# Integration of Virtual Instrumentation into a Compressed Electricity and Electronic Curriculum

**Arif Sirinterlikci**

**Ohio Northern University**

## Background

Ohio Northern University Technological Studies Department has a Technology Program that offers Industrial Technology curriculum under its Industry Track. Students in this track, take only two technology courses (TECH 261: Fundamentals of Electricity and Electronics, TECH 362: Digital Electronics: Concepts and Applications) relating to electricity and electronics before taking higher-level courses with automation and robotics emphasis. The curriculum is very compressed and it is a challenge for the instructor to establish a healthy and balanced base of theory and practice.

Previously the department owned out-dated electronics workstations (experimenters) and a simulation package that was not current and suitable for integration with hardware. Since practice is an important part of the program just like any other technology program, laboratory activities took a good portion of the two courses mentioned above. There was limited time available for simulation, hence the students lacked computerized design and analysis skills. This paper elaborates on the efforts of improving the quality of electricity and electronics education with the help of simulation and virtual instrumentation tools.

The author obtained 9 NI (National Instruments) ELVIS (Educational Laboratory Virtual Instrumentation Suite) workstations seen in Figure.1 to be utilized in both courses for a maximum number of 18-student laboratory sessions. Each workstation has a prototype board, variable power supply, function generator, physical digital multi-meter (DMM) and oscilloscope interface, and communicates with a PC through a data acquisition board (PCI card). The power supply and function generator can be controlled manually or by corresponding virtual (software) instruments in PC environment. Other virtual instruments available include DMM, oscilloscope (shown in Figure.2), Bode analyzer, dynamic signal analyzer, arbitrary waveform generator, digital bus reader and writer, impedance analyzer, and two- and three-wire current-voltage analyzers. During the acquisition of this hardware, a major design, simulation, and data acquisition software (NI LabView) was obtained as well.

Approach towards Integration

With utilization of data acquisition ability, the newly acquired workstations are interfaced with the LabView software allowing real-time monitoring and control of electrical/electronic circuits or mechatronic systems enhancing physical capabilities of the program. A new teaching

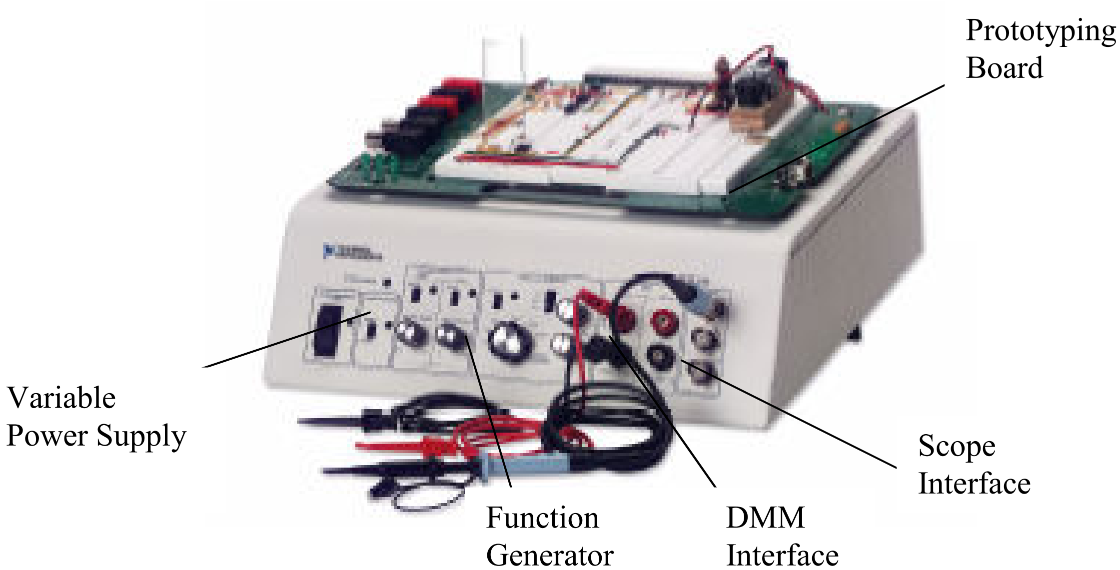


Figure.1 NI ELVIS hardware[1]

