DESIGN AND REALIZATION OF A MULTI-EPROM PROGRAMMER

Dagnachew Birru And Hailu Ayele
Electrical Engineering Department
Addis Ababa University

ABSTRACT

The design and implementation of a versatile, low-cost and portable EPROM programmer is presented. In contrast to EPROM programmers dedicated to a single or few types of EPROMS, this programmer could be used to programme various types using a single EPROM socket. The hardware is built using commonly available components and is used to select appropriate pin positions and functions on the EPROM socket and deliver appropriate supply and programming voltages. The selection of the EPROM type, the process of programming, reading and displaying of the contents of an EPROM is controlled by software. The software is implemented using turbo PASCAL, version 5 on an IBM PC/AT personal computer.

INTRODUCTION

An EPROM programmer is one of the essential tools necessary for the development of digital and microprocessor based systems. Several types of programmers are currently available on the market, some of these programmers are designed for a single EPROM type only [1], while the others are for few types with one type of programming algorithm [3]. Some programmers require data to be entered using a built-in keyboard [1] while others can be interfaced to a computer using a dedicated interface card and an accompanying software [2]. The use of dedicated interface card imposes a limitation on the range of computers that can be used with such programmers. In addition, most of the supporting software are designed to operate using some specific hexadecimal file formats of cross-assemblers [1].

In developing countries where equipment and components are procured from diverse sources, it may be difficult to obtain appropriate types of EPROM programmers which can be used without difficulty with the available equipment and software. In such a situation laboratories will be handicapped in the effective utilization of the limited equipment and components they have procured. The EPROM programmer that is presented here affords the possibility of programming different types of EPROMs and thereby avoid the expenses that would have been necessary for the purchase of different programmers.

The paper presents the design and operation of a multi-EPROM programmer which can be configured or programmed for use with different 24-and 28-pin single rail supply EPROMS.

The amount of hardware components used in the design have been minimized by taking advantage of the compatibility of pin positions in different EPROMs and the use of simple circuits and software. The programmer can be connected directly to any IBM parallel printer port without requiring any additional interface card. A description of the circuits for the hardware and software that would enable one to build such a programme is provided.

The special features of this programmer are:

(a) the number of hardware components is small;
(b) it supports the standard 50ms pulse and other intelligent programming algorithms;
(c) it uses a single EPROM socket for different EPROMs;
(d) it does not require any additional interface card and can be used with any computer with a parallel printer port;
(e) it provides the possibility of reading and writing a particular byte.

