FPGA Data Acquisition of Electrical Parameter

Kennedy Ahamefula Iroanusi

Abstract — A data acquisition system using a Programmable Logic Gate Array (FPGA) and Graphical User Interface (GUI) for visual enhancement designed for Personal Computer is shown. The data acquisition of voltage (V), current (A) and temperature ($\theta_C$) signals and/or parameters transmitted at high frequency in real time via the system-on-chip (SOC) created on Spartan 6 FPGA. The system-on-chip is achieved by programming the FPGA with a high-speed hardware description language (Verilog) code written for the system. Printed Circuit Board (PCB) was designed for the system and the GUI has been created using a graphical approach utilizing LabVIEW to enable data monitoring on Personal Computer (PC) display. The FPGA requires digital input signals; therefore, an analogue to digital converter (ADC) is required for the convert sensor data from analogue signal from sensors to digital signals. A voltage level shifter is required to normalise the voltage level standards within the circuitry in between the 5V from the ADC converter and the 3.3V voltage requirement for the FPGA. The Spartan 6 FPGA receives data from the analogue sensors via the ADC, the data are wrapped up in packets and transmitted through RS-232 serial port to the PC. The three aforementioned parameters are monitored on the GUI on the PC presented in both numerical and graphical format and all data can be store in a file for backup storage, maintenance or reference purposes.

Index Terms — FPGA, GUI, VHDL, UART, ADC, LabVIEW.

I. INTRODUCTION

XILINX’S Spartan 6 Field Programmable Gate Arrays (FPGA) are reprogrammable integrated blocks consisting of four major FPGA architecture consists of three category of configurable elements blocks and memory block namely the configurable elements are input/output blocks (IOB), core array of configurable logic blocks (CLB), resources for interconnection. The FPGA consists of program controlled data multiplexers, numerous combinational logic array and flip-flops. The CLB also consist of random access memory (RAM) cells to achieve logic function of several variables and stored in the truth table form, the elementary logic gates required for implementation to realize the functions is not limited to the user could optimize the interconnection (VLSI). However, there are minimal trade-offs between ASIC and FPGA project is that the ASIC is extremely faster in terms of processing speed in comparison with FPGA and very ideal for projects where speed optimization is essential; the user could optimize the operating speed as they required. ASIC device requires enormous resources and cost implication needed for ASIC design tool compared to FPGA, especially at the prototyping stage. However, the FPGA requires more power consumption in complex System-On-Chip implementation in comparison with the ASIC device.

II. MAJOR PROGRAMMABLE DEVICES COMPARISON

A. Application Specific Integrated Circuit (ASIC)

The device is not a reprogrammable device like CPLD and FPGA; consequently, the programmer requires sufficient time to design for ASIC Very large scale integration (VLSI) system on chip code for implementation. The trade-off between ASIC and FPGA project is that the ASIC is extremely faster in terms of processing speed in comparison with FPGA and very ideal for projects where speed optimization is essential; the user could optimize the operating speed as they required. ASIC device requires enormous resources and cost implication needed for ASIC design tool compared to FPGA, especially at the prototyping stage. However, the FPGA requires more power consumption in complex System-On-Chip implementation in comparison with the ASIC device.

B. Complex Programmable Logic Device (CPLD)

Complex Programmable Logic Device is adequate for small digital logic projects as it is a smaller FPGA edition with simplified architecture. However, there are minimal difference in terms of memory types used in CPLD and FPGA; the CPLD memory is based on electrically erasable programmable read only memory (EEPROM) whose contents can be erased and reprogrammed using a pulsed voltage and FPGA is based on read only memory (RAM) loses its memory with power loss.

Submitted on September 12, 2020.
Published on June 25, 2021.
Kennedy Ahamefula Iroanusi.
(e-mail: mr.iroanusi@gmail.com)

DOI: http://dx.doi.org/10.24018/ejers.2021.6.4.ID

Vol 6 | Issue 4 | June 2021
C. Digital Signal Processors (DSP)

This type of programmable devices is solely for signal processing based on DSP algorithms implementation. It does not function as a microcontroller or field programmable Gate array to control a signal. However, the DSP chips has a lot of DSP blocks capable of handling very complicate signal and image processing capacity better than FPGA (Bishop 2009).

D. Field Programmable Gate Array (FPGA)

The field programmable gate array is highly used in prototyping and verification processes and has the capacity to replace DSP and microcontroller in medium to highly complex project. It is not as cheap as MCU, but also complements the function of both DSP and MCU.

E. Microcontroller (MCU)

The microcontroller is the most used programmable device in embedded systems; it is a highly optimized integrated circuit chip for general-purpose controller with wide range of embedded applications. However, the microcontroller does not have the capacity for some complex digital signal processing as DSP chips and FPGA but could handle simple signal processing. It is the cheapest amongst programmable devices equipped with excellent functions with minimal power consumption. The processing power and speed is not high and fast in comparison with FPGA.