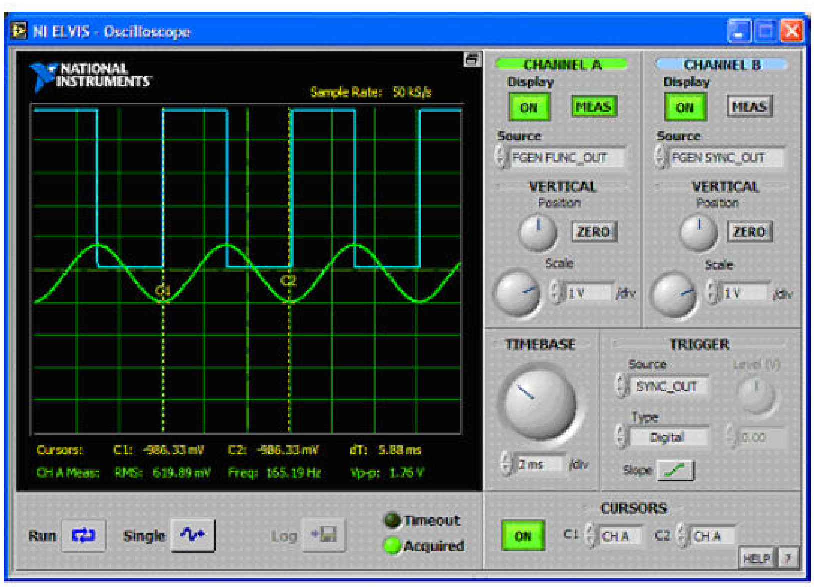


Figure.2 Virtual oscilloscope[1]

approach for both courses was proposed and is currently being implemented. It is based on integrated study of design, simulation, execution and monitoring, and troubleshooting of circuits or systems. The following sequence is followed in the learning process.

- Students design their circuits/systems in LabView and simulate them for justification of proper functioning.
- They build the actual physical circuits/systems and run them on the workstations.
- They monitor and troubleshoot their circuits/systems by using virtual instruments of the NI ELVIS software interface by connecting to the physical interfaces on the front panel of the workstation.
- They can also use the monitoring capability through data acquisition to obtain further information to be processed in LabView or as feedback for real-time controls.

In their laboratory assignments, students still use the actual physical instruments such as oscilloscopes, meters, logic analyzers or probes. However, they can now solve their problems in a more observable environment and swiftly obtain more information during the design process leading to shorter lead times and less burden in their and the instructor's part. Since the software and hardware are based on the same technology, both elements are very easy to learn and employ simultaneously.

Laboratory Experiences

This section of this study presents sample laboratory exercises representing some important components of the approach. Examples are mainly taken from TECH 362: Digital Electronics: Concepts and Applications since the author recently taught the course, and development efforts are continuing in full-force for that one.

First example is about justification of the results of objective design process. Figure.3 shows an automatic optical gauging station[2]. Task at hand is to design a simple digital control circuit using fixed-IC's. Parts to be inspected move on a conveyor while approaching a light curtain made from 3 light beams serving as inputs to the control circuit. Any broken beam results in logic 1 for the corresponding input signal while unbroken beams return logic 0. Part rejection (R= logic 1) mechanism is driven by the output of the control circuit. Parts that break the top beam (labeled as G for go) are deemed as either too tall or at wrong orientation, and they are rejected. Parts that are too short to break the second beam (labeled as N for no-go) are deemed as out of tolerance or at wrong orientation, and they are rejected as well. The latter type only breaks the part presence beam (labeled as P). Parts breaking the P and N beams are allowed to continue. The rejected parts fall down through the chute opening as the door opens. The student is asked to create the truth table behind the system at the first step of the design process as seen in Figure.4.

