A Microcomputer-based Logic Signal Generator and Recorder/Analyzer for a Beginning Digital Electronics Laboratory

J. Sherwood Charlton, Jeffrey J. McFarland, James D. Shuffield
University of Arkansas

The Electrical Engineering Program at the University of Arkansas is in the process of integrating laboratories with each required course instead of having a separate sequence of laboratories. This creates some problems with regard to our beginning digital systems course. This course covers binary and complement arithmetic, Boolean algebra, combinational logic, sequential logic, and the various digital IC'S simpler than microprocessors. The course includes the study of many of the TTL, and their CMOS equivalent, digital IC's including terminal properties and other data sheet information.

The beginning digital systems course is arranged so that it can be taken as a corequisite with the first electric circuits course; portions of the digital course requiring Kirchhoffs laws, Ohm's law and first-order RC circuits are presented after those topics are covered in electric circuits.

Coming this early in the student's electrical engineering education, the student has not been exposed to pulse generators, logic analyzers, or even oscilloscopes. Normally under these conditions, a digital trainer would be used. Most digital trainers provide only a clock waveform, a certain number of switches to supply steady logic levels and perhaps a push-button switch as inputs to a test circuit, To record the output(s) of the test circuit, LED's are provided. While this is adequate for combinational logic circuits it is a poor substitute for studying sequential logic circuits and provides no permanent record.

It was desired to have a extremely user-friendly, programmable, logic-level signal generator with several outputs and the ability to simultaneously record several digital signals. It was also desired to keep this system inexpensive and to have a permanent record of test circuit input/outputs signals.

SOLUTION

Our solution is based on the ready availability of outdated computers and dot-matrix printers. As the computer labs have been upgraded, a surplus of unused computers and printers has occurred. This equipment was destined for storerooms or surplus property offices. The system described herein is based on 80286 computers with EGA graphics, but could be adapted to more or less powerful platforms. The printers are generic, ink-ribbon, dot-matrix printers.

As can be seen in Figure 1, the students may define up to 8 inputs to his test circuit which is built on a prototyping board. Logic levels of up to 8 points in his circuit maybe recorded and displayed on the computer screen and printed, The interface between the computer and the prototyping board is through a parallel port of the computer, The student uses primarily the cursor keys, the 1 or O key for a TTL 1 or O to the test circuit from the computer on any one of 8 lines of the parallel port. the execute function key, and the print screen key, Some details of the system are presented in the next two sections,

