International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:12, 2011

Implementation of Parallel Interface for Microprocessor Trainer

Moe Moe Htun, Khin Htar Nwe

***Abstract*—**In this paper, parallel interface for microprocessor trainer was implemented. A programmable parallel–port device such as the IC 8255A is initialized for simple input or output and for handshake input or output by choosing kinds of modes. The hardware connections and the programs can be used to interface microprocessor trainer and a personal computer by using IC 8255A. The assembly programs edited on PC's editor can be downloaded to the trainer.

***Keywords*—**Parallel I/O ports, parallel interface, trainer, two 8255 ICs.

I. Introduction

DEVELOPMENTS in computer technology area have increased the importance of computer control and interfacing system. The interfacing system is also consisting of hardware, software, or both that allows two dissimilar components to interact. The control and that system are microprocessor control, microcontroller control and the computer aided control/interfacing. Computer aided control and interfacing can be implemented by serial port, parallel port prepared by special purposes. The parallel port can be very useful input/output channel for connecting our own circuits to personal computer and other devices. Interfacing between the microprocessor trainer and PC uses the address decoder device and programmable input/output device. This paper covers both hardware and software, including how to design the microprocessor trainer that connect to the port as well as how to write programs to input/output and download from the other computer.

II. Related work

The 8255A Programmable Peripheral Interface (PPI) IC provides all the facilities for interfacing parallel inputs/outputs. It can be programmed to three simple I/O ports (mode0), two handshaking I/O ports (mode1), or bidirectional I/O port with five handshaking signals (mode2). It has 24 lines of digital input/output, two groups of twelve lines, or three groups of eight lines. The two groups of input/output pins are Group A and Group B. Each of these two groups contains a subgroup of eight input/output called 8-bit port and another subgroup of four lines called a 4-bit port. Thus Group A contains an 8-bit port A along with a 4-bit port C upper (PortC4-PortC7) and Group B contains an 8-bit port B and a

4-bit port C lower (PC0- PC3). The Group A and Group B block receive control from the CPU and issues commands to their respective ports. The 8255A consists of a buffer and address decoder circuit. The buffer circuit buffers the signals between the 8255A and the Microprocessor Trainer board, and the address decoder circuit assigns the 8255A an address on the address bus. Data is transmitted or received by the buffer on execution of input or output instructions by the CPU.

The chip select, CS is to communicate between CPU and 8255. RD, read control enables the CPU to read the data in the ports or the status word through data bus buffer to the device controlling the 8255. WR is write control and the CPU can write data on to the ports or on to the control register through the data bus buffer. RESET is a high on pin clears the control register and all ports are set to the input mode. A0 and A1, the input address lines allow the selection of one of three ports or the control register. [3], [12].

The 74LS138, 3-to-8 line decoder, 16-pin IC decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. The features of 74LS-138 are designed specifically for high speed, memory decoders and data transmission systems. [14].

The parallel port's hardware includes the back- panel connector and the circuits and the system's expansion bus. [13].

III. Contribution

Most of the interfacing using 8255IC describes simple input and output, strobed input and output, and handshake input and output. In this paper, I initialize a programmable parallel-port device for simple input or output and for handshake input or output by choosing kinds of modes and how parallel data is sent to a microprocessor trainer from a personal computer on a handshake basis. I concentrate on the devices and the hardware connections. And also include the input/output operation by using the program.

Moe Moe Htun, Lecturer, at the Department of Electrical Communication, Technological University (Hinthata), Myanmar and the author was graduated from the University of Computer Studies, Yangon. (phone: 0943033207; e-mail: dmmhtun@ gmail.com).

Khin Htar Nwe,,Lecturer, at the Hardware Department, Computer University (Lashio), Myanmar, the author was graduated from the University of Computer Studies, Yangon. (phone: 0949281527 ;e-mail: khinhtarnwe@gmail.com).

IV. Hardware Design of The Microcontroller-Based Parallel I/O Interface System

The hardware and the software implementation will be summarized. In the Microprocessor Trainer system, the program debugger, a kind of DMA (Direct Memory Access) controller, is contained in one of the hardware module. There are totally six modules (CPU module, Memory module, User Interface module, DMA module, PIO (Parallel Input /Output) module and Power Supply module) in the Trainer system and all modules are provided as separate chip, linked together via bus connections on a printed circuit board and supplied 5V DC power. The complete block diagram of the system is shown in Fig.1.