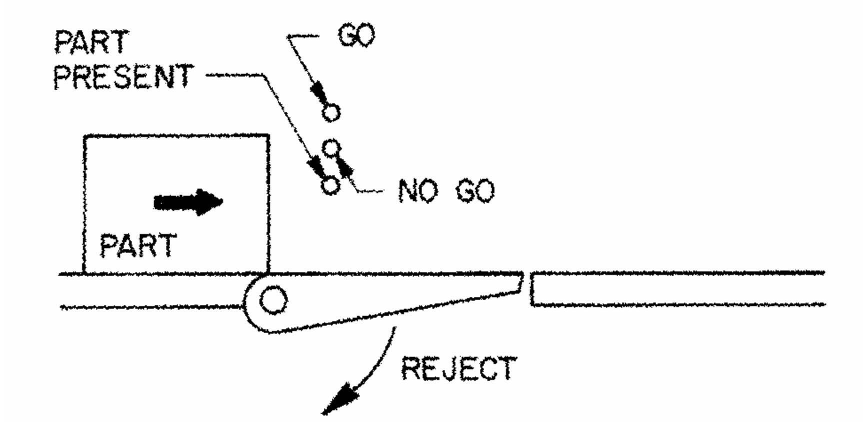


Figure.3 Automatic gauging station

| **Inputs** | | | **Output** |
|---|---|---|---|
| **G** | **N** | **P** | **R** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | DC |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | DC |
| 1 | 0 | 1 | DC |
| 1 | 1 | 0 | DC |
| 1 | 1 | 1 | 1 |

Figure.4. Truth table for the automatic gauging station (DC: DON'T CARE)

Having three inputs, there are eight scenarios (outputs) or eight possible input sets. When the system is functioning properly only the four described conditions will occur. Rest is deemed as DON'T CARE's and represents unexpected conditions which will most likely not to occur. 0's could be assigned as initial values to these conditions. Applying Sum-of-Products method to the truth table, the Boolean expression $R = \overline{G}x\overline{N}xP + GxNxP$ is obtained, but not simplified. The next step is to justify this model's adherence to the gauging station's control logic or truth table in LabView. Figure.5 is the front panel of the LabView model indicating three push-button switches simulating the three light beams and an LED for the rejection output. Once these four elements are placed on the front panel, the student switches to the block diagram environment. The digital control circuit is then generated by using AND, OR, and NOT gates leading to Figure.6. Data flow during the execution of the program for all inputs (G, N, and P) having logic 1 state can be seen in Figure.7 to lead R to logic 1.