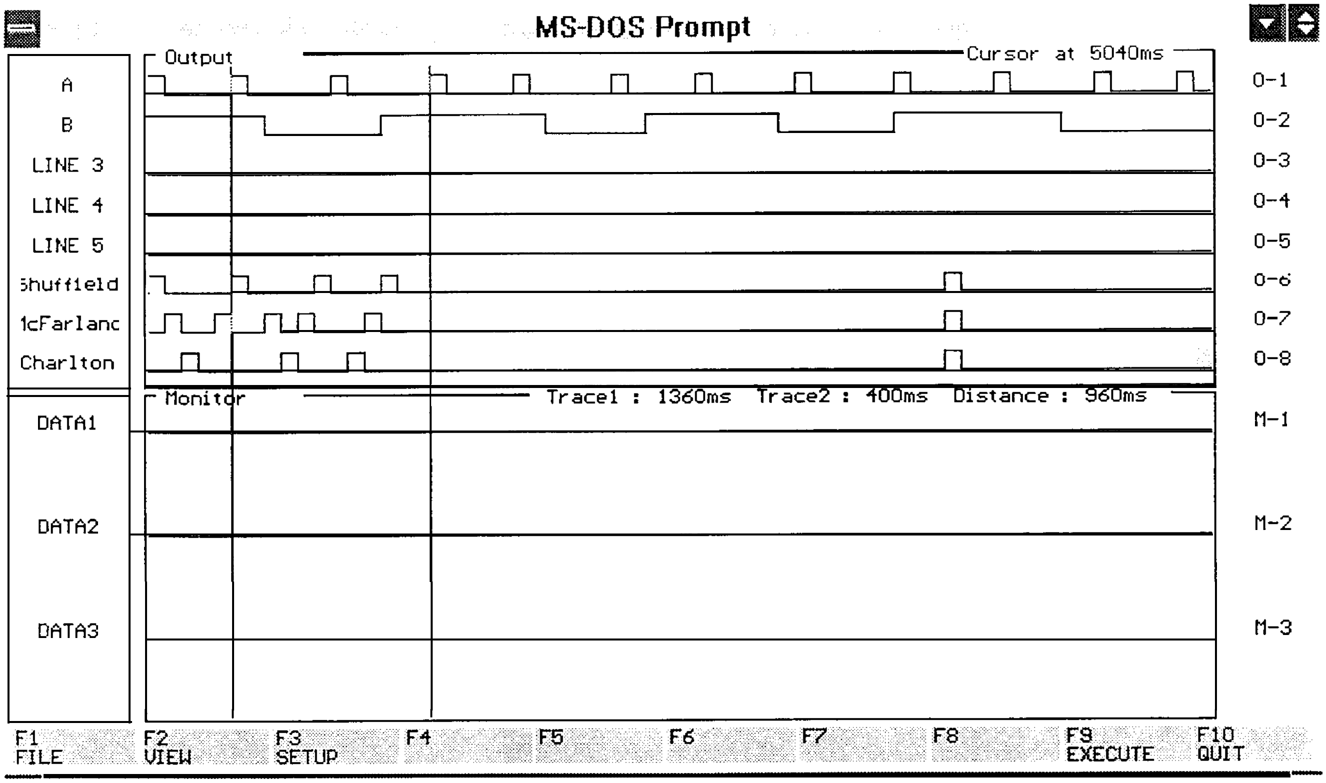

Figure 1. Graphical User Interface

HARDWARE DESIGN

A small (less than 6 x 5 x 1.5 cm.) enclosure plugs into the prototyping board. It contains 16 buffer/line drivers (2 74LS244'S) to isolate and protect the computer's parallel port, The buffers have Schmitt-trigger inputs and three-state outputs which are wired enabled. A DB25 connector is mounted on the enclosure for the cable to the parallel port. The schematic is shown in Figure 2. The student gains access to the generated waveforms and recording channels by inserting wires into the proper wiring holes on the prototyping board. Information on wiring and configuring the parallel port was obtained from personal computer technical references [1,2]. The Data Bit lines of the parallel port carried the generated waveforms from the computer to the 74LS244 inputs and parallel port control lines were used to monitor logic levels in the test circuit.

The only other hardware required is a second parallel port or an A/B switch for the printer. We found that extra parallel port cards for our computers were no more expensive than A/B switches and do not require the student to remember to switch from the prototyping board to the printer.

SOFTWARE DESIGN

The program is both very powerful and easy to use, with a learning curve consistent with the knowledge of any student in an introductory digital electronics class. The software has two main components: A user-friendly Graphical User Interface (GUI) and a parallel Input/Output interface.

The GUI is broken up into three fictional areas: Output/Monitor waveform display, Output/Monitor Label display, and the Command/Menu display for ease of understanding and display of all information needed to the student. The GUI is completely designed using standard C EGA graphics functions. Figure 1 shows a screen shot of the GUI.