F. Xilinx Spartan 6 FPGA

The choice of medium range FPGA product is the Xilinx Spartan 3E and Spartan 6. The Spartan 3E is the most popular board, but also an older version of the Spartan family and Spartan 6 is the newest type. The Spartan 6 has better features, functionality and overall performance compared to Spartan 3E; in terms of the functionality Spartan 6 consist an on-board Digital Signal processing chip and memory controller block, whilst Spartan 3E has none of the aforementioned features. The Spartan 6 is equipped with several values such as programmable system integration with features such as increased pin-count to logic ratio for I/O connectivity, more than forty I/O standards for simplified system design and PCI Express® with integrated endpoint block. The FPGA has increased system performance in terms of up to 8 low power 3.2 GB/s serial transceivers and 800Mhz DDR3 with integrated memory controller; consume less power with enormous data transfer capability. The FPGA board value also includes the bill of material (BOM); cost reduction was applied to the predecessor board to reduce board components and optimise the board. The are several add-on functionalities on the new board, which includes on-board digital logic components, processors (e.g., SOC, central processing units (CPU’s), microprocessor units (MPU’s), DSP’s), mixed-signal and analogue components (e.g., amplifiers, analogue-to-digital converters (ADC’s). Others includes digital-to-analogue converters (DACs), on board sensors (e.g., temperature, pressure, humidity), power supplies, thermal management components, volatile and non-volatile memory components, passive components (e.g., resistors, capacitors, inductors), safety, security, and reliability components, protocol physical layer (PHY) components and clocking components. Therefore, BOM reduction results to an overall cost-optimisation for system I/O expansion and an integrated MicroBlaze™ processor consisting of a soft IP to eliminate external processor or Microcontroller unit (MCU) components. The aforementioned design improvements have led to total power reduction to achieve a 1.2V core voltage or 1.0V core voltage option, a zero power consumption at ideal state achieved with the utilisation of hibernate power-down mode. Finally, we would state the design accomplishments has also led to accelerated design productivity in the area of fast design closure using integrated wizards and the use of ISE® Design Suite at no-cost, front-to-back and FPGA design solution for both Linux and Windows environment.

G. Electronic Data Acquisition and Communication Protocol

The key requirement for the use of data Acquisition Cards (DAQ) are because the Personal Computer (PC) cards comprises of both with analogue and digital I/O connections. Therefore, the use of Graphic User Interface (GUI) or Human Machine Interface (HMI) possible; the use of LabVIEW or programming language such as C#, JAVA or MATHLAB to create the GUI easier. The use of LabVIEW is easier as it is a graphical design approach and requires little or no knowledge of programming on the aforementioned programming languages for the GUI.

The required communication protocols/links typically used for sensors interfacing is the asynchronous serial communication protocol, which comprises of a universal asynchronous receive and transmit (UART). The communication link will consist of one receiver line and one transmitting line, whereby the nodes must match baud rate and communication protocol. It is evident that the RS232 Serial Port on PC’s utilizes UART format at a bipolar voltage signal of ±12 V; therefore, the need of UART to RS232 converter chip such as MAX232 is required.

The use of synchronous serial communication option is readily available based on the serial peripheral interface (SPI), which requires one clock signal, one bidirectional data and one chip select/enable signal. The I2C communication protocol is an Integrated Circuit bus originally developed by Philips Electronics for communication on Television print circuit board (PCB) and has been utilized for numerous commercial sensor systems. It is imperative to highlight the use of IEEE P1451; which is a sensor communication standard as there are so many different sensor communication protocols for different applications.

The asynchronous communication standard does not need the clock to control the output based on effects of changes to input, while the synchronous communication standards require the use of clock signal to control the events occurring on the output signal.

H. Accuracy, Resolution, Precision and Sensitivity

The determination of the accuracy, precision, resolution and sensitivity of a device is ideal in Electrical and Electronics prototyping and manufacturing. The accuracy is the quantity of uncertainties because of gain errors and associated offset parameters present in a measurement when compared with a universal standard. Offset errors could be stated in terms of the measurement unit such as volts, amperes, or degree centigrade and could be considered to be independent of the range or magnitude signal being measured, but gain errors is very dependent on the magnitude
of the measuring signal and could written in terms of percentage. However, total accuracy is the combination of both aforementioned independent and dependent method. If 5 V is expected, but the voltage ranges of 4.995 V to 5.005 V is measured; therefore, the accuracy determined to be ±5 mV (The difference between measured value and expected value) or 0.1 percent of error of expected voltage (((Measured – Expected) × 100%)/Expected).