The Microprocessor Trainer used the 16-bit wide address bus (A_0 to A_{15}) and the data bus (D_0 to D_7) which connect all of the modules and used four control signals such as MEMWR, MEMRD, IOWR and IORD. Memory read (MEMRD) and memory write (MEMWR) are used to connect the address lines to the memory module. And the I/O read (IORD) and I/O write (IOWR) are also used to connect to the I/O CPU and CPU modules to control the keypad, display modules and the input/output data. These four memories and I/O control lines are connected with DMA module and the system buses are used by the main processor and the DMA controller alternatively.

The main part which involved in the parallel input/ output interface system are two IC8255A, 74LS138, eight 7-segments, parallel printer port . 8255A has two address lines, indicating that it has 4 read/write registers within it which the CPU has to address. The address bits from PIO chips are connected to the 74LS138 decoder of the address lines to decode the address from the 16 address bits from the trainer. The 74LS138 is used to produce chip selects signals for 8255 and other I/O devices. This will generate the appropriate output to drive the chip selects of the respective chips. The block diagrams are as shown in Fig. 2(a) and Fig. 2(b).

At first, in one of the 8255 PIO device, port C is initialized in mode 1 for handshaking data transfer and Port A is used as the data port. Port C 4 is used as the strobed line for the printer port (Pin number 1). The acknowledge pin (Port C 7) is connected to the printer port of pin number 10. The input buffer full, IBF signal on Port C5 of the PIO chip would normally be used as the busy pin of the printer port or parallel port (pin number 11). In addition, the other 8255A PIO is set up in mode 0 because of the data coming from the printer port can be seen. And that Port A is used to connect to eight LEDs. And then the variables are initialized.

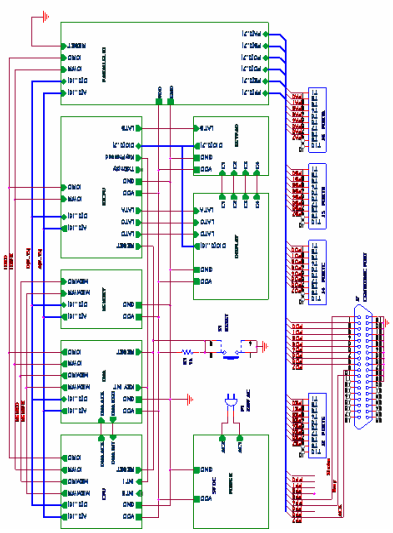

Fig.1 Complete Block Diagram of Microprocessor Trainer

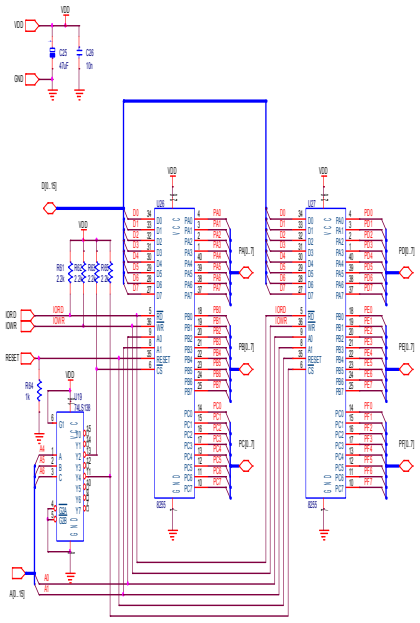

Fig. 2(a) Circuit Diagram of Simple Input/ Output

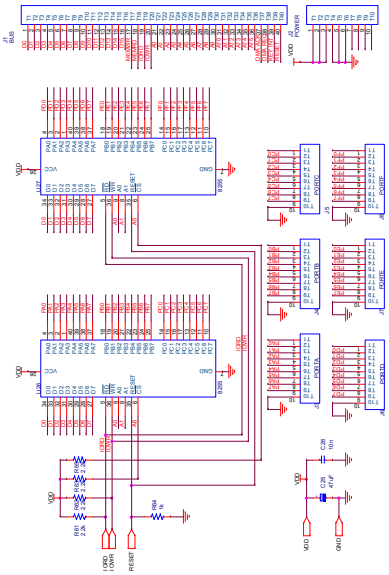

Fig. 2(b) Circuit Diagram of Parallel Input/ Output (continued)