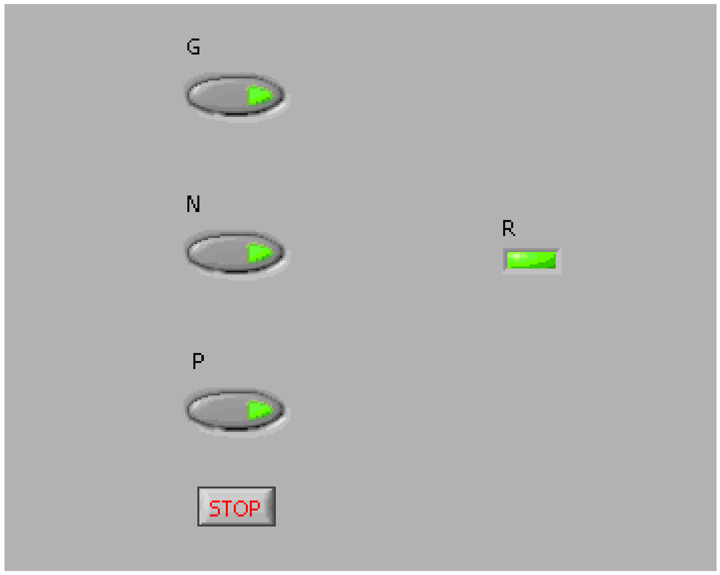


Figure.5 Front panel for the automatic gauging station model

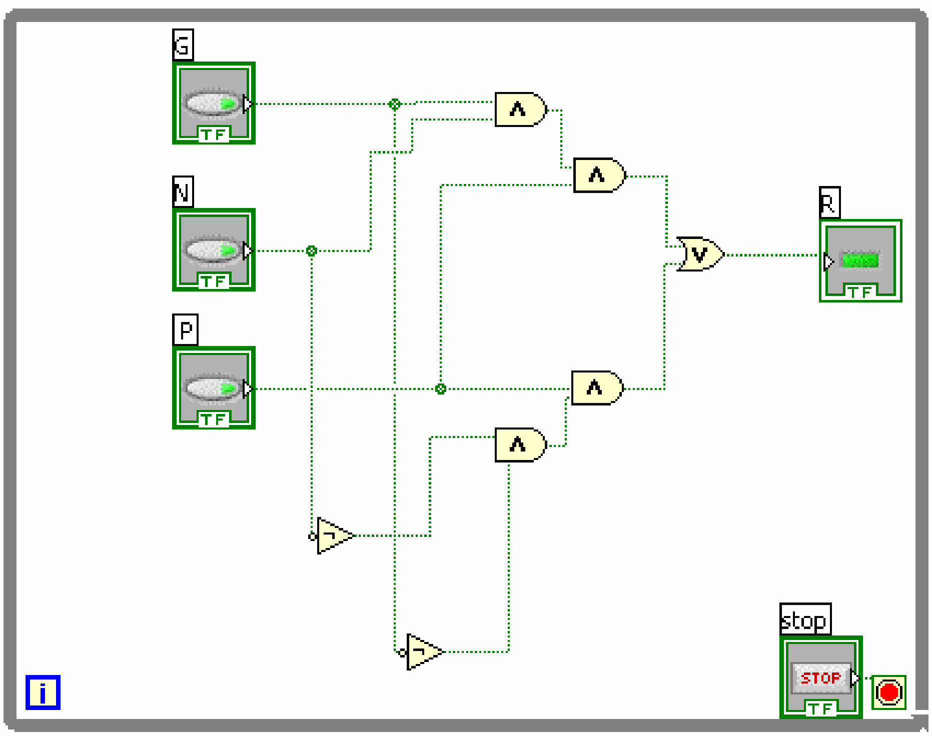


Figure.6 LabView block diagram for the automatic gauging station model

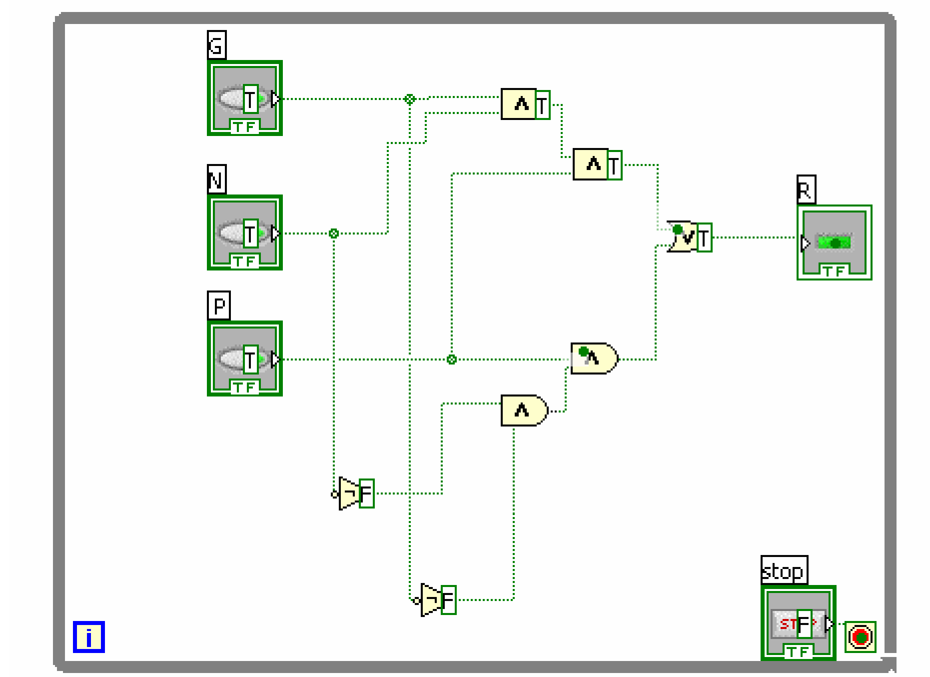


Figure.7 Data flow within the LabView model during execution

Second example is about understanding the behavior of complex digital components such as latches or flip-flops [2]. When a pocket calculator is used to add two numbers, the numbers are entered one after the other. Once the first number is entered, it must be stored within the circuit until the second is entered. Then the addition takes place. The sum must also be stored so it remains displayed. Combinational Logic circuits such as the one seen in the first example are utilized in the addition process. However, a different type of logic called Sequential Logic must be employed to accomplish the storage. Logic states of such circuits are dependent upon the order of application of inputs. A simple example of such a circuit can be found in the manual push-button electric motor starters. These circuits are used to start and stop electric motors, a memory element must be provided by a relay and its contacts. The circuit remembers whether the START (Set) or the STOP (Reset) button was pushed last. This can be represented by the following truth table in Figure.8. The same table represents a SR flip-flop/latch integrated circuit. This IC memory element flips and flops between two states, logic 1 or 0, or ON or OFF.