Precision based on the reproducibility, repeatability and/or refinement of measurement, calculation, or specification, which is the degree of agreement of tests results or measurement(s), especially as represented by the number of digits given. A high degree of precision or repeatability is essential in testing medical equipment. Resolution is the ratio between the maximum signal measured to the smallest part that can be resolved with the aid of analogue-to-digital converter or expressed as the degree of change that theoretically calculated or detected, typically expressed in term number of bits. This relates the number of bits of resolution to the actual voltage measurements. The resolution of a system using an 8-bit ADC; a measurement system required to measure across a 5 V range using an 8-bits AD converter. Therefore, the smallest possible increment we can detect at 8 bits, 2^8 = 256, or 1 part in 256, so 5 V=256 = 19.53125 mVls per AD count. Therefore, the smallest theoretical change we can detect is 19.53125 mV. It is important to take into consideration the other factors such as noise, which may reduce the overall resolution into the equation to minimize the theoretical number of required bits. Therefore, a data acquisition system with an 8-bit resolution might also comprise eight counts of noise. Considering this noise, the eight counts equal 3 bits (2^3 = 8); therefore, the 8 bits of resolution specified for three bits reduce the measurement system, so the ADC actually resolves only 5 bits, not 8 bits. However, the averaging technique improves the resolution slightly by minimizing the noise by the square root of the number of samples, but it sacrifices speed, if speed is not required as a requirement for the design. The averaging technique procedure requires multiple readings to be added together and then divided by the total number of samples; therefore, the eight counts of noise averaging 64 samples would reduce the noise contribution to one count, √64 = 8; 8+8 = one. However, most of the associated noise will be a Gaussian distribution type and the technique does not have the capacity to minimize the effects of non-linearity. Sensitivity is an absolute quantity, the minimal absolute amount of change that observed by a measurement. The system input voltage is 5V with three counts of noise, if the A/D converter resolution is 2^n the peak sensitivity will be three counts multiplied by the ratio of two and two hundred and fifty-six (2^n256) equals to 23.4375 mVpk; it will determine how the sensor responds. A sensor rated as 1000 units with an output voltage of 0-5 volts (V); the equivalent measurement will be 5mV equals to one unit; therefore, with a sensitivity of 23.4375 mVpk, it will take four units before the input detects a change. The ADC0809 chip is a complete Data Acquisition System (DAS) for both absolute standards and ratio-metric conversion systems. The input voltage range of ADC0809 is equal to the supply range; therefore, the transducers can be connected directly across the supply and the outputs connected directly into the multiplexer inputs.

The sensitivity of the system at 25C can be determined as the product of the resolution and the half least significant bits root mean square value (±5/2 LSB RMS). The total error of the system with be the summation of the gain error, the offset error and the linearity error. Whereby, the degree to which the voltage output from the sensors selected varies in direct proportion to the primary measuring parameter through its full-scale amplitude; nonlinearity in the output can be attributed to the saturation of the flux concentrator approaching the full-scale value. In a situation, where there are no signal conditioning circuit and/or filter capacitors utilized at the analogue inputs and the signal source impedances is negligible. The comparator input current will not introduce converter errors; even though the comparator input current varies directly with clock frequency and with input voltage, as the transient produced by the capacitance discharge will degenerate before the comparator output is strobed and any other effect may be quantified conventionally.

III. HARDWARE DEVELOPMENT

The hardware design and development steps, which includes the development of the FPGA firmware, hardware schematic design, PCB design and LABVIEW GUI development. Power consumption and heat management are important in any engineering or Radio Frequency that operates at very high frequencies; therefore, several semiconductor manufacturers optimize their design using various technique. Therefore, managing and monitoring heat dissipation and power consumption is important using parameters such as current, voltage and temperature.

A. Functional Requirement

Functional requirement is tan essential requirement that must be accomplished by developers for the developer such as the measurement function, data collection and monitoring function and the respective parameters for the key performance indicators/performer:

![Diagram](Image)

Fig. 1. Functional requirement.

B. Non-Functional Requirement

Non–functional requirement is the requirement, which mostly related to the user or other potential users. They are also important for the project. Developer needs to pay attention to them rather than only focus on functional.
They concretely include these factors:
1. FPGA Verilog codes with comments for future modification.
2. Hardware schematic design.
3. Double sided printed circuit board (PCB) design with component positions and routing.
4. PCB board size.

5. Graphic User Interface (GUI) or Human Machine Interface (HMI).

IV. UNIFIED MODELLING LANGUAGE (UML)

A. GUI User – Case Diagram

The GUI creates a provision for monitoring of data obtained from RS-232 link and for further data processing; the use-case description is below:

<table>
<thead>
<tr>
<th>Stage</th>
<th>User</th>
<th>GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COM number is selected as a communication parameter setup process</td>
<td>Setup parameter is saved</td>
</tr>
<tr>
<td>2</td>
<td>The button is clicked to open COM</td>
<td>Facilitate communication between GUI and serial communication (RS-232 link)</td>
</tr>
<tr>
<td>3</td>
<td>GUI channel selection enabled with the channel select switch.</td>
<td>In accordance with the select channel by the user; display processed data</td>
</tr>
<tr>
<td>4.</td>
<td>Either export data Excel file or print out data figure</td>
<td>Generate and populate the excel file with data or provide snap of the data displayed</td>
</tr>
</tbody>
</table>

B. Console Diagram

The GUI console diagram consist of 5 boxes situated at the top of GUI console represent seven segment display module which for numeric data rather than the Liquid Crystal Display (LCD) typically used in modern digital Multimeter. However, for efficient monitoring process, it is essential to add graphical display to the GUI and button has been added to increase user experience and GUI functionality. The GUI console diagram, the hardware use-case diagram to illustrate access, hardware sequence diagram, sequence channel manipulation to demonstrate operational sequence of the hardware and to channel change, respectively.
C. Hardware User-case Diagram

Fig. 4. Hardware User-case diagram.

