KOMPUTER KORNER

PAUL E. FIELD, DAVID G. LARSEN, PETER R. RONY, and JONATHAN A. TITUS*

THIS MONTH, WE RETURN TO THE SUBJECT OF the substitution of software for hardware, *i.e.*, the substitution of machine-level routines and subroutines for specific digital hardware devices that store, manipulate, transmit, or receive digital information. The hardware device that we will discuss is the universal asynchronous receiver/transmitter, or UART—a 40-pin integrated circuit that contains an independent 8-bit asynchronous receiver and an independent 8-bit asynchronous transmitter. Data rates range from DC to 60,000 bits per second. The receiver and transmitter sections of the IC can be programmed for 5, 6, 7, or 8 data bits; 1 or 2 stop bits; even or odd parity; and parity or no parity. The IC contains a variety of flags.

An interface circuit for a simplified *software UART* is shown in Fig. 1. Owing to the nature of the specific application that the circuit was designed for, there was no need for special flag-bits or error checking. Thus, the interface circuit consists of a single three-state input buffer gate (SN74126), a single output data-latch (SN7474), two input device-select pulses, and one output device-select pulse. With appropriate modifications of the device select pulses, this circuit can be used with almost any microprocessor IC. In our case, an 8080A-based microcomputer operating at 750 kHz was used. This generates and detects, asynchronous serial ASCII-coded 5-volt TTL data. For teletype operation, additional hardware is required to convert the 5-volt logic levels to 20 mA current-loop operation.

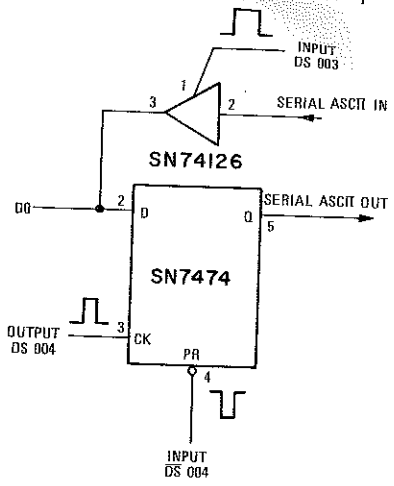

FIG. 1

TABLE 1—MICROCOMPUTER SUBROUTINE that demonstrates the asynchronous serial transmission of an eleven-bit ASCII word at a teletype speed of 110 Baud.

LO memory address	Instruction byte	Mnemonic	Description
•	•	•	
•	•	•	
•	Accumulator contains 8-bit ASCII word. Bit 8 is the parity bit that can be set for even or odd parity, or no parity.		
144	056	MVI L	Set ASCII word bit counter to 013
145	013	013	
146	267	ORA A	Set carry bit to logic 0
147	027	RAL	Rotate carry bit to DO in accumulator
150	323	OUT	Output carry bit to SN7474 latch
151	004	004	
152	315	CALL	Call 9.09 ms time-delay subroutine
153	'B2'	'B2'	LO address byte of time-delay subroutine
154	'B3'	'B3'	HI address byte of time-delay subroutine
155	037	RAR	Rotate bit in ASCII word to DO in accumulator
156	067	STC	Set carry bit to logic 1
157	323	OUT	Output bit to SN7474 latch
160	004	004	
161	055	DCR L	Decerement bit counter by 1
162	302	JNZ	If bit counter has a value of zero, ignore this instruction. If all of the bits in the 11-bit ASCII word have not yet been transmitted, jump to address LO = 152 above.
163	152	152	LO address byte
164	'B3'	'B3'	HI address byte
•	At this point, the 8-bit ASCII word contained in the accumulator has been transmitted. Two stop bits have been added at the end of the eight bits and a single start bit, at logic 0, has been added at the beginning of the eight bits.		
•			
•			
•			

*This article is reprinted courtesy American Laboratories. Dr. Field and Mr. Larsen, Department of Chemistry, and Dr. Rony, Department of Chemical Engineering, are with the Virginia Polytechnic Institute & State University. Mr. Titus is president of Tychon, Inc. Dr. Field is guest author of this month's column.

continued on page 24

continued from page 22

Transmit subroutine

The transmit subroutine, shown in Table 1, for the software UART occupies twenty to twenty-five successive program steps in memory once the appropriate PUSH, POP, and RET instructions have been included. Also required is a 9.09 ms time-delay subroutine that corresponds to an asynchronous serial ASCII data transmission rate of 110 Baud, i.e., teletype speed. The program in Table 1 can be described as follows:

Register L is used as the bit counter for the 11-bit ASCII word, and is set initially to octal 013. The seven data-bits plus the parity bit, which is Bit 8, are assumed to be present in the accumulator. At the LO memory address 146, the accumulator is OR'ed to itself to clear the carry bit (shown on the far left in Fig. 2.) In Fig. 2, the least significant data bit

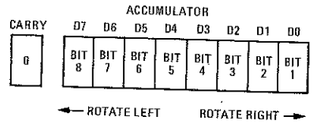

FIG. 2

is Bit 1. At address LO = 147, a RAL instruction is performed to rotate the start bit to bit position DO in the accumulator. Fig. 3 should provide you with assistance in understanding the four different rotate instructions in the 8080A microprocessor instruction set.