V. Hardware and Software Implementation of Parallel Interface

A. Controlling with Assembly Software

1. Flow of the Parallel Port Driver

This program is written on the memory module, EEPROM in the trainer to read data from the printer port, to send the ACK to the PC to tell it has been accepted and send it the next data and to store the data. Before storing the data in the word counter, the data length is divided by two to get a word count. All of bits in sending data can't be read at one time since the data bits are 16 bits. Thus the data is read twice when it is sent from the PC and then it gets the data as a word.

In sending address type, it also uses the polling method. The address lines include the 16 bits address. So it can't be sent at one time and stored in the address counter. So the data is stored in the MSB address counter and LSB address counter, by shifting eight times to the left. The address counter is increased and the word counter is decreased. If the word counter is not zero, the data will be reread again from the printer port by polling. If the word counter is zero, all of the data from the printer port has been read. The Fig.3 (a) and Fig.3 (b) show the flow charts of the parallel port driver.

2. Flow of the Polling

In polling method, the input buffer full flag (IBF) of the PIO is read to poll the data transfer from the printer port. If the IBF is set, the PIO device reads the data from the parallel port. After reading the data, the IBF is reset. The Fig. 4 is shown flow chart of polling.

3. Flow of the Acknowledge

Before using the ACK pin, first it is cleared and wait to read the data. It is shown by Fig.5 of flow chart of acknowledge.

B. Controlling with Assembly Software by using C#.Net

1. Flow of the Download Program

This software is written on the PC to download the test download program from it. First a number of lines are calculated to get a line in the test download program and extract byte count, address MSB, address LSB and data type. If the data type is equal to 01, the data byte is sent zero and sent the address MSB, sent the address LSB and sent the data type 01- the end of the signal. The length or byte count, address MSB, address LSB are converted to hex numbers after checking the data type is equal to 00. Sending the start signal, checking the busy signal, and sending strobed signal use write port program.

After sending the data and strobed signal, the ACK is checked. If the ACK isn't send from the PIO, the counter is counted until three times. If the number of count is three, the message is displayed that the port is busy. But if not, the start signal is sent again using write port program. This process is shown as a flow chart in Fig6 (a) and Fig.6 (b).

The hex file format in the test download program is the first character (:) indicates the start of a record (start code, ASCII colon **":"**) and the next two characters, 10 (10h) indicate the record length (byte count, two hex digit pairs) in the data field. It is at least 0 to most 16 bit, 16 (0x10) bytes of data. The next characters E100 give load address. It is address type (four hex digits). The next two characters indicate the record type (two hex digits), 00 to 05, defining the type of the data field. Then we have the actual data. The last two characters 6F are a checksum (sum of all bytes += 6F), two hex digits.

Record types in this program are 00 is data record, 01 - End of file record and 04 is extended linear address record. The rests are 02 - extended segment address record, 03 - start segment address record and 05 - start linear address record.

2. Flow of the Write Port

In write port program, the port of PIO is checked whether busy or not. And if the port is busy, the time is also checked to be time out or not. The timer operates during five seconds. If it is overtime, the error message is displayed. If the PIO port isn't busy, the data is sent. The strobed pulse signal is generated after 5µs. And then the ACK from the PIO is checked. Fig.8 shows the flow chart of write port.

3. Flow of Running Light

The running light test is used to able to see with LEDs when the program is downloaded from the PC to the trainer. In testing, the eight LEDs are running when the program is downloaded. As a result of running light test program, the

