54L04, SN54LS04, SN54S04, SN7404, SN74H04, SN74LS04, SN74S04 HEX INVERTERS

REVISED DECEMBER 1983

- Package Options Include Both Plastic and Ceramic Chip Carriers In Addition to Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

description

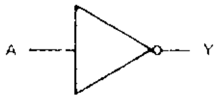
These devices contain six independent inverters.

The SN5404, SN54H04, SN54L04, SN54LS04 and SN54S04 are characterized for operation over the full military temperature range of − 55°C to 125°C. The SN7404, SN74H04, SN74LS04 and SN74S04 are characterized for operation from 0°C to 70°C.

FUNCTION TABLE (each inverter)

INPUTS	OUTPUT
A	Y
H	L
L	H

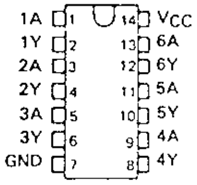
logic diagram (each inverter)



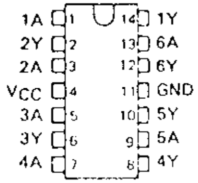
positive logic

Y = $\overline{A}$

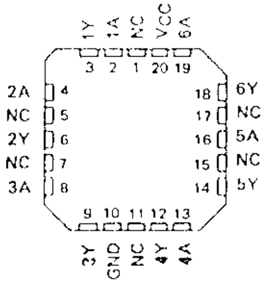
SN5404, SN54H04, SN54L04 . . . J PACKAGE
SN54LS04, SN54S04 . . . J OR W PACKAGE
SN7404, SN74H04 . . . J OR N PACKAGE
SN74LS04, SN74S04 . . . D, J OR N PACKAGE
(TOP VIEW)



SN5404, SN54H04 . . . W PACKAGE
(TOP VIEW)



SN54LS04, SN54S04 . . . FK PACKAGE
SN74LS04, SN74S04 . . . FN PACKAGE
(TOP VIEW)



NC - No internal connection

CONCLUSION

Our Project model printer buffer has been sucessfully designed and fabricated. Printer buffer will get data from any computer system at faster rate and store it in its memory and then dumps data to the printer data slow rate. It can also be used at same time by the another computer while printing. This process is multiplexing.

It can thus be concluded that the computer utilisation time can be increased by a great proportion. This is especially so when the data to be printed is very large.

The cost of the prototype was Rs. 1500/- but if produced commercially, the cost could be estimated to not exceed Rs. 300/-

APPENDIX

List of Components

Semi Conductors

1.	IC1 & IC5	-	74 LS04 HEX INVERTER
2.	IC2	-	74 LS74 DVAL D - TYPE FLIP FLOP
3.	IC3	-	Z80 MICROPROCESSOR
4.	IC4	-	74 LS27 TRIPLE 3-INPUT NOR GATE
5.	IC6 & IC7	-	4464, 64 x 4 bit DRAM
6.	IC8, IC9, IC11	-	74 LS157 Quad 2 to 1 line selector
7.	IC10	-	74 LS393, Dual 4-bit binary counter
8.	IC12	-	74 LS32 Quad 2 - input OR Gate
9.	IC13,IC18	-	74 LS00, Quad 2 - input NAND Gate
10.	IC14	-	74 LS175, Quad D-type flip flop
11.	IC15	-	74 LS138, 1 to 8 decoder demultiplexer
12.	IC16,IC17,IC19,IC21	-	74 LS373, Octal transparent latch
13.	IC22	-	2732 EPROM

II Resistors (all ¼w, 15% carbon)

14.	R1 to R12	-	33 ohm
15.	R13 to R16	-	120 ohm
16.	R17	-	10 kilo ohm
17.	R18	-	1.2 kilo ohm
18.	R19	-	330 ohm
19.	R20	-	180 ohm