Short Notes

Introductory UART Experiments for a Microcomputer Course

ROBERT J. PLESSET AND JOHN C. FIELD

***Abstract*—A course is briefly described which deals with microprocessor interfacing. One portion of the course covers programmable LSI interface chips. Introductory experiments are described for one of these chips, the Motorola 6850 asynchronous communications interface adapter (ACIA). The experiments allow the student to, in effect, "single-step" through the operation of the ACIA by inputting clock pulses asynchronously. The values of important signals, including the clock, are indicated in LED's. Thus, the student gets visual reinforcement of the ACIA's characteristics.**

Introduction

The Department of Electrical Engineering at the University of Maine, Orono, offers two undergraduate lecture/laboratory courses in microprocessors. They are similar in content to the initial two courses in the curriculum described by Leventhal [1]. The first is at the freshman level and emphasizes machine and assembly language programming for the Motorola 6800 microprocessor [2].

The second course is offered in the junior year after the student has had several courses in digital and linear electronics. Its emphasis is on interfacing. The topics covered in this course include software development (including the use of an emulator), addressing (full and partial decoding), using RAM's, ROM's, and EPROM's, interrupts, parallel data transfer using a 6821 peripheral interface adapter (PIA) [2], and serial data transfer using a 6850 asynchronous communications interface adapter (ACIA) [2].

Included in the PIA experiments are both input and output data transfers. Inputs are from switches or A/D converters while the outputs are used to drive LED's, stepper motors, or D/A converters. The first PIA experiment [3] is a simple one where the status of eight on/off switches are read through one port. These values are then output to eight corresponding LED's through the other port. This visually reinforces the operation of the PIA for the student.

It is the purpose of this paper to present a similar approach for presenting the properties of an ACIA. A brief overview of the ACIA will be given first.

ACIA Overview

The 6850 ACIA is Motorola's implementation of a universal asynchronous receiver transmitter (UART) [4]. It has four accessible registers, a receive data register (RDR), a transmit data register (TDR), a control register (CR), and a status register (SR). Both the RDR and the TDR are double buffered. Thus, for example, serial data received by the ACIA are clocked into a receiver shift register and then transferred into the RDR. The microcomputer can then access them there in parallel form. Analogously, to transmit data, they are sent in parallel form to the TDR. These data are then transferred to a transmitter shift register and clocked out in serial form.

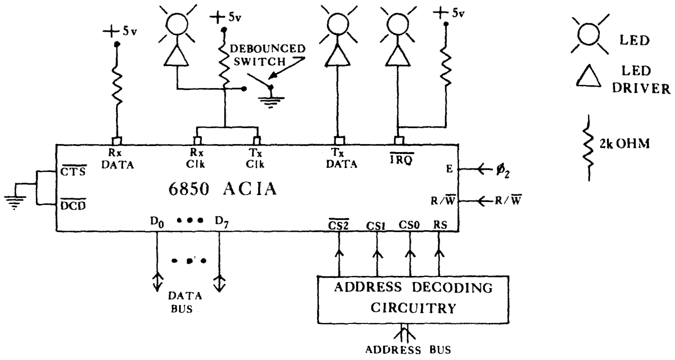

Fig. 1. ACIA wiring showing setup for TDR experiment.

The ACIA also adds a start bit, one or two stop bits, and a parity bit if desired. The parity bit may replace the most significant bit of the data byte or be appended to the data byte. These options are selected by setting or resetting certain bits in the CR through software. The setting of the CR also determines the baud rate (by dividing an applied clock signal), whether or not an interrupt output is enabled, and the value of a modem control signal ($\overline{\text{RTS}}$).

Various bits in the SR indicate if the RDR and TDR are full or empty, the value of two modem signals ($\overline{\text{CTS}}$ and $\overline{\text{DCD}}$), if an interrupt has been requested, and finally, if any errors have been detected in a received signal. The ACIA can detect three types of errors, framing, parity, and overrun.

It should be noted that the appended start, stop, and parity bits in a received signal are all stripped by the ACIA. Also, a user would not know which bits were received first, the most or least significant. Thus, by examining the contents of the RDR, a student can learn nothing about the structure of the serial data word that was sent to the ACIA.

However, from a pedagogical point of view, it is worthwhile for the student to see all the bits that would be transmitted by an ACIA. It is also useful for the student to be able to generate easily serial data streams containing the errors detectable by the ACIA. Introductory experiments that inexpensively achieve these goals are presented next.

A TDR Experiment

This experiment illustrates the structure of the serial data produced by the ACIA and their relation to the Tx clock signal. The importance of double buffering and the use of interrupts with the SR are also presented.

The ACIA is wired as shown[1] in Fig. 1 for this experiment. Using the microcomputer's memory change function, the CR is configured for the divide by 1 clock option and the transmitter interrupt is enabled (TDR empty will cause an interrupt). The number of stop and data bits, as well as the parity desired, is optional. Various combinations of these are tried by the students. A data word is then loaded into the TDR. Next, the student generates the input clock by opening and

Manuscript received June 13, 1980; revised November 26, 1980. This work was supported in part by the National Science Foundation under Grant SPI-7827173.

The authors are with the Department of Electrical Engineering, University of Maine, Orono, ME 04469.

[1] The LED's and LED drivers were on an LR-6 monitor outboard and the debounced switch was on an LR-7 pulser outboard, both from E & L Instruments, Inc.