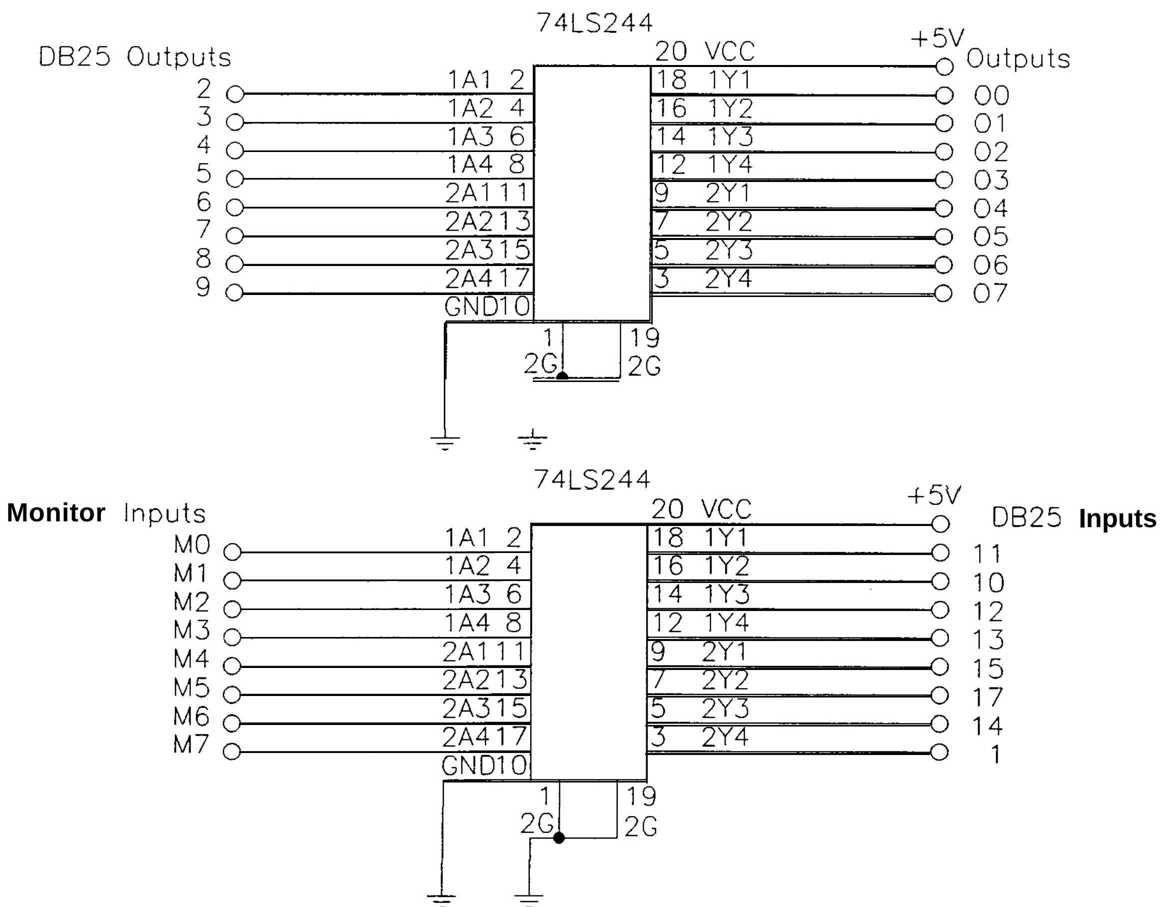

Figure 2. Schematic Diagram

The Output/Monitor area forms the basis for the logic signal generator and the logic recorder/analyzer. This area is divided into two display boxes, Output (referring to the computer generated signals serving as inputs to the test circuit) and Monitor (the recorded signals from the test circuit), each containing up to **8** waveforms. These waveforms are displayed much like the timing diagrams that the student would draw in an introductory digital electronics class. The Output waveforms consist of 80 time dices, each 80 ms. wide, for a total possible output time of 8 seconds. The Monitor waveforms are 8 times oversampled so they consist of 800 time slices each 8 ms. wide for a total possible monitor time of 8 seconds. Two trace cursors, which are simultaneously displayed in both the Output area and the Monitor area, are manipulated using the J,K,N, and

M keys. J and K move trace cursor 1 left and right, respectively. N and M move trace cursor 2 in the same manner, A display which shows the current position of each trace cursor in milliseconds and the difference between the two trace cursors in milliseconds is also provided. Each cursor is a different color for ease of distinction, An Edit cursor is provided in the Output section for editing of Output waveforms as well as a display showing its current position in milliseconds. LEFT and RIGHT arrow keys move the cursor left and right on the waveform in 80 ms. increments, while the 1 and O keys set the current time slice to logic 1 or O, respectively. The whole 8 second waveform is not displayed on the screen, so scrolling is used to move the waveform according to the movement of the edit cursor.

C EGA graphics functions were too slow for this task, so direct video memory reading and writing is used. When finished editing one waveform, the student is able to move from one waveform to another by using the UP and DOWN arrow keys. The TAB key moves the TAB focus from the Output/Monitor waveform display to the Output/Monitor Label area,

The Output/Monitor Label area is to the left of the Output/Monitor waveform display and allows for associative labeling of the waveforms. This area is divided into two display boxes, Output and Monitor, each containing 8 associative labels, 9 ASCII characters in length. A cursor is provided to show which label is the current label for editing and can be moved with the UP and DOWN arrow keys, Editing of the labels is accomplished using the DELETE key to erase the current label and any ASCII key to reenter the label. The TAB key changes the TAB focus from the Output labels to the Monitor Labels, and then back to the Output waveforms, Editing is only possible in the area that currently has the TAB focus, The Command/Menu display, however, is available at all times to the student.

The Command/Menu display consists of 10 blocks at the bottom of the screen which represent the function keys F1-F1O. Text on the block associates the function key with its function.