At address LO = 150, the start bit is output to the SN7474 data latch. The program then goes into a 9.09 ms time-delay subroutine, after which Bit 1 is rotated into

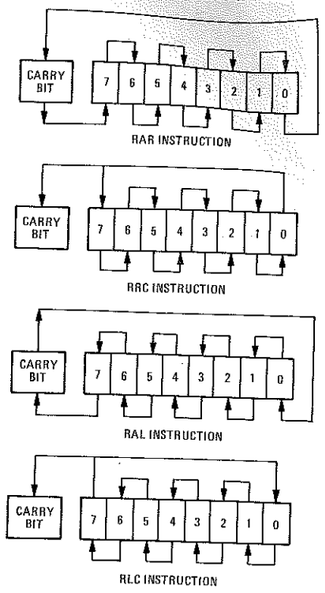

FIG. 3

the DO accumulator position and the carry bit is set to logic 1. Bit 1 is fed to the SN7474 latch, the ASCII word bit counter in register L is decremented and program control is returned to the time-delay subroutine that is called at address LO = 152. The loop from LO = 152 to LO = 164 is executed a total of eleven times, after which register L becomes zero and the JNZ instruction at address LO = 162 is ignored.

A software UART transmit subroutine possesses a flexibility equivalent to the original 40-pin UART chip. With appropriate modifications to the program or the original accumulator data, you can transmit 5, 6, 7, or 8 data bits; 1 or 2 stop bits; even or odd parity; and parity or no parity. The time-delay subroutine can be modified so that you can transmit at data rates from 60 to 9600 Baud for a 750-kHz clock rate and higher for 2-MHz and 4-MHz clock rates.

The conversion from one data transmission rate to another is easily accomplished with the aid of appropriate software time-delay subroutines that replace R–C time-constant circuits. An additional advantage that is gained from the use of software is the potential to perform code conversions. For example, 5-level Baudot KSR machines are in widespread use and can still be obtained for under $50. It is not too difficult to develop software that converts ASCII to Baudot and thus produce an inexpensive hard-copy terminal for the laboratory scientist, engineer, ham or computer buff.

Receive subroutine

The software UART receive subroutine requires 50 instructions and will not be repeated here. (Copies of the transmit and receive subroutines and a description of the

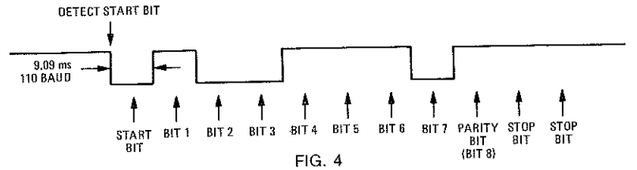

FIG. 4

smart data-entry station are available from Professor Paul Field, Department of Chemistry, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.) The basic programming concepts associated with the receive subroutine are shown in Fig. 4, which represents an 11-bit asynchronous serial ASCII word that is being detected by the 8080A-based microcomputer with the aid of the SN74126 three-state buffer gate shown in Fig. 1. The program repeatedly tests the serial ASCII input line for a logic 0 state. Once a logic 0 state is detected, which corresponds to a start bit, the program goes into a 4.54-ms wait loop. Upon leaving the wait loop, the program again inputs the logic 0 into bit position DO in the accumulator, thus testing the validity of the start bit. The start bit is rotated to the carry bit and the program then enters a 9.09-ms wait loop, after which it inputs Bit 1 into position DO in the accumulator. Register H is used as the SAVE register that stores the growing ASCII data word. The SAVE register is rotated one position, and the 9.09-ms wait loop is again entered, after which Bit 2 (a logic 0 in Fig. 4) is input into bit position DO in the accumulator. The input of successive data and parity bits continues until the entire 8-bit data word is entered into the SAVE register. The two stop bits are also detected. With appropriate modifications, the program can detect parity or framing errors or an overrun condition. A data-ready flag signal can also be generated from software with the aid of a second SN7474 latch.

Applications

The above software UART routines were used in a "smart" remote data-entry station that was tied via a 20 mA current-loop to a PDP 8/L minicomputer in a physical chemistry laboratory. The data entry station intercepted the 20-mA teletype current-loop tied to the minicomputer. The remote data-entry station permitted students to load data into memory and then transmit it as a block to the minicomputer, which analyzed the data and provided a print-out. With the 20-mA current loop operating in the full duplex mode, ten or more remote data-entry stations could be tied to the minicomputer.

This column provides a good demonstration of the software-hardware tradeoffs that can be accomplished using microcomputers. Similar, and perhaps more comprehensive, routines have already been written for all of the popular microprocessor chips, such as the 16-bit PACE or the 8-bit 6800. The faster and less expensive that microcomputers become, the more likely that all moderate speed digital functions will be executed via software. The theme of software replacing hardware is an important one, and we will return to it many times in future columns. **R-E**

continued from page 67

able A and the second value to B. Line 20 assigns to variable C the value of the square root of the sum of the two entered values to the third power. Line 30 outputs the value of C to the terminal. Line 40 transfers execution back to line 10, and you're ready to execute the problem for two new values.

When you have all the results you require, you push the ESC key (escape) to stop the program, then sign off or enter a new program.

Notice the similarity between BASIC and English:

- INPUT—to input a value for a variable
- LET—to let a variable equal a value
- PRINT—to print the results at the terminal
- GOTO—to go to another part of the program.

This similarity exists throughout the BASIC language, which makes it ideal for people who are not computer programmers but need or desire to use a computer. **R-E**