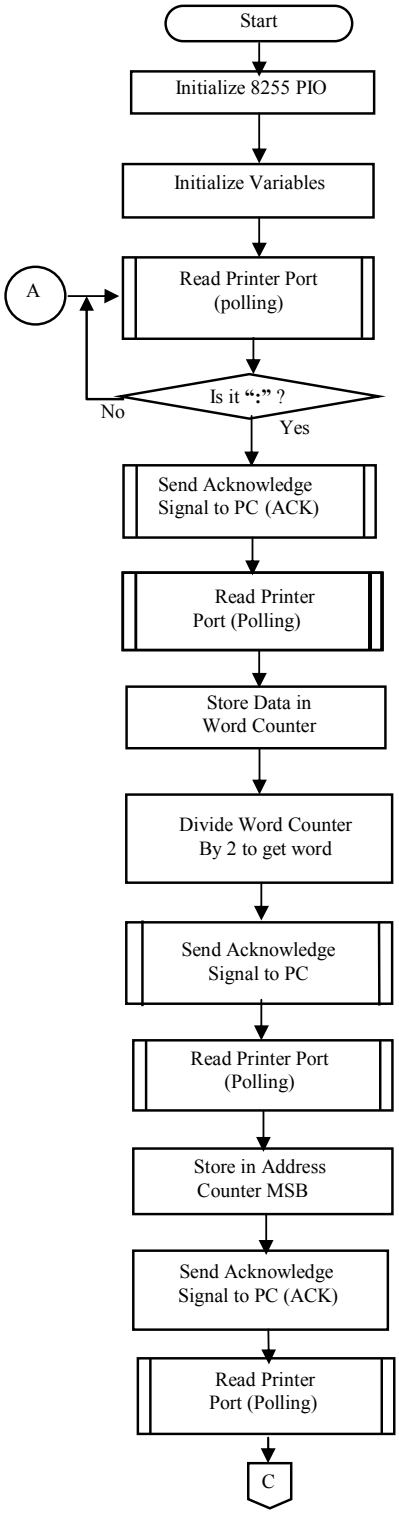

Fig.3 (a) Flow Chart of Parallel Port Driver

download process can be checked by LEDs lighting. The running light test is shown as flow chart of Fig.9 and Fig.7 is the circuit diagram of LED display module.

4. Flow of Input Test

It is written to test the input/output data transfer from the microprocessor trainer and shown in Fig. 10 of flow chart of input test.

5. Flow of bit test program

It is written to be clear the PIO chips for using the Port C bits for status. Thus Port C is making turn to on and off while the parallel port of the 8255 PIO chips are doing correctly. Fig.11 is the flow chart of bit test.

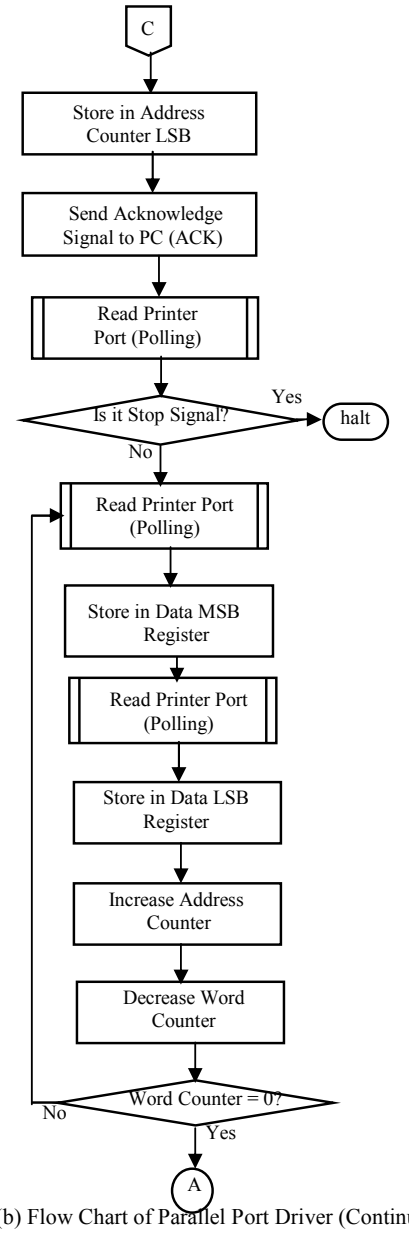

Fig. 3(b) Flow Chart of Parallel Port Driver (Continued)

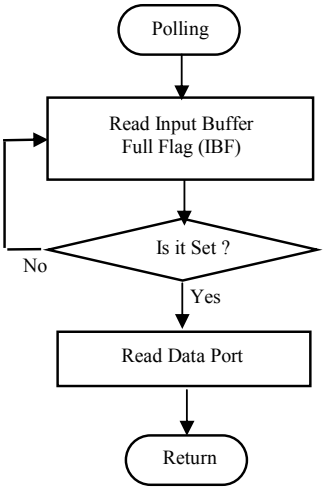

Fig. 4 Flow Chart of Polling

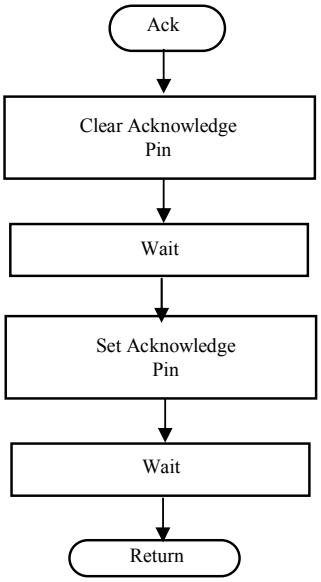

Fig. 5 Flow Chart of Acknowledge

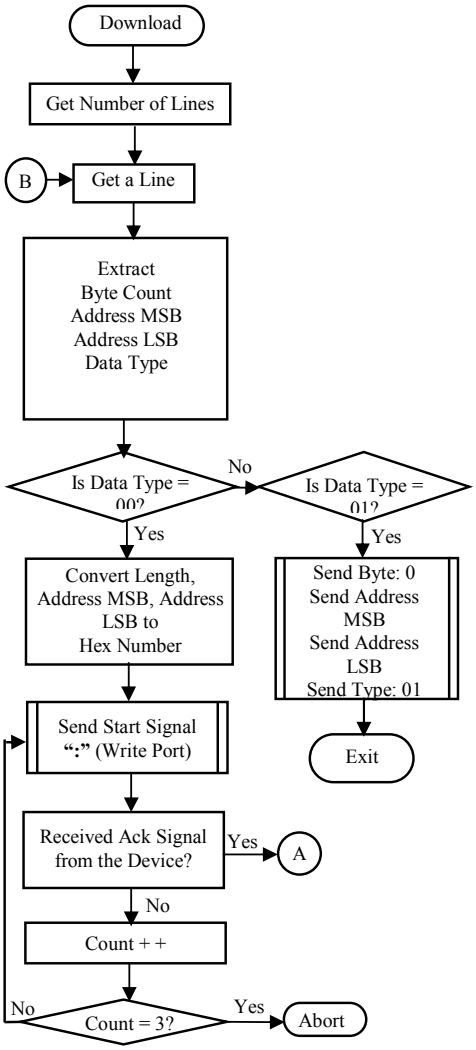

Fig. 6(a) Flow Chart of Download

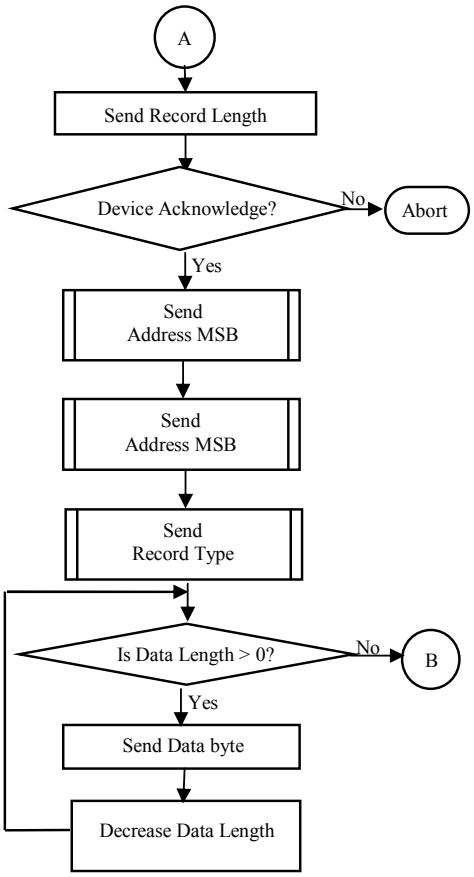

Fig. 6(b) Flow Chart of Download (Continued)

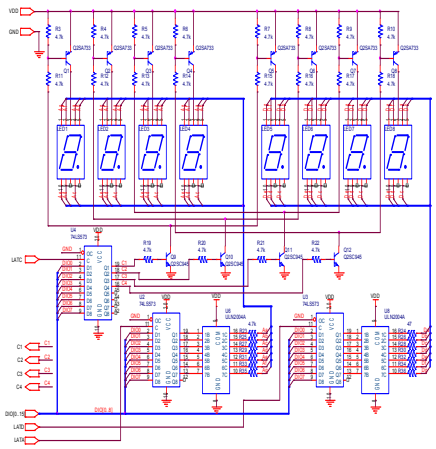

Fig. 7 Circuit Diagram of LED Display Module

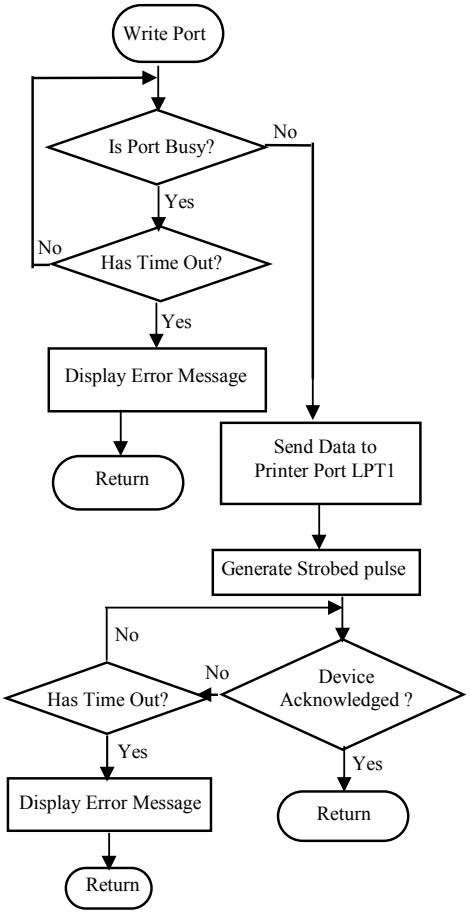

Fig. 8 Flow Chart of Write Port

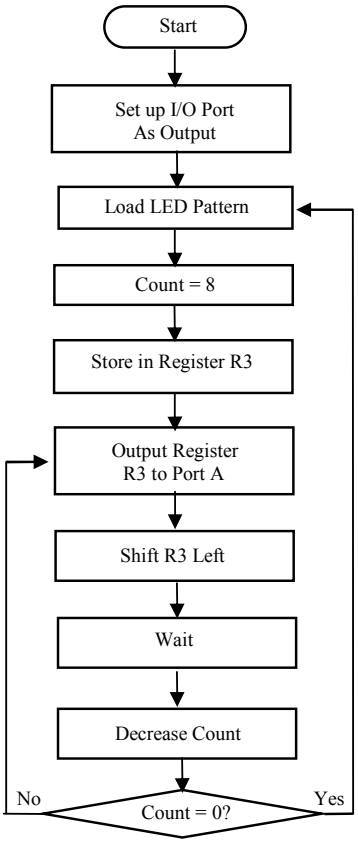

Fig. 9 Flow Chart of Running Light

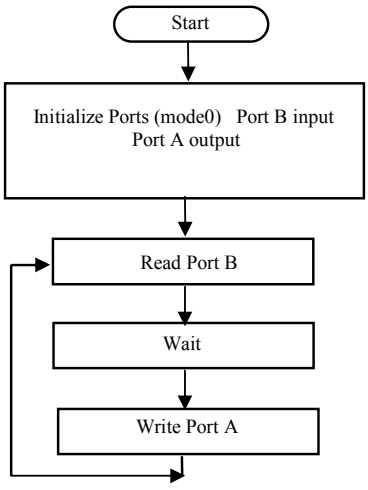

Fig. 10 Flow Chart of Input Test

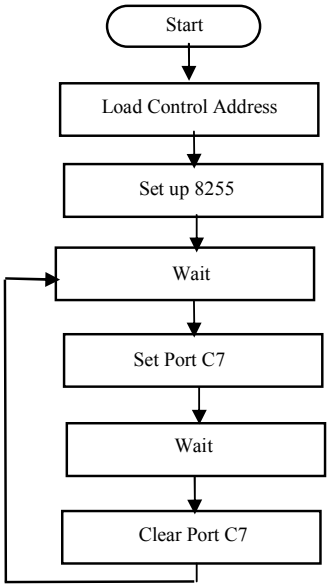

Fig. 11 Flow chart of bit test

VI. Conclusion

The design concept for the parallel port- based interfacing system has been defined. This interfacing system is perfect to input/output or parallel input/output from the microprocessor trainer. And it is also perfect to download any program from the PC to the microprocessor trainer. It is the best low cost, good quality interfacing. In this paper, it is impossible to make the interfacing systems with only the hardware design. It is able to make both the hardware and software or the driver that control them are just used.

Two 8255A chips are used. One is used as the printer interface and another is used to handle all other parallel I/O activities. The circuit is constructed and tested electrically without any chips installed. A driver program is loaded into EEPROM that configures the parallel ports for handshake data transfer. A test downloading program is then loaded and it is shown out from DMA to the display. The input/output test program is also loaded under control of the CPU and outputted with eight LEDs.

This paper helps to the learners who are familiar with the peripheral parallel input/output chip. It would be a good tool for learners to learn about the instructions written in microprocessor trainer complier.

In this paper, there are a few tests and a few programs can be downloaded from the PC. Thus, as an extension, the learners who learn the microprocessor trainer can download a lot of programs as needed and can use this chip as a serial port to interface the peripherals and to download programs from the PC.

ACKNOWLEDGMENT

First of all, the author would like to express Her Gratitude to Professor Dr. Nilar Thein, Rector of University of Computer Studies, Yangon, for overall supporting during her paper and encouragement for this paper.

The author would like to express Her Deep Appreciation to Pro Rector U Kyaw Swa Soe, University of Computer Studies, Yangon, for providing the facilities, guidance, invaluable advice and supervision throughout her paper.

The author would like to thank Deputy Director General U Win Khine Moe, Myanmar Science and Technological Research Department who advise her to do this paper.

The author is also grateful to Professor Daw New Ni, University of Computer Studies, Yangon, for this suggestion and helpful advice throughout her paper.

The author is also grateful to Dr. Mi Mi Thet Thwin, Rector, University of Computer Studies, Mandalay, for their kind permission to use the required laboratory instruments in this research and gave valuable suggestions at the seminars and made it possible for me to complete the whole paper.

The author is very thankful to Professor Dr. Win Aye, Computer University (Mandalay) for her invaluable guidance, advice, encouragement for this paper.

The author is particularly indebted to the volunteers, who participated in this study and would like to thank to all who have responded so willing to her request for technical assistance in the preparation of this paper.

Lastly, the author has great pleasure in expressing her gratitude to her family for their constant encouragement.

REFERENCES

[1] Andrew S. Tanenbaum, "structured computer organization"
[2] Arthur B. Williams, Editor in Chief Research, and Development "Designer's andbook of integrated circuits" Coherent Communications Systems Cop. Hauppauge, N.Y
[3] Douglas V.Hall, "Microprocessors and interfacing programming and hardware" econd edition
[4] Hutchings, Howard "British Library Cataloguing in Publication Data" Interfacing with C/ Howard Hutchings. First published 1995
[5] John Uffenbeck, "Microcomputers and microprocessors" the 8080, 8085, and Z-80 programming, interfacing, and troubleshooting. Second Edition. ©1991, 1985 by Prentice-Hall, Inc. A Division of Simon & Schuster Englewood Clifts. NJ 07632
[6] Ken Arnold, Embedded Controller Hardware Design".
[7] Mark Balch, "COMPLETE DIGITAL DESIGN" A Comprehensive Guide to Digital Electronics and Computer System Architecture.
[8] Martin Bates, "Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation".
[9] Murry Sargent 111 and Richard L. Shoemaker, "The IBM personal computer rom the inside out" revised edition.
[10] Roy goody, "Programming and interfacing the 8086/8088 microprocessor".
[11] Sajjan G. Shiva,"Computer design and architecture" third addition, revised and expended.
[12] S.A.Money, "Microprocessor data book".
[13] Jan Axelson, "Parallel Port Complete Programming, Interfacing, & Using the PC's Parallel Printer"
[14] James M.Sibigtroth, "M68HC05 Family, Understanding Small Microcontrollers-Rev. 2.0",