The Command/Menu system is implemented using arrays of 10 menu structures, each containing a title for the menu block and an associated function, for each level of the menu system. The associated function can be either next_level (a function to display the next menu level), deadfunction (a function that does nothing) , or the actual command function.

The Top-level menu contains FILE, VIEW, SETUP, each of which are next menu levels, and EXECUTE and QUIT, which are both commands. The EXECUTE command outputs the Output waveforms and samples the Monitor waveforms using parallel Input/Output. A warning message is displayed in the Monitor waveform display area to tell the student the waveforms are being output and monitored, The QUIT command allows the student to exit the program.

The File-level menu contains the OPEN and SAVE command functions which allow the student to either load a previously saved session, or save the current session. Loading a previous session sets the Output waveforms and Output/Monitor labels to the values stored in the saved session. The TOP next level function contained on this level returns the student to the Top-level menu.

The View-level menu contain the ZOOM, HOME, and END command functions which manipulate the display in various ways. The ZOOM command enables the student to change the zoom factor of the Output/Monitor waveform displays to one of 4 different levels. This function redraws the waveforms at a magnified resolution for a closer look at transitions. The HOME and END commands move the cursor to the beginning or end of the waveforms, respectively, and scrolls the screen accordingly.

The Setup-level menu contains OUTPUT and MONITOR next level functions which take you to Output-toggle level menu and Monitor-toggle level menu, respectively. These levels contain 8 functions which toggle the current on or off status of the 8 waveforms in the Output and Monitor display area. Waveforms can be turned on or off to make the waveform and label displays less cluttered and also to remove unused waveforms from the display.

The parallel Input/Output interface consists of the output of Output waveforms and the sampling of Monitor lines. This interface also contains the hardcopy function that allows the screen shot of Figure 1, To understand this function a small understanding of the code and the hardware involved is required.

Simultaneous input and output accross the parallel port is accomplished by using a hi-directional mode and the readback buffer on the parallel port. When the student uses the EXECUTE command, information about the timing diagram the student has drawn is converted into bytes which are then output through the parallel port to the circuit of interest. This value is latched for 80 ms. while 8 data bytes are read back into the parallel port at 10 ms. intervals. These bytes are split according to the proper line and the values, logic 1 or O. and are displayed to the Monitor waveform display as a timing diagram. This process is repeated 80 times to output and monitor the whole 8 second time interval. When the process is completed the waveforms are displayed to the screen in the appropriate area ready to be analyzed or edited by the student.

The final parallel port function is actually not part of our code. A shareware terminate and stay resident program that prints EGA graphics screens when the Shift-Print Screen keys are pressed is used to print a hard copy of the current EGA graphics screen as in Figure 1, This feature allows students to have a permanent record of their session, that can be analyzed in the future or offered as an assignment.

CONCLUSION

Since cast-off computers and printers were available to us, we have been able to provide to our students, a logic signal generator and recorder/analyzer for about $25 a setup. Our students perform experiments parallel to material presented in their texts and in lectures in a format similar to this material, They have had no trouble testing combinational logic circuit by just defining all possible combinations of the inputs on their GUI, executing and then recording their test circuit's output(s) for these combinations, Sequential circuits have proved even easier. The student "draws" the clock waveform, initialization signal, etc. and then applies them to the test circuit and records the results. We have found that most aspects of the course can be studied in the laboratory with the exception of the effects of nanosecond propagation delays,

1. IBM Corp., IBM Personal Computer Technical Reference Manual, Boca Raton, FL: IBM Corp., 1981.
2. Stewart, Zhahai, Interfacing the IBM PC Parallel Printer Port, version 0.96, http: //rainbow.rmii.com/-hisys/parport.html, 1994.

J. SHERWOOD CHARLTON, Ph. D., is Associate Professor of Electrical Engineering, Department of Electrical Engineering, BELL-3217, University of Arkansas, Fayetteville, Arkansas 72701. EMail: j sc@engr.uark. edu

JEFFERY J. McFARLAND, M. S. E. E., is a Member of Technical Staff, Symmetrix, Inc., 1301 Capital of Texas Hwy. South, Suite C200, Austin, TX 78746. EMail: jeff@symtx.com

JAMES D. SHUFFIELD, B. S. E. E., is a graduate student and instructor in the Department of Electrical Engineering, BELL-32 17, University of Arkansas, Fayetteville, AR 72701. EMail: jds3@engr.uark.edu