EENG6730 Mini Project Cover Sheet

Title: Assignment - Digital Systems Realisation

Module: EENG6730 Digital Systems Design

Work Setter: Dr Md Moinul Hossain (M.Hossain@kent.ac.uk)
and Dr P Assimakopoulos (P.Asimakopoulos@kent.ac.uk)

Demonstrator: Jihad Dib (J.Dib@kent.ac.uk)

Assessment: This assignment contributes 28.0% of the module.

Deadline: Submission deadline (for the **report**):
Group 2: 12:00 noon, Tuesday 14 December (KV Week 20)
Group 1: 12:00 noon, Friday 17 December (KV Week 20)
Marked work return date: Friday KV Week 25
Submit: On-Line

Intended Learning Outcomes: This assignment contributes to the assessment of the EENG6730 learning outcomes:
8.2 design digital systems which are easily testable
8.3 use a range of software tools which synthesize digital systems from VHDL

Feedback method: Feedback will be via: Written comments on work. Further feedback will be available on request.

You are reminded of the University rules on Academic Integrity as stated in the student handbook. Examples of conduct regarded as a breach of these regulations include:

- *Plagiarism: reproducing in any work submitted for assessment or review (for example, examination answers, essays, project reports, dissertations or theses) any material derived from work authored by another, without clearly acknowledging the source*
- *Duplication of Material*
- *Conspiring with others to reproduce the work of others without proper acknowledgement, including knowingly permitting work to be copied by another student*
- *Falsification of data/evidence.*

Please note that IET accreditation requires that when coursework contributes 30% or more of a module then credit cannot be obtained if either your coursework or examination scores are more than 10% below the module pass mark. This is irrespective of your module overall mark.

%	**Marking Criteria**
>70	The work is very clearly presented and structured to show the method and analytical, or design process. Assumptions are stated and values are given to appropriate accuracy, significant figures, & correct units are used. High standard engineering practice is demonstrated (e.g. use of Nearest Preferred Component values). Comparisons, judgements and critical assessments are made where appropriate. Understanding beyond the provided teaching materials may be demonstrated.
60-69	Good presentation and structure with little or no ambiguity or omitted stages. Correct values are obtained, though not necessarily to appropriate significant figures, units are provided – though not always using appropriate engineering prefix (e.g. pF compared to F), required results are obtained, but no appropriate comment is made.
50-59	Acceptable presentation and method, though some stages may lack detail or contain errors. Results may be incorrect though the method is broadly appropriate. Units may be missing or occasionally incorrect. Some sections may be missing.
40-49	A significant amount of the required work is submitted, though there may be errors and presentation not be clear in places. There may be errors in the calculation but methodology is broadly appropriate.
0-39	Only a small amount of the required work is submitted, the structure and presentation are very unclear and difficult to follow. Many errors in calculations, and little evidence of understanding.

Design and Implementation of a Digital Phase Shift and Frequency Measurement Circuit using a Xilinx FPGA

Introduction

In this mini-project, you are required to design, verify and implement a digital phase shifter and frequency measurement circuit using a Xilinx FPGA. In the past, a similar design has been used as a driver for a resonant frequency load-cell similar to the one described in Figure 1.