| **Inputs** | | **Output** |
|---|---|---|
| **S** | **R** | $\mathbf{Q_{t+1}}$ |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | $Q_t$ |
| 1 | 1 | * |

Figure.8. Truth table for SR flip-flop/latch (*: neither defined nor acceptable, $Q_t$: Q remains as previous value) for active high inputs

Figure.9 and .10 represent the two ways of building SR flip-flops/latches in LabView environment. Because of the nature of such components, feedback of signals has to be accomplished by either of the two means, feedback nodes or shift registers. Figure.9 indicates a feedback node (an arrow) used with only one NAND gate within the circuit. Circuit in Figure.10 returns the feedback values over the While Loop through arrow heads representing shift registers. Before building SR flip/flops students are asked to simulate the two circuits making it easier for them to visualize the importance of feedback and inner workings of the SR flip-flop.

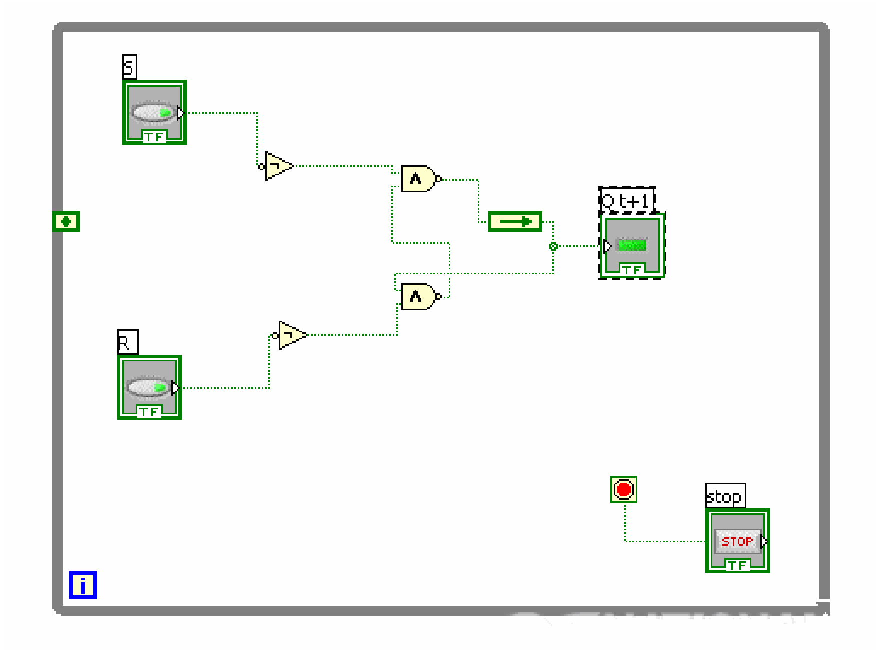


Figure.9 SR flip-flop circuit accomplished by feedback nodes

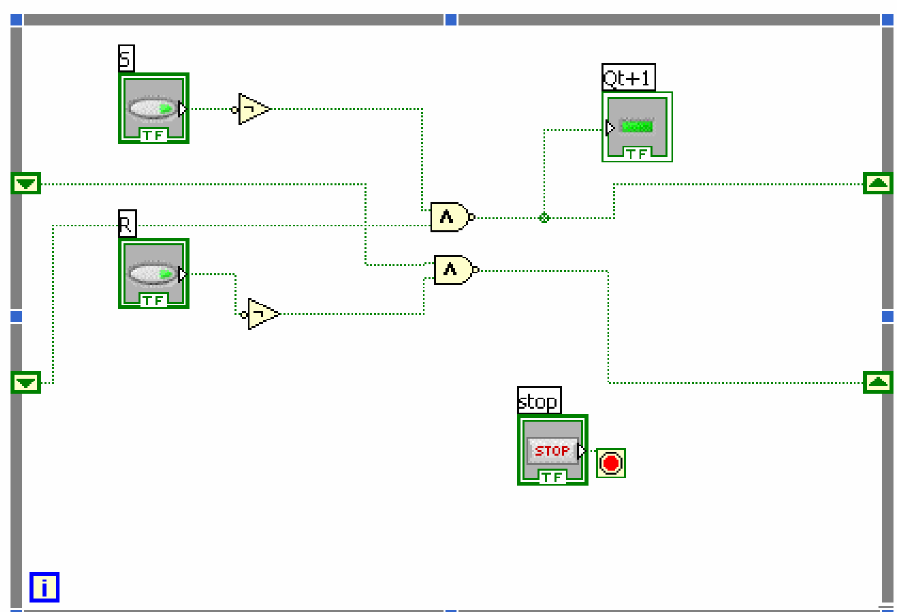


Figure.10 SR flip-flop circuit accomplished by shift registers

The third example utilized in the class is a modified version of a laboratory exercise from the NI courseware[1]. Its main objective is to create a timing diagram modeling a logic analyzer and run it within the data acquisition application. First students create a 4-bit binary counter by using a clock source, a 555 timer chip and related equipment (a set of two resistors and a capacitor) and a 7493 TTL counter chip. The four outputs and the clock signal of the 7493 are plugged into read pin sockets on the ELVIS prototype board. In the block diagram seen in Figure.11 *DIO initialize virtual instrument* initializes the read function on device 1 (ELVIS) and creates a refnum shown in green. The *DIO read virtual instrument* reads the parallel port (read pin sockets). The *DIO close virtual instrument* closes the operation to limit the memory utilization and to pass error messages onto the front panel. After reading the port, the program presents the bits as a numeric number shown in blue line. LabView then converts the numeric number into an 8-bit Boolean array shown in thick green line. Bits on the port are mapped to the elements (through matching indexes) of the array. The index array virtual instrument extracts a particular bit as in index 5 and sends the bit to the B1 and then onto the plotting function. Each Boolean bit is converted back to a numeric value (0 or 1) then bundled together with the others for plotting of the timing diagram. The result is the timing diagram shown in Figure.12.

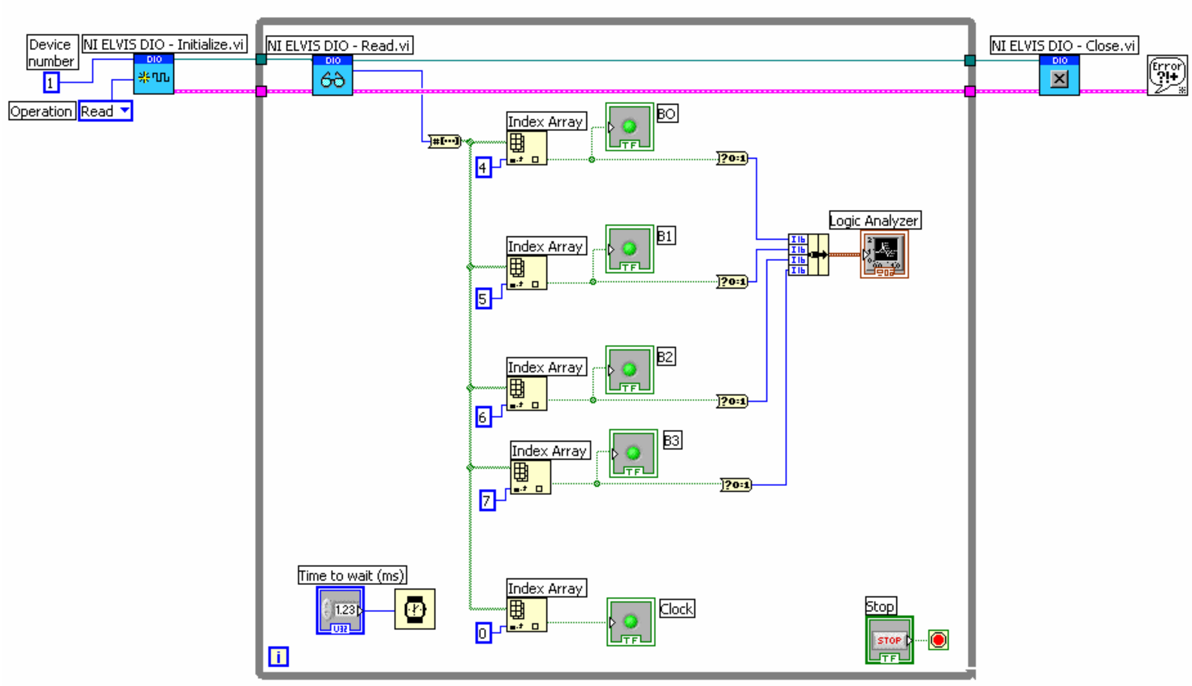


Figure.11 Data acquisition example

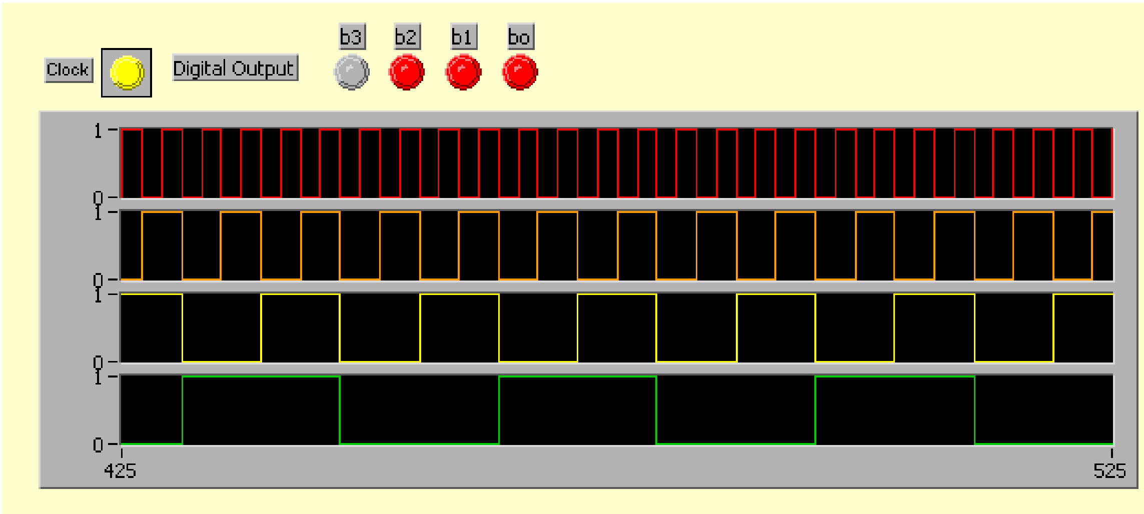


Figure.12 Timing diagram for the data acquisition example

Various other examples, especially in controls are currently being developed. These include variations of the exercises obtained from various resources available in NI coursework[1]. Examples such as PID (Proportional Integral Derivative) Control of DC motor rotational speed (rpm) or 2-way Stoplight Intersection Auto-Control are included within the newly developed exercises like other analog and digital control experiences that are taken from or inspired by real-life cases such as the automatic gauging station in the very first example of this paper.

Conclusions

Most of the today's engineering and technology student is computer oriented. Using a computer based system actually triggered student interest with little effort and the instructor was able to maintain student involvement at high levels. Students did enjoy using virtual instruments which are very new tools to them.

In terms of the learning experience, the addition of modeling, data acquisition and control abilities greatly enhanced the delivery of the courses mentioned in this paper. Student feedback showed that most of the students were able to visualize and justify the circuits before they build them. They were also able to easily troubleshoot the circuits due to accessibility of data from the circuits cutting lead times of design drastically. Students were able to understand complex digital components such as flip-flops, memory and ALU circuits with ease. But more importantly, since the hardware and software were based on the same technology and interfaced together, it enabled the instructor to utilize these new components with less effort and time consumption from students compared with teaching a different software and hardware based on different technologies.

Bibliography

[1] Paton, B., *Introduction to NI ELVIS Courseware*, National Instruments, Austin, TX, 2000.
[2] Streib, W., J., *Digital Circuits*, The Goodheart-Willcox Company, Tinley Park, IL, 1997.

Biography

ARIF SIRINTERLIKCI is a faculty member at Ohio Northern University Technological Studies and Honors Programs. He holds a Ph.D. degree from Industrial & Systems Engineering Program of the Ohio State University and M.S. and B.S., both in Mechanical Engineering from Istanbul Technical University, Turkey. His previous work experiences include various appointments and projects in Mechanical and Manufacturing Engineering fields.