0018-9359/81/1100-0287$00.75 © 1981 IEEE

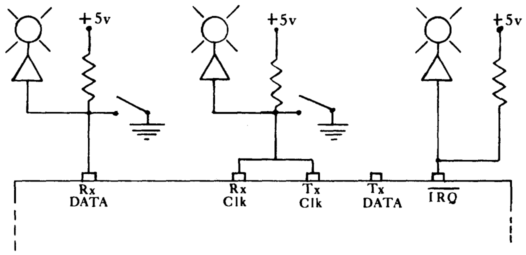

Fig. 2. ACIA wiring showing modifications from Fig. 1 for RDR experiment. Only Rx DATA and Tx DATA pins are affected.

closing the debounced switch. The Tx clock LED and the Tx data LED indicate visually the values of the clock and transmitted bits, respectively. Thus the start, data, parity, and stop bits can all be observed and compared to predicted values.

The $\overline{\mathrm{IRQ}}$ LED is used to indicate when the TDR is empty. Thus, the students can see that the TDR becomes empty as soon as its contents are transferred to the transmitter shift register. One then has the time it takes to serially shift out the data to load the TDR with new data. This illustrates the utility of double buffering. The student can also examine the SR as soon as $\overline{\mathrm{IRQ}}$ goes low and verify that the TDR-empty bit is, indeed, set. In this the student simulates the function of an interrupt handling routine polling to find the source of an interrupt. To make full use of double buffering, some such routine would have to be used.

An RDR Experiment

This experiment illustrates the structure of the serial data which the ACIA can receive and their relation to the Rx clock signal. The types of errors detectable by the ACIA and the importance of double buffering for receivers are also covered.

For this experiment the ACIA is rewired as shown in Fig. 2. Again the ACIA is configured for the divide by 1 clock option. However, now the transmitter interrupt should be disabled and the receiver interrupt should be enabled so that when the RDR is full, an interrupt will be generated.

The values set by the Rx data switch are clocked into the ACIA by opening and closing the Rx clock switch. This is very instructional, as the student must construct the entire incoming data word, including the start, data, parity, and stop bits. It also allows the student to intentionally (and sometimes unintentionally) generate parity, overrun, and framing errors. These are the errors which are detected by the ACIA and indicated in the SR. The student may then compare the contents of the RDR and the SR with predicted values.

The $\overline{\mathrm{IRQ}}$ LED is used to indicate when the RDR is full. The student is able to see that this occurs 1 clock cycle after a stop bit is clocked into the receiver shift register. Again a polling simulation is performed in that the SR is examined to verify that the RDR-full bit is set. Also, the utility of receiver double buffering is illustrated because the data in the RDR remain valid while the next word is being clocked into the receiver shift register.

It should be emphasized that since the student is generating the clock, the contents of the RDR and the SR can be examined at any point in the experiment. The TDR and the CR are write-only and thus can never be examined.

Laboratory Description

Heathkit ET-3400 microcomputer trainers are used in the above experiments to send data to the ACIA's and to read and display ACIA register contents.

The experiments are performed during one laboratory period lasting approximately two and a half hours. The students have the option of working alone or in groups of two. Most choose to work alone.

The ACIA experiments are performed near the end of the semester so that most of the topics mentioned in the Introduction have been covered. Most important for an understanding of the ACIA, interrupts and the PIA have already been discussed. The former is important for an appreciation of how various ACIA states may be used to interrupt a microcomputer and thus receive prompt handling. The latter is important for contrasting serial and parallel data transfers and also to introduce the concept of a programmable interface device.

Conclusions

Introductory experiments have been described which utilize a visual display of important signal values together with a "single-step" approach similar to that used in microcomputer debugging. These experiments are used to demonstrate the basic characteristics of an ACIA. They do not require expensive equipment (e.g., a logic analyzer) and so can be implemented at every lab station.

Performance on examinations indicates that the characteristics were learned very well. The students also were well prepared to handle more involved communication-link oriented laboratories. These were similar in nature to the problems presented in Southern [5]. Interfacing to a fiber optic link was also demonstrated.

The Motorola 6850 ACIA is specifically mentioned here. However, the techniques would be applicable for any similar device. Independent of any chip, they are also useful for demonstrating various serial data transmission characteristics, e.g., start, stop, and parity bits and various types of errors.

References

[1] L. A. Leventhal, "An example of a microprocessor curriculum," *IEEE Trans. Educ.*, vol. E-20, pp. 112-113, May 1977.

[2] *The Complete Motorola Microcomputer Data Library*. Motorola, Inc., 1978.

[3] *Heathkit Individual Learning Program in Microprocessors*. Benton Harbor, MI: Heath, 1977, pp. 10-67.

[4] J. B. Peatman, *Microcomputer-Based Design*. New York: McGraw-Hill, 1977, pp. 288-297.

[5] B. Southern, *Programming the 6800 Microprocessor*. Phoenix, AZ: Motorola Semiconductor Products, 1977, ch. 7, 11.

A Statistical Analysis of the Grades of B.S.E.E. Recipients

RAYMOND T. STEFANI

Abstract—**A statistical analysis is performed on 11 course grades for over 200 B.S.E.E. students. These students graduated from the California State University, Long Beach (C.S.U.L.B.) over a four-year period. The resulting means and correlations along with certain other data provide an interesting insight into academic tendencies. Some conclusions are drawn from these tendencies as to future industrial success and as to effective course development.**

Introduction

In the Department of Electrical Engineering, California State University, Long Beach, there are eleven courses which are offered to a rapidly changing population of students by a varying population of faculty. These eleven courses are required for the B.S.E.E. degree. Since 82 percent of the students

Manuscript received July 25, 1980; revised March 6, 1981.

The author is with the Department of Electrical Engineering, California State University, Long Beach, CA 90801.

0018-9359/81/1100-0288$00.75 © 1981 IEEE