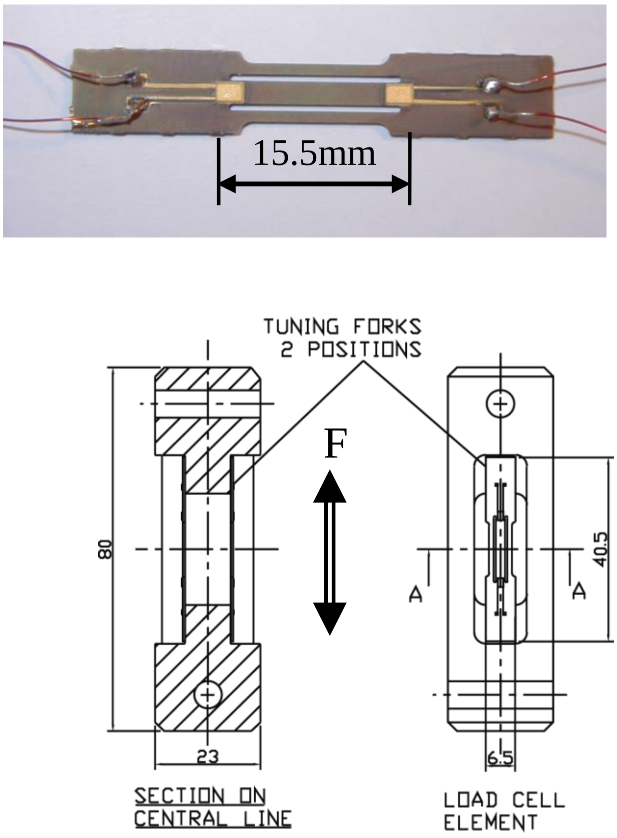

Figure 1 Thick film PZT Metallic Resonator with a cross-section of Load Cell design.

The operating principle of this system is to keep the load cell resonator oscillating at (or very close to) its resonant frequency. When a load is applied to the resonator load cell there is a change in the resonant frequency which is proportional to the load applied. By measuring the frequency, it is possible to determine the load applied to the cell.

The circuit to be designed here is used to maintain the sensor in resonance over a (wide) frequency range while also measuring the frequency of oscillation. The input (**SENSOR_IN**) to the circuit is a digital square wave (0 – 3.3V) of varying frequency output from the resonator which must be used (by the FPGA) to generate a 90^{o} phase-shifted version of the signal which is then used to drive the resonator and keep it oscillating. The FPGA also needs to use the output from the load-cell resonator to measure the oscillating frequency of the resonator.

A top-Level block diagram of the system which defines the inputs and outputs is shown in Figure 2

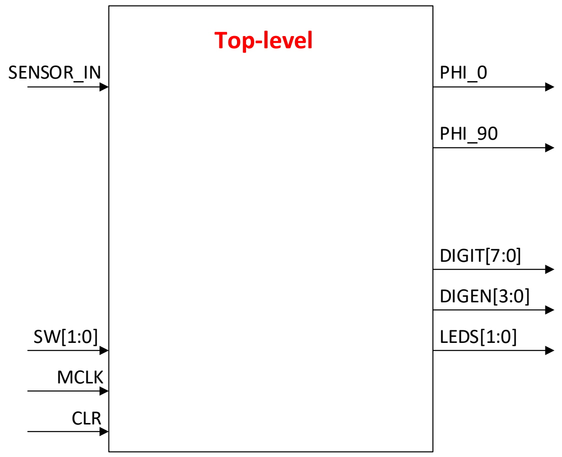

Figure 2. Top-level diagram of Digital Phase Shift and Frequency Measurement Circuit

A block diagram of all the different component circuits within the top-level used to build the overall system on the FPGA is shown in Figure 3. Further details of the internal structure of the blocks are outlined in the sections below. The two main components that need to be designed are the **Phase_Shifter** circuit and the **Frequency_Calculator** circuit. The **Display_Interface** circuit (which is provided) is also needed to drive the FPGA's LED hex display (used to indicate the measured frequency), the range control (using the slider switches on the FPGA board) and the range indicators (using the individual LEDs that are available on the FPGA Board). The VHDL description of this circuit is provided.