D. Generic Hardware Sequence Diagram

Fig. 5. Sequence diagram.

E. Hardware channel modification sequence

The generic use-case descriptors are shown below.

Fig. 6. Sequence diagram for channel selection.
TABLE II: HARDWARE USE-CASE EXPLANATION

<table>
<thead>
<tr>
<th>Stages</th>
<th>User</th>
<th>Hardware system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compulsory</td>
<td>Switch on</td>
<td>Hardware operation commencement</td>
</tr>
<tr>
<td>Stage 1</td>
<td>power</td>
<td>FPGA commences the command signal for ADC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ADC read the sensor current output based on the complementary chosen channel.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The FPGA commences ADC control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The FPGA obtains sensor data from ADC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The FPGA converts raw data into RS-232 packaged data format and transmit through RS-232 interface to PC</td>
</tr>
<tr>
<td>Optional</td>
<td>Select a different channel on</td>
<td>The chosen channel’s data is stored within the hardware system</td>
</tr>
<tr>
<td>stage 2</td>
<td>PCB and/or GUI</td>
<td></td>
</tr>
</tbody>
</table>

Stage 1 reset option
Switch on the reset switch
The option causes the system to activate the reset statue; thereby aborting data transmission.

V. HARDWARE SCHEMATIC DESIGN AND PROTOTYPING

The following resources are required for design and development of the hardware system are shown below:

TABLE III: DESIGN AND DEVELOPMENT RESOURCE REQUIREMENTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spartan 6</td>
<td>Utilise for the implementation of the System on Chip</td>
</tr>
<tr>
<td>Xilinx ISE Design</td>
<td>Code development and uploading to FPGA</td>
</tr>
<tr>
<td>Suite 14.3</td>
<td></td>
</tr>
<tr>
<td>Altium Designer 10</td>
<td>Utilise for schematic and PCB design</td>
</tr>
<tr>
<td>Modelsim SE-64 10.1c</td>
<td>Utilise for code simulating; the FPGA code written in Verilog</td>
</tr>
<tr>
<td>Docklight V2.0</td>
<td>Use for effortless monitoring of packaged RS-232 packages transmitted from the FPGA</td>
</tr>
<tr>
<td>NI LabVIEW</td>
<td>Utilise for GUI development</td>
</tr>
</tbody>
</table>

A. Schematic

The Spartan 6 FPGA and the ADC0809 integrated chip communicates with one another through the level shifter; the 74LVC245 and 74HCT245 level shifters convert the 5V logic level to 3.3V logic level and 3.3V logic level to 5V logic level respectively. The switches are added to implement manual channel selection rather than automated channel selection approach; thereby creating a provision to change FPGA code control to user control. It is essential to understand the current and voltage logic levels of the FPGA I/O port; a 3.3V level signal may not be ideally fed into the FPGA I/O pins directly because of the current level. The FPGA obtains data from switch and 74LVC245, before the FPGA sends packaged data the PC for GUI display.

B. Experimental Logic level transfer

Fig. 7. Hardware PCB design.

Fig. 8. 74LVC245 chip waveform level conversion.

Fig. 9. 74HCT245 level shifter waveform level conversion.
VI. SOFTWARE DEVELOPMENT

A. Xilinx ISE Verilog Code

The Xilinx ISE allows the FPGA to be programmed using either VHDL or Verilog code, the use schematic design approach is widely sorted as it can be associated to the popular bottom-up design method; whereby the entire system is subdivided into numerous small blocks and the blocks are linked-up together to form the complete system. This approach does not require a good knowledge of coding but linking blocks with a wire-like symbol and developers could see or envisage the system’s functionality in a graphical mode without studying each line of the code. However, the schematic is editable, and the schematic symbols are required to be created for the codes.

It is important to complete the schematic symbol creation processes for every Verilog files before Xilinx schematic design commences. The first process is to add all of the required symbols into the Xilinx ISE design window, the second process it to edit all symbols not adequate for wiring in their default state and finally to link and assign input and output (I/O) appropriate for the symbols.

The schematic design below comprises of 5 blocks, which includes switch, analogue to digital (AD), transmitting block (TX), frequency dividers for 50HZ (clk_50) and 640 KHz (clk_640K) clock signals implemented utilizing Verilog Hardware description language on Xilinx.

The Clk_50 block is described with the aid of an activity diagram created with unified modelling language (UML) to highlight the implementation procedure for the aforementioned block. The function block and Verilog code are to generate a 50Hz clock signal using frequency division approach, by dividing much faster frequency into slower frequency; the 50Hz clock frequency obtained from this method is used to control the analogue-to-digital Verilog code/block. A very fast clock frequency is not ideal for the analogue-to-digital Verilog code as the output data will become extremely too fast to capture. Therefore, a First-In-First-Out (FIFO) storage or a buffer circuit is essentially required, but since the clock signal provides up to 20ms of conversion period and cannot be consider being too slow.

![Fig. 10. Xilinx Schematic design.](image1)

![Fig. 11. Clk_50Hz clock frequency activity diagram.](image2)

![Fig. 12. Simulated 50Hz clock frequency.](image3)

ModelSim simulation result of the 50Hz frequency divider (clk_50) waveform shown below:

The 50 Hz clock frequency (clk_50) signal waveform result generated from an input signal of the 50MHz on the FPGA result shown on the oscilloscope below:
The 640 KHz (Clk_640K) clock frequency generate is to enable the 8-bit Analogue-to-digital converter hardware chip (ADC0809) to be driven as recommended on the datasheet. The implementation of the 640 KHz clock frequency (Fc) follows a similar methodology applied to the generation of 50 Hz from 50 MHz. The function block/Verilog code are to generate a 640 KHz clock signal, which translates to a period of 1.56 μs using frequency division approach, by dividing much faster frequency into slower frequency; the 640KHz clock frequency obtained from this method will be utilized to drive changes to the events on the ADC0809 Hardware analogue-to-digital converter. The description for the implementation of the 640 KHz (clk_640K) clock signal is accomplished with the aid of an activity diagram below:

ModelSim simulation result of 640 KHz clock signal (clk_640K) generated from 50MHz clock signal through frequency division mode is shown below:

The ideal clock frequency required was 640KHZ, but the realistic frequency achieved is 641.028 KHz frequency waveform generated from the 50 MHz clock frequency signal by the FPGA is shown below:
B. Switch

The switch part of the Verilog code is to reduce or eliminate the effect of the noise; the switch is used in combination with 40ms delay for signal detection, it provides a good security benefits, large files could be sent as a collective small packets and minimizes data corruption. The activity diagram shows the functionality of the switching Verilog code.

Fig. 18. Switch activity diagram.

The switching simulation result on ModelSim can be seen below:

Fig. 19. Switching simulation.

Noise occurred from 4.2ms to 6.7ms for a duration of 2.5ms; the design has clearly shown that the effects have not been seen on the value of key out as the high level of key_in is sustained for 40ms from 6.7ms until 46.70002ms to enable the status of the key_out value to be changed.

C. Analogue to digital Block (AD)

The function of the block is to control the analogue to digital convertor with utilization of the timing diagram from the ADC0809 datasheet and the Verilog case statement code is written for the implementation of an AD control logic, which has been explained with the aid of the activity diagram.

The successive approximation register (SAR) of the AD converter executes eight iterations to achieve a near realistic input voltage value, resetting could be achieved on the positive edge of the start conversion pulse (SC or START), which is followed by the commencement of the conversion on the falling edge of the aforementioned start conversion pulse. However, the conversion in process could be interrupted by the acceptance of another new start conversion pulse. A continuous conversion could be achieved with the
aid of the end-of-conversion (EOC) output to the SC input; when it is utilized, an external start conversion pulse is required after power up as the EOC will be low between zero and eighth clock pulses after the rising edge of start conversion. An essential feature of the AD converter is the comparator, which is It is ensures that the ultimate accuracy of the entire converter is achieved with respect to the comparator drift with the utmost effect on the repeatability of the device. The Analogue to digital converter tremendously insensitive to wide drift input offset errors and changes to temperature because of the chopper-stabilized comparator feature. The chopper-stabilized comparator imparts the most credible approach of satisfying all the converter requirements; converting direct current (DC) input signal into an alternating current (AC) signal and fed via a high gain AC amplifier and then restores the DC level. The aforementioned technique limits the DC drift component of the amplifier as the AC amplifier blocks it.

The waveforms below are acquired from the FGPA and the ADC0809 integrated chip namely the Output Enable (OE), start conversion, address latch enable (ALE) and the End-of-conversion from ADC (EOC) waveforms.

![Diagram](image.png)

**Fig. 20. AD control logic activity diagram.**

**D. Analogue to Digital (AD) tested with Oscilloscope**

![Waveform](image.png)

**Fig. 21. Output Enable (OE).**

![Waveform](image.png)

**Fig. 22. Start Conversion (SC).**
VII. DATA TRANSMISSION (TX)
The function of block is to package the data for transmission utilizing the RS-232 format and transmitting via the RS-232 interface. The activity diagram below illustrates detailed description of the Verilog code created for the TX function of sending data through the serials port.

---

**Fig. 25. Serial Port Data TX activity diagram.**
A. ModelSim Simulation of Data TX

![Fig. 26. Data TX Simulation result.](image)

On the above simulation result, the order of the digital data on the Din is 01110101 and the data achieved on the TX is 10101110, which is exactly a reversed data and ready for transmission via the serial port.

B. Seven Segment Display Module

The graphic user interface (GUI) development has been accomplished utilizing LabVIEW without enormous programming using any of the aforementioned programming languages or in-depth knowledge of any programming language as it is a graphical approach using several tools provided by LabVIEW. The GUI will consist of the functionalities of an oscilloscope, Multimeter and display the equivalent digital bits on the same GUI for better understanding of the data. The segment (SMG) display is utilized by Multimeter for reading display and would be incorporated into the GUI; however, the data required to be displayed will consist of integers and decimal numbers, it is imperative to utilize the dot function for each integer digit controlled by an enable pin. The LabVIEW design of the segment display can be seen below.

![Fig. 27. Segment Display Front panel (SMG) and block diagram.](image)

The SMG has inputs for the activation of individual Light Emitting Diode (LED) making comprised within the segment display from A to H inputs; therefore, a data structure is required in the form of cluster data package. Therefore, a segment display group is achieved by using encapsulated several single inputs and single terminal as enable control as shown below.

![Fig. 28. Enveloped SMG cluster of 8 Input elements.](image)

The SMG encoder data from the above table will be implemented on LabVIEW for segment display as shown below.

![Fig. 29. The SMG encoder data table.](image)

![Fig. 30. Single segment display test.](image)

C. Single Segment Display Front panel (SMG) and Block Diagram with Encoder

The default display value of zero on the SMG is obtainable, when the input is 0, button for dot (Dp) and enable (EN) is
activated will result to 0. The LabVIEW program will execute the following task in chronological order below:

- For Zero (0) number input entries, the multiplexer (select switch structure) and the Dp status must be True.
- The LabVIEW will know the digital data corresponding to zero and a dot for SMG display.
- LabVIEW locates the input value for “0”, which is 252 on the encoder table and the value of “1” is added to represent the dot resulting to a total value of 253.
- The output of 253 is provided to the next block (Multiplexer structure) to be converted into correct data structure for display purposes.

The process for the incoming input number(s) are as follows below:

The LabVIEW library ensures the conversion of input numbers into bit array from number to Boolean array; in a situation the input decimal number is 253 for the display of “0.” on the SMG, the corresponding output after Number to Boolean array will be \([T, T, T, T, T, T, T, T]\), represent from least significant bit (LSB) to the most significant bit (MSB). Whereby, T represent true or equivalent of logic “1” and F represent false or logic “0”.

The utilisation of Index array allows all individual bits of the Boolean array data to be recombined subsequently after the use of Bundle.

The applied input:

The desired output: \([T, T, T, T, T, T, T, T]\)

The modus operandi for manipulating the order of the bits respectively shown on the table below:

<table>
<thead>
<tr>
<th>SMG</th>
<th>Bit1</th>
<th>Bit2</th>
<th>Bit3</th>
<th>Bit4</th>
<th>Bit5</th>
<th>Bit6</th>
<th>Bit7</th>
<th>Bit8</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit6</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit5</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit4</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit3</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit2</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit1</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The output bits \([T, T, T, T, T, T, T, T]\) are utilized for the purpose of controlling the segment display as shown below on the table:

<table>
<thead>
<tr>
<th>SMG</th>
<th>Bit1</th>
<th>Bit2</th>
<th>Bit3</th>
<th>Bit4</th>
<th>Bit5</th>
<th>Bit6</th>
<th>Bit7</th>
<th>Bit8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit6</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit5</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit4</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit3</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit2</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Bit1</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The display of “0.” on the SMG, it is appropriate to control the dot through an autonomous control without expanding the magnitude of the encoder table. There is no need of writing case statement for the dot(s) with the use of multiplexer switching select structures; only add the value of 1 into initial value to cause the dot to light up as a control mechanism. It is essential to cascade multiple single segment display block into a collective segment display module, but it is also important to have a comprehensive understanding on the procedure for the separation of input number values into single bit for displaying to the appropriate segment display block of the aforementioned collective segment display module. Therefore, the LabVIEW program has been designed below to handle the separation of bits into the appropriate individual SMG.

The design precision for the monitoring part of the system is to be able to measure up to 0.001 of temperature, current and voltage. Therefore, in a situation where the measured input value to be displayed on the cascaded SGM is 18.520; how would we display on individual segment display of the cascaded SGM, we need to perform some calculation
involving simple data manipulation and 10% modulus below:

Tenths: $18.520 \times 10^{-1} \% 10 = 1$, Units: $18.520 \% 10 = 8$,
One-tenth value: $18.520 \times 10^1 \% 10 = 5$ , One-hundredth value: $18.520 \times 10^2 \% 10 = 2$

One-thousandth value: $18.520 \times 10^3 \% 10 = 0$

Therefore, the expected array conversion from the input number of 18.520 is an output in a one by five array of [1,8,5,2,0] an suitable for a cascaded segment display module comprising of 5 individual SMG.

**VIII. LABVIEW SERIALS PORT DESIGN**

The GUI needs a LabVIEW design for a serial port to receiving incoming signals for monitoring purposes. The design will require five basic steps to achieve the aforementioned aim of the design below.
The GUI requires the following steps below to achieve the set out objectives, the measured data from sensors monitoring voltage, current and temperature are monitored on the GUI via the serials port. Therefore, the steps below show the processes involved in acquiring data via the serial port.

A. Stage 1 (Set)

The setup parameters are shown on the table below:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>38400</td>
</tr>
<tr>
<td>Number of bits</td>
<td>8bits</td>
</tr>
<tr>
<td>Stop bit</td>
<td>1bit</td>
</tr>
<tr>
<td>Parity check</td>
<td>None</td>
</tr>
<tr>
<td>Maximum timeout</td>
<td>10 seconds</td>
</tr>
<tr>
<td>Flow control</td>
<td>None</td>
</tr>
<tr>
<td>Termination character</td>
<td>Enable</td>
</tr>
<tr>
<td>COM number</td>
<td>User select</td>
</tr>
</tbody>
</table>

B. Stage 2 (Read)

The delay is added to enable us to have good control of the data being sent and/or received for the purpose of monitoring the data on the GUI; a delay time of 10ms has been utilized. If the delay is too long, the enormous data the port and make the system not responding. However, if it is too short, the user cannot monitor the data disrupt the port by jamming and if the timing is too small; then system becomes too quick and monitoring will be of an issue. Therefore, we could state that the delay time is a trade-off between user experiences based on system efficiency or system stability, every time a byte of information is received from the system buffer.

C. Stage 3 (Process)

This process stage involves three sub-processes as expressed below.

The RS-232 data format is an ASCII values; therefore, data conversion into hexadecimal is essential for data process and display; the LabVIEW design for the required conversions from ASCII to integer format is presented below.
The test on ASCII format to integer format; the buffer 33 was read and value was converted appropriately below.

![Diagram](image1)

**Fig. 42. LabVIEW ASCII format-to-integer format block design.**

It is evident that the substring bit value has been manipulated in a reverse order, the data alteration was achieved within the array; whereby the most significant bit become the least significant bit for the Numeric number bits equivalent as shown on the figure below.

![Diagram](image2)

**Fig. 43. ASCII format to integer format conversion test result.**

It is imperative that the correct measured parameter (current and/or voltage) values are calculated appropriate, therefore, further data process is required, and the LabVIEW design can be seen below.

![Diagram](image3)

**Fig. 44. Bit manipulation.**

Conversion from voltage quantization integer value into decimal number and secondly to display voltage quantization integer value in 8-bit representation. In a situation whereby, the voltage quantization integer value is 204; therefore the create a provision for the conversion from integer into decimal number and bit seen on the GUI.

This virtual LED bar is connected at the output of the Analogue to Digital converter for the purpose of displaying the present byte information (8 bits) for the user via the GUI and essential for system debugging. The LabVIEW design for the virtual LED bar and GUI data can be seen below.

**D. Virtual LED bar test**

![Diagram](image4)

**Fig. 45. Further Parameter Processing LabVIEW design.**

![Diagram](image5)

**Fig. 46. Eight bits LED bar display Front panel and block schematic.**
analogue to digital converter is precision \( \frac{1}{2^n} \) and as the reference voltage value is 5V; the decimal number of quantization value could be determined as the product of precision, reference value and the input integer value as can be seen below: \( \frac{1}{256} \times 5 \times 204 = 3.984V \)

The virtual LED bar front panel will display 2047 in the format below in line with the analogue and digital converter is D0 representing the most significant bit and D7 representing the least significant bit as connected on the output of the AD converter:

![LED BAR 2](image)

Fig. 47. Digital value representation on the virtual LED bar front panel.

The next point of focus is to utilize the decimal number to deduce the correct display value with relevance to the channel select through a calculation process as other channels has a divergent calculation approach for its display value. The required channel calculation for the voltage value; it involves three processes

![Process block for Y-axis labels and data plot name](image)

**Process 2** – The process involves altering the parameter’s label and Y-axis’s label in accordance with the channel

![Voltage = IN](image)

**Process 3** – The LabVIEW design utilizes the corresponding algorithm to deduce the correct display value for any of the three aforementioned parameters. In a situation whereby the user has selected the voltage as a parameter to display, the decimal number of quantization voltage value is 3.984; therefore, the first process will output “Voltage” as the channel name and the unit as “V”. The second process involves the changes by LabVIEW in regards to parameter title to “Voltage” and the Y-axis is labelled as “V” on the wave window. The third process commences by using the
appropriate theoretical calculation approach in accordance to the selected input parameter and outputs data for expected values; the decimal display value and the display is achieved with the use of cascaded segment display module.

The data displayed on both cascaded segment display module and the wave window; the user also has the option of saving the displayed data using two different options, either to save as Excel” and/or “Print waveform” utilizing buttons adjacent to the wave window.

E. Serial Port Closure

The serial port is deactivated at the end of all process and stages, it is also important to have an error reporting block mechanism at the time the serial port is active and the LabVIEW block symbol for closing serial port and the error reporting block can be seen below.

F. Others

The LabVIEW blocks utilized in previous pages and their function are briefly explained in the table below.

<table>
<thead>
<tr>
<th>Block names</th>
<th>GUI front panel</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wave toolbar window</td>
<td></td>
<td>The block diagram has the capacity for zooming in and out and also panning the figure around</td>
</tr>
<tr>
<td>Combo box (COM select)</td>
<td>COM SELECT</td>
<td>The user chooses a COM number, it allows the combo box to be used to identify the selected com device on the PC.</td>
</tr>
<tr>
<td>Voltage reference (Vref) setting box</td>
<td>Vref 5</td>
<td>The reference voltage creates a provision for the calculation required for analogue to digital conversion.</td>
</tr>
<tr>
<td>Read indicator buffer</td>
<td>read buffer</td>
<td>The buffer indicator allows the buffer to be data values to be read from serials port.</td>
</tr>
<tr>
<td>Button (Open COM)</td>
<td>Open COM</td>
<td>The COM button enables the opening or closure of COM.</td>
</tr>
<tr>
<td>Button (EN Monitor)</td>
<td>EN Monitor</td>
<td>The EN monitor button is utilised to open or close the segment display.</td>
</tr>
</tbody>
</table>

The printing of temperature data and storing of the temperature data on an Excel file was also achieved using the button for waveform printing option and the save as button option on the GUI.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save As Excel</td>
<td>The button to save data in Excel; if it has be clicked by the user will result to the output data being stored and downloaded as an excel file</td>
</tr>
<tr>
<td>Print Waveform</td>
<td>The user has the capacity to take a screenshot of present waveform plotted on the wave window and output the screen of the data to the working folder.</td>
</tr>
</tbody>
</table>

IX. EXPERIMENTAL TEST

It is imperative to conduct several subunit development test, and a complete system test; to minimize sources of error in the complete system. The subunit test carried out includes the logic level transfer, switch, CLK_50 frequency division, CLK_640 frequency division, Switch Verilog code using the test-bench, analogue to digital (AD), Transmission (TX), cascaded segment display module. The LabVIEW software by National Instruments enable us to achieve an ASCII to integer conversion, other functionalities are utilized were the LabVIEW virtual LED bar, all channels (voltage, temperature and current) of GUI tested with the aid of Docklight and the post development test can be seen below.
The post development test of the complete system was carried out; 5V was measured and expected to be displayed on the GUI via the voltage channel, but 4.98V is displayed on the GUI, which signifies an acceptable accuracy.

X. CONCLUSION

The complete system was developed using sensors, level voltage shifters, analogue to digital converter and Spartan 6 FPGA to handle a data acquisition of three parameters namely voltage, temperature and current; therefore, data were transmitted using serial communication protocol. Monitoring of the acquired data is crucial to compare the acquired data with standard measuring device is tested. It is imperative that user puts several processes in place before being view after acquisition has taken place, as the viewer cannot comprehend the binary values of data from the ADC. The binary data is required to be converted into a format fit for displaying on the GUI and easily comprehended by the user(s); we could state that the GUI developed with LabVIEW and the FPGA, as both performs a unique of binary data translation and clock functions for monitoring purposes. It is also important to highlight that the FPGA SOC was created as a Verilog code on Spartan 6 FPGA and preferable in the industry in comparison with Very High Speed Hardware Description Language (VHDL).

REFERENCES


Kennedy A. Iroanusi is currently a Lecturer at City of Wolverhampton College, teaching Electrical and Electronics Engineering related units. He studied Electrical and Electronics Engineering at University of Birmingham and carried out some research at Coventry University. He formerly worked at Coventry University and aim to complete a PhD by publication at Warwick University. Whilst at Coventry University; I lectured undergraduates on Analogue and Digital Electronics, Electrical Engineering module, Very Speed Hardware Description Language (VHDL) both at Undergraduate and postgraduate students, Digital Signal Process (DSP) and communication. He has a keen interest in Analogue and Digital Electronics, Electrical Engineering and Microprocessor, Human Visual systems, Car security products, high-speed digital processing solutions, Microwave systems, Antenna design, Radio Frequency (RF) subsystem, , Field programmable gate array (FPGA) or Complex Programmable Logic Device (CPLD) prototyping, application specific integrated circuits (ASIC).System on Chip (SOC), Verification, Mixed Signals (Digital and Analogue). He is a Fellow of Higher Education Academy (FHEA), a professional Scrum Master (PSM) and holds a Qualified Teachers Learning and Skills status (QTLS). He has vast interests on Bioengineering related research topics.