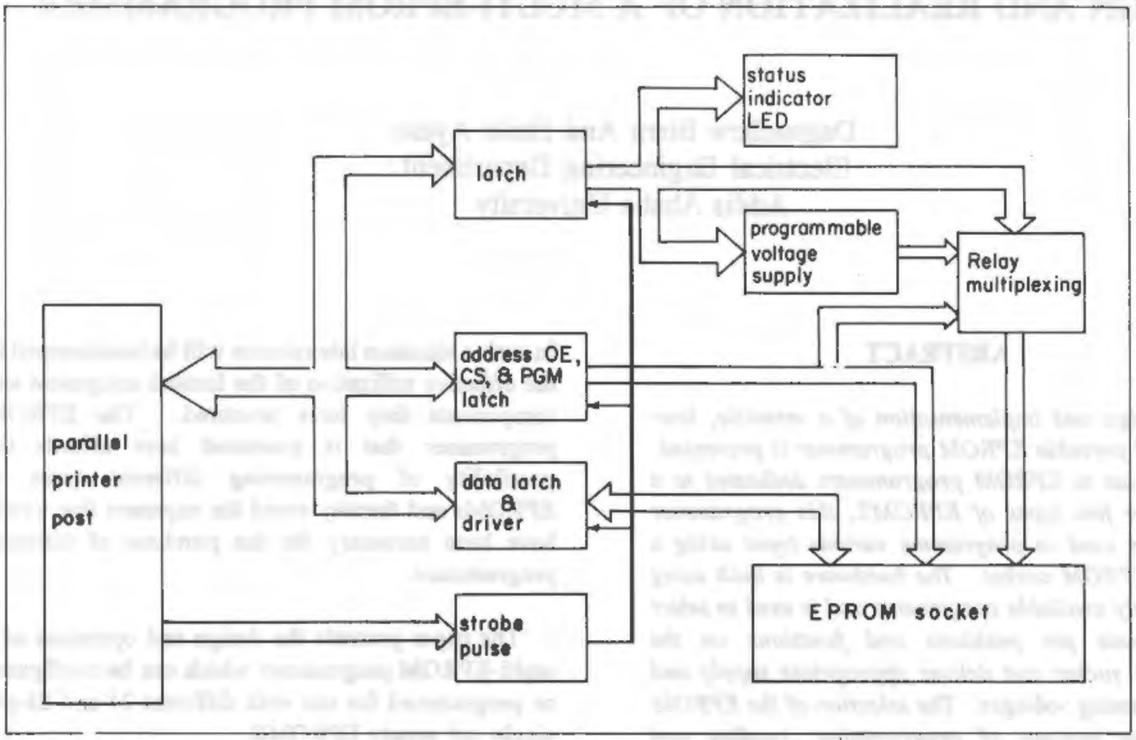

Fig. 1 System Block Diagram

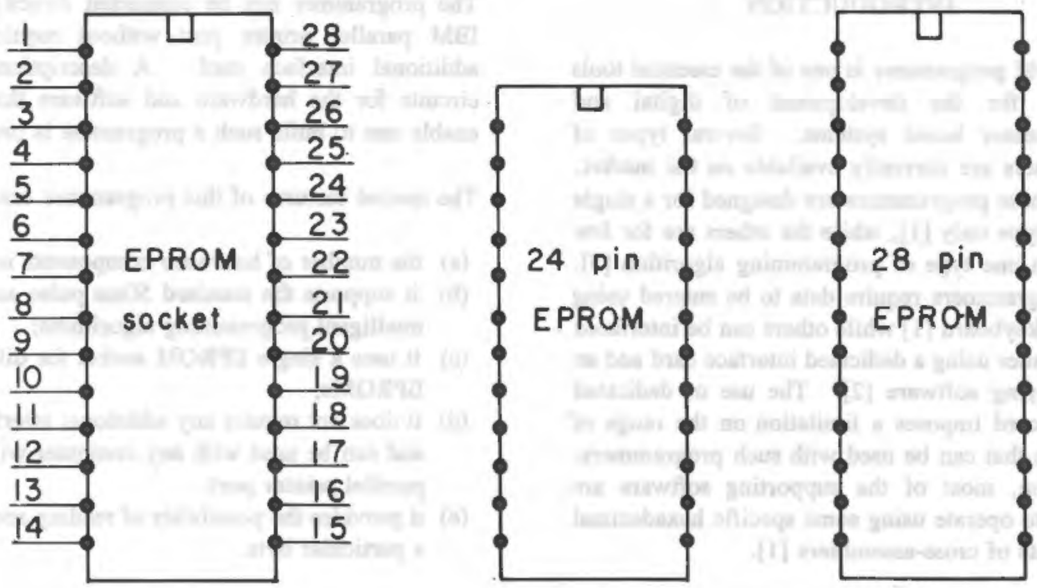

Fig. 2 Direction of the Eprom Relative to the Socket

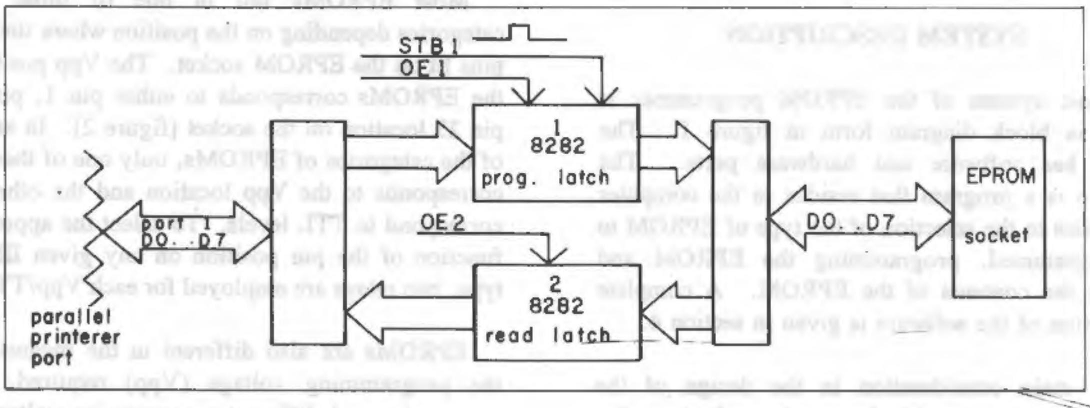

Fig. 3 Data Latch Connection

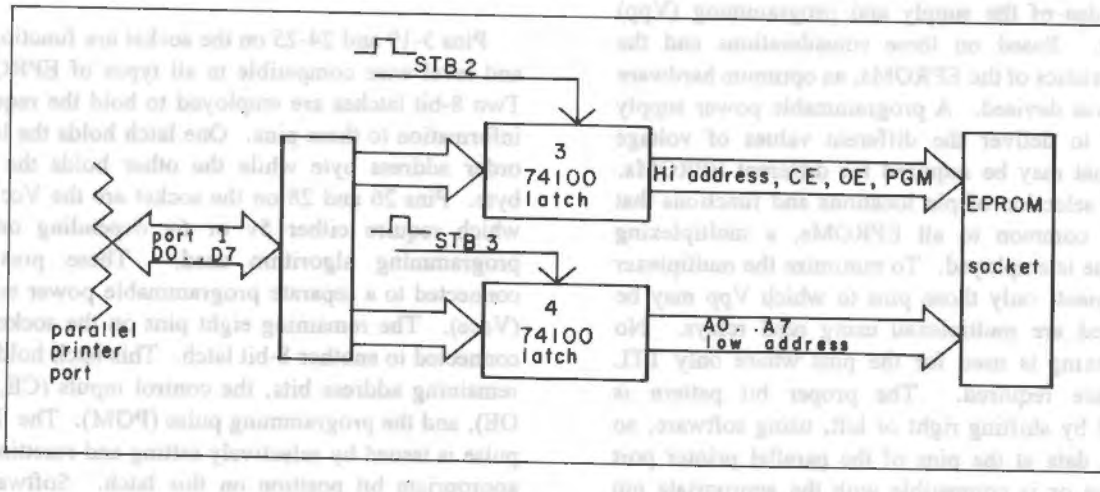

Fig. 4 Address Tatch Connection

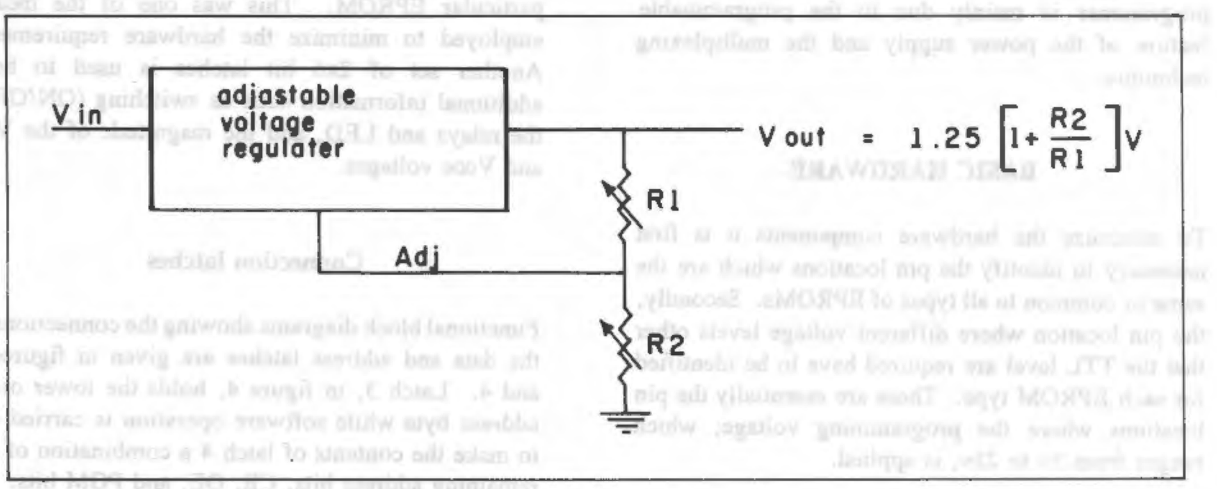

Fig. 5 Variable Voltage Regulator

SYSTEM DESCRIPTION

The basic system of the EPROM programmer is shown in block diagram form in figure 1. The system has software and hardware parts. The software is a program that resides in the computer and assists in the selection of the type of EPROM to be programmed, programming the EPROM and reading the contents of the EPROM. A complete description of the software is given in section 4.

The main consideration in the design of the hardware part were (a) the number of pins of a particular EPROM, (b) the relative locations of the pins of the EPROM socket, (c) the similarities and differences of various EPROMs and (d) the different magnitudes of the supply and programming (Vpp) voltages. Based on these considerations and the characteristics of the EPROMs, an optimum hardware set up was devised. A programmable power supply is used to deliver the different values of voltage levels that may be required for different EPROMs. For the selection of pin locations and functions that are not common to all EPROMs, a multiplexing technique is employed. To minimize the multiplexer requirement, only those pins to which Vpp may be connected are multiplexed using reed relays. No multiplexing is used for the pins where only TTL levels are required. The proper bit pattern is adjusted by shifting right or left, using software, so that the data at the pins of the parallel printer port coincides or is compatible with the appropriate pin locations for the EPROM selected. Latches are used to store information (data, address, OE, CS, PGM) during read and write cycles. The versatility of this programmer is mainly due to the programmable feature of the power supply and the multiplexing technique.

BASIC HARDWARE

To minimize the hardware components it is first necessary to identify the pin locations which are the same or common to all types of EPROMs. Secondly, the pin location where different voltage levels other that the TTL level are required have to be identified for each EPROM type. These are essentially the pin locations where the programming voltage, which ranges from 5v to 25v, is applied.

Most EPROMs fall in one of three major categories depending on the position where their Vpp pins fit on the EPROM socket. The Vpp position of the EPROMs corresponds to either pin 1, pin 2 or pin 23 location on the socket [figure 2]. In any one of the categories of EPROMs, only one of these pins corresponds to the Vpp location and the other pins correspond to TTL levels. To select the appropriate function of the pin position on any given EPROM type, two relays are employed for each Vpp/TTL pin.

EPROMs are also different in the magnitude of the programming voltsge (Vpp) required. The commonly used different programming voltages are 5v, 12.5, 21v vand 25v [3,6]. To generate the required Vpp for a particular EPROM a programmable voltage supply was designed and built.

Pins 3-19 and 24-25 on the socket are functionally and level wise compatible in all types of EPROMs. Two 8-bit latches are employed to hold the required information to these pins. One latch holds the lower order address byte while the other holds the data byte. Pins 26 and 28 on the socket are the Vcc pins which require either 5v or 6v depending on the programming algorithm used. These pins are connected to a separate programmable power suppiy (Vcce). The remaining eight pins on the socket are connected to another 8-bit latch. This latch holds the remaining address bits, the control inputs (CE, CS, OE), and the programming pulse (PGM). The PGM pulse is issued by selectively setting and resetting the appropriate bit position on this latch. Software is used to shift right or left the other bit positions in this latch so that the bit pattern contained in this latch corresponds to the appropriate bit position for particular EPROM. This was one of the means employed to minimize the hardware requirement. Another set of 2x6 bit latches is used to hold additional information such as switching (ON/OFF) the relays and LED, and the magnitude of the Vpp and Vcce voltages.

Connection latches

Functional block diagrams showing the connections of the data and address latches are given in figures 3 and 4. Latch 3, in figure 4, holds the lower order address byte while software operation is carried out to make the contents of latch 4 a combination of the remaining address bits, CE, OE, and PGM bits.

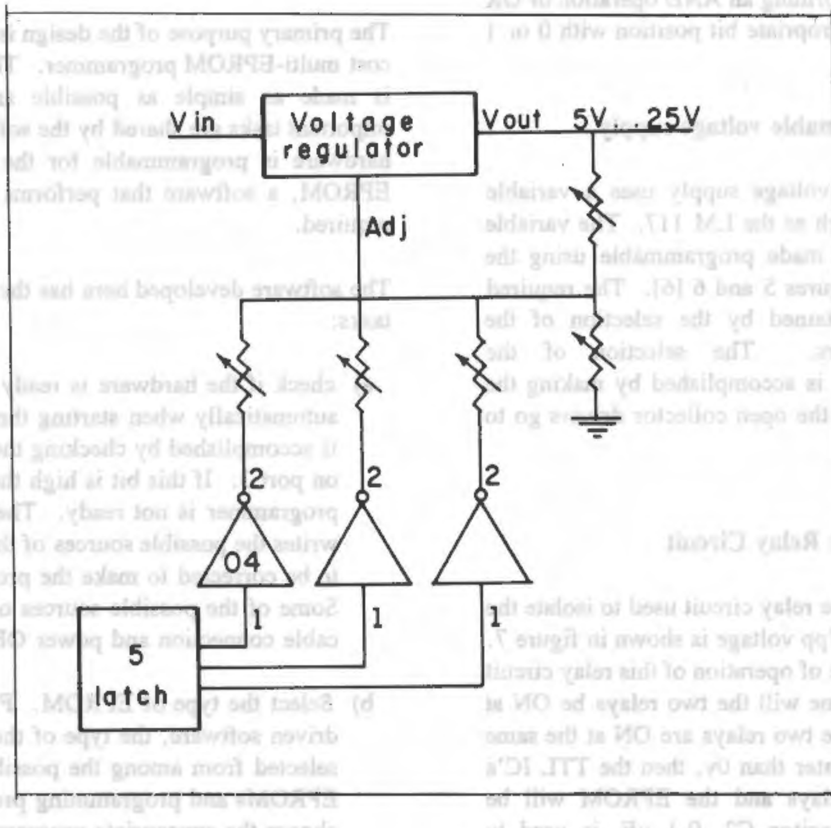

Fig. 6 Programmable Voltage Supply

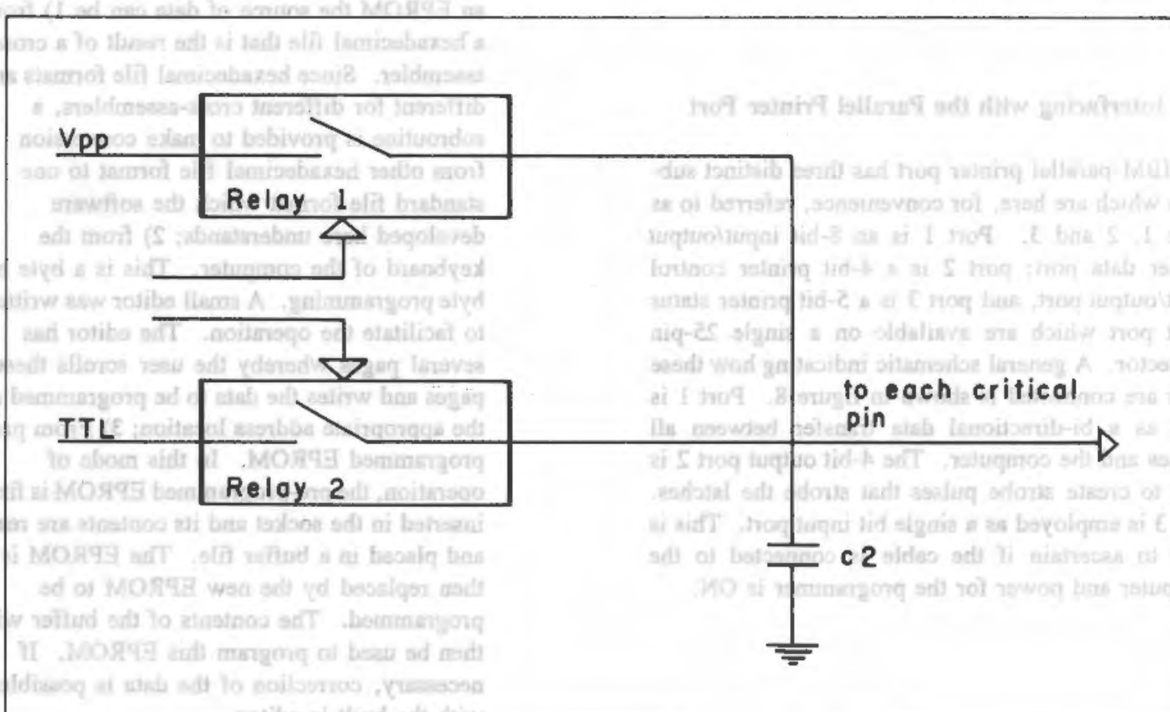

Fig. 7 Two Relays Connected to isolate the TTL Pins from Vpp

This is performed by shifting the required bits either left or right and performing an AND operation or OR operation on the appropriate bit position with 0 or 1 respectively.

Programmable voltage supply

The programmable voltage supply uses a variable voltage regulator such as the LM 117. The variable voltage regulator is made programmable using the scheme shown in figures 5 and 6 [6]. The required voltage level is obtained by the selection of the appropriate resistors. The selection of the appropriate resistors is accomplished by making the output transistors of the open collector drivers go to the low state.

The Relay Circuit

The connection of the relay circuit used to isolate the TTL pins from the Vpp voltage is shown in figure 7. The normal sequence of operation of this relay circuit is such that at no time will the two relays be ON at the same time. If the two relays are ON at the same time and Vpp is greater than 0v, then the TTL IC's connected to the relays and the EPROM will be damaged. The capacitor C2, 0.1 μF, is used to suppress spurious and transient surge voltages which may damage the device [3].

Interfacing with the Parallel Printer Port

The IBM parallel printer port has three distinct sub-ports which are here, for convenience, referred to as ports 1, 2 and 3. Port 1 is an 8-bit input/output printer data port; port 2 is a 4-bit printer control input/output port, and port 3 is a 5-bit printer status input port which are available on a single 25-pin connector. A general schematic indicating how these ports are connected is shown in figure 8. Port 1 is used as a bi-directional data transfer between all latches and the computer. The 4-bit output port 2 is used to create strobe pulses that strobe the latches. Port 3 is employed as a single bit input port. This is used to ascertain if the cable is connected to the computer and power for the programmer is ON.

SOFTWARE

The primary purpose of the design is to construct low cost multi-EPROM programmer. The hardware part is made as simple as possible and some of the important tasks are shared by the software. Since the hardware is programmable for the desired type of EPROM, a software that performs several tasks is required.

The software developed here has the following major tasks:

a) check if the hardware is ready. This is done automatically when starting the programme. It is accomplished by checking the status of the bit on port 3. If this bit is high then it implies that programmer is not ready. The software then writes the possible sources of the error that have to be corrected to make the programmer ready. Some of the possible sources of error are loose cable connection and power OFF condition.

b) Select the type of EPROM. From the menu driven software, the type of the EPROM is selected from among the possible types of EPROMs and programming process and also to choose the appropriate programming algorithm.

c) Programming the EPROM. In programming an EPROM the source of data can be 1) from a hexadecimal file that is the result of a cross-assembler. Since hexadecimal file formats are different for different cross-assemblers, a subroutine is provided to make conversion from other hexadecimal file format to one standard file format which the software developed here understands; 2) from the keyboard of the computer. This is a byte by byte programming. A small editor was written to facilitate the operation. The editor has several pages whereby the user scrolls these pages and writes the data to be programmed at the appropriate address location; 3) From pre-programmed EPROM. In this mode of operation, the pre-programmed EPROM is first inserted in the socket and its contents are read and placed in a buffer file. The EPROM is then replaced by the new EPROM to be programmed. The contents of the buffer will then be used to program this EPROM. If necessary, correction of the data is possible with the built-in editor.

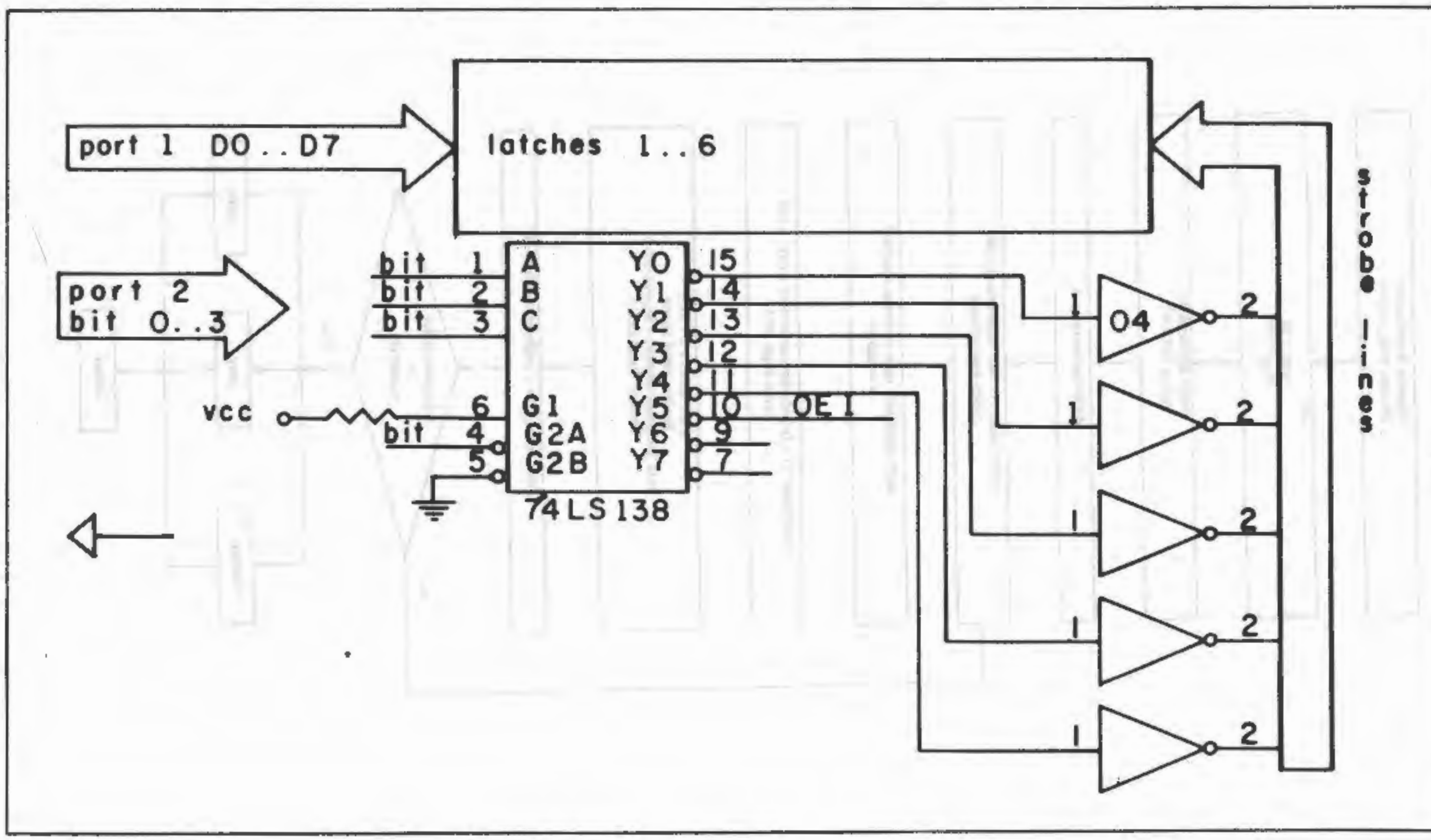

Fig. 8 General Diagram of the Interfacing Section

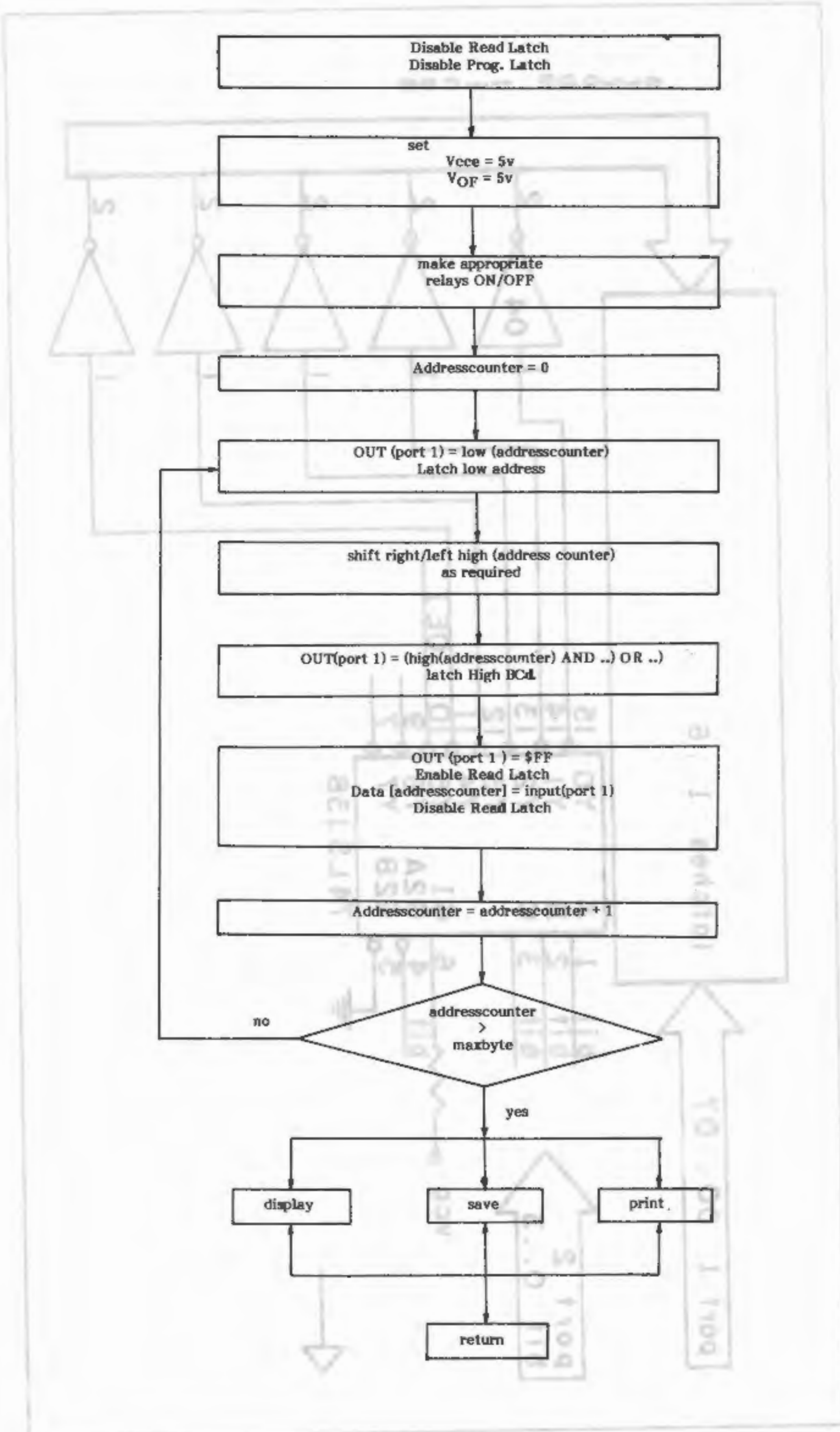

Fig. 9 Flow Chart

d) Reading EPROM content and save to a disk, display on the console, or print the contents. This procedure follows the general flow chart given in figure 9.

CONCLUSION

The hardware and software were tested. using Intel's Intelligent Programming algorithm [1], it took only 30 seconds to write zeros into the 2048 hyte positions of the 2764 EPROM. The same task required about 110 seconds using the standard 50ms pulse technique on the 2716 EPROM. An additional special feature of this programmer is its capacity to duplicate the contents of a pre-programmed EPROM using a single EPROM socket. The programmer was also tested using different parallel ports from different manufacturers on the IBM PC and PS/2.
and looks like the familiar SideKick [8] with similar window and menu selection techniques. Software and hardware facilities that can provide automatic power shut-down if no operation is performed on the programmer for more than a given time can easily be incorporated. Presently the programmer is capahle of programming 2,4 and 8-K byte EPROMs. Further, with only an additional latch and two relays, the programmer can also be used to program 16- and 32 Kbyte EPROMs.