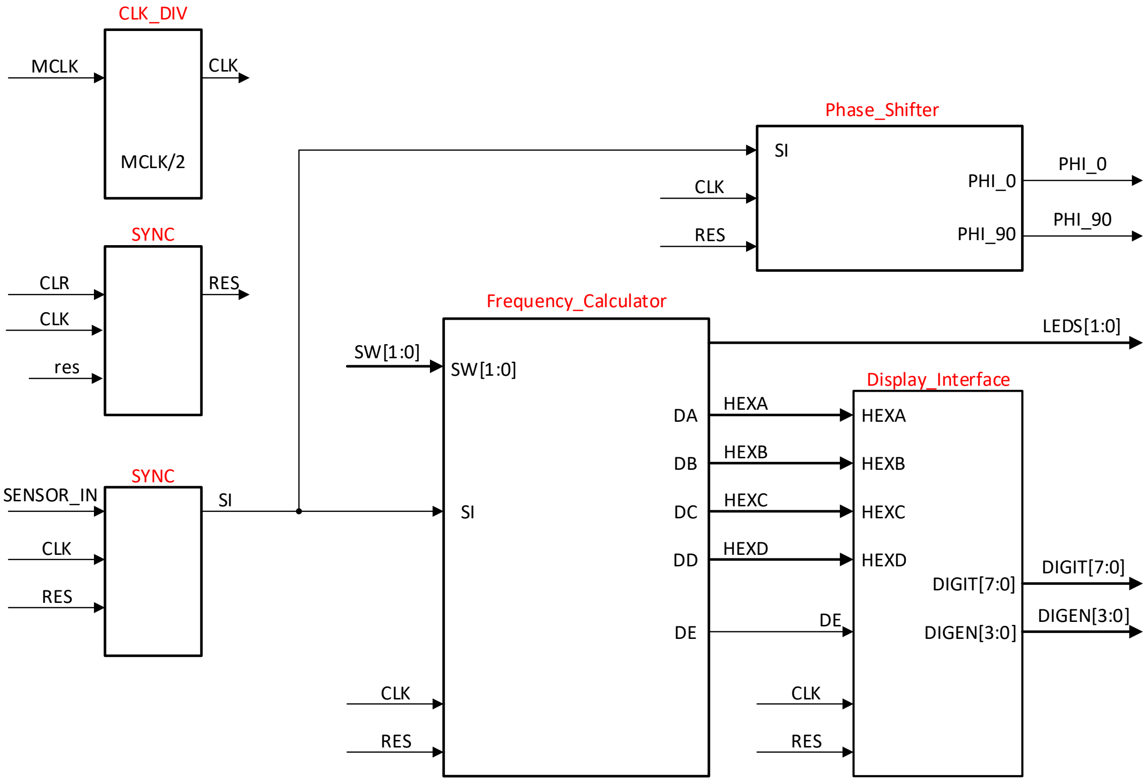

Figure 3. Block diagram of Digital Phase Shift and Frequency Measurement Circuit (inside view of top-level in Fig. 2)

For final testing in the laboratory, a signal generator will be required to input a test pulse waveform into the FPGA (a rectangular pulse waveform with an input voltage range of 0 – 3.3V) and an oscilloscope used to compare the input and output pulse waveforms to confirm that the output is a 90^o phase-shifted version of the input. The signal frequency should also be displayed on the four hexadecimal displays on the FPGA development board.

The Phase_Shifter Circuit

Figure 4 is a suggested solution for the Phase_Shifter circuit which is based on two counters. A timing diagram that indicates the operation of this circuit is shown in Figure 5. The counters count up at *half* the rate that they count down. The logic required for the two “count enable” signals (**CEA** and **CEB**) is also shown in Figure 4. Use Xilinx ISE® to design, build and simulate this circuit to confirm its operation.

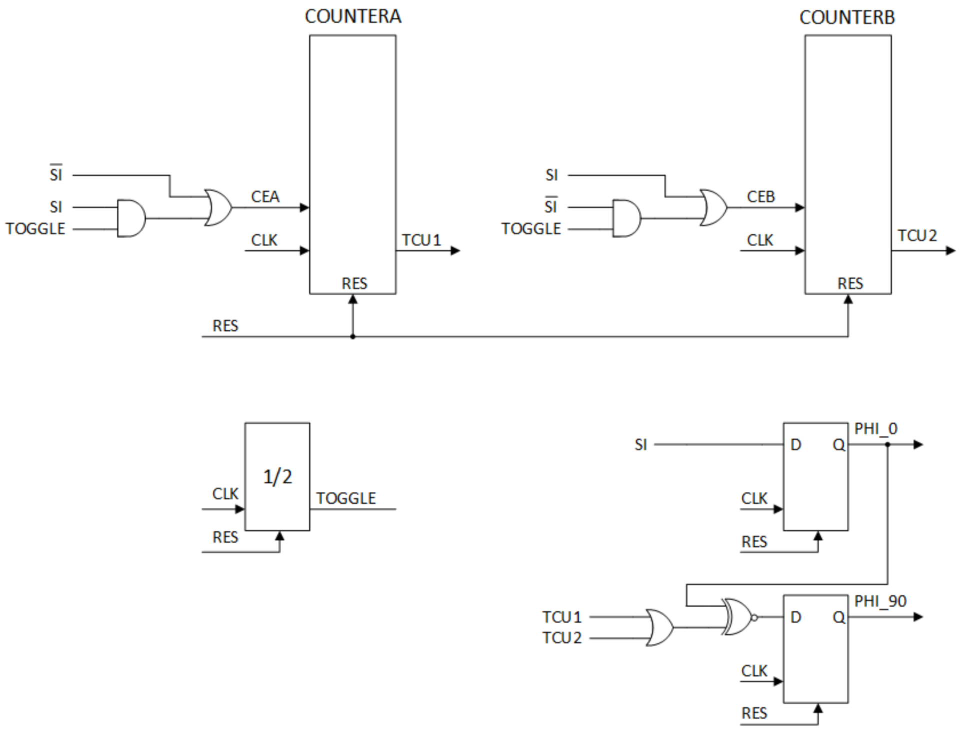

Figure 4. Suggested Phase_Shifter circuit

A timing diagram for the Phase_Shifter circuit is shown in Figure 5. The circuit is built using two up/down counters configured such that the <u>count-down rate is twice that of the count-up rate</u>. The timing diagram also includes another interim signal that is needed to produce the two output signals **PH_0** and **PH_90**.

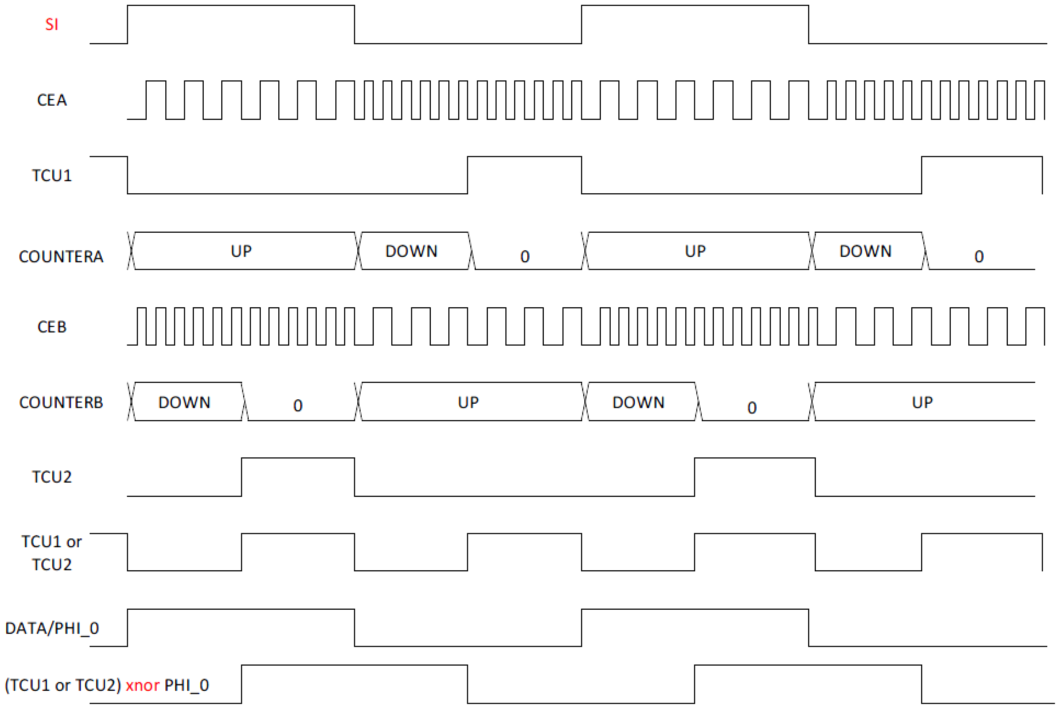

Figure 5. Timing diagram of Phase_Shifter circuit

The Frequency_Counter Circuit

Figure 6 is a suggested solution for the Frequency_Counter circuit. It is used to count the number of rising edges of the input pulse that occur in one second (using decimal or MOD10 counters) and translate them into valid hexadecimal outputs for display on the HEX LEDs available on the FPGA board.

Two slider switches are used to be able to change the range of the frequency measurement. The positions of the slider switches are to be indicated by individual LEDs on the development board.

Outputs

On the Xilinx development board, there are 4 hexadecimal display modules. The basic setting allows us to display a frequency from 0 – 9999 Hz. It is possible to extend this range or accuracy.

How would you use the display to show that the input frequency is out of range?

To extend the range, the slider switches and LEDs can be used to “scale” the output from 0 – 99990 Hz, 0 – 999900 Hz and 0 – 9999000 Hz.

How would you expect the accuracy of the frequency to change? How could this be improved?

Display_Interface Circuit.

A sample circuit for the Display_Interface circuit can be found on Moodle. A similar circuit has already been used when you were introduced to the Xilinx development boards as part of the EENG5680 (was EL568) introduction to VHDL and Xilinx.

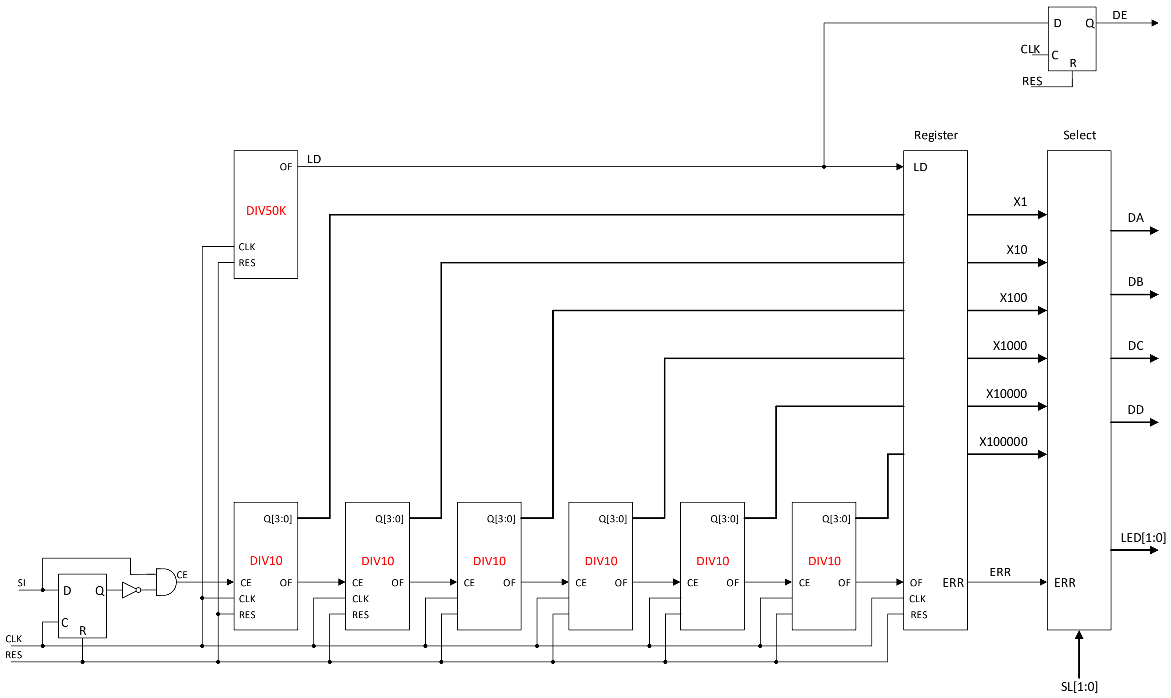

Figure 6. Suggested FREQUENCY_CALCULATOR circuit

Synchronisation

The data input (**SENSOR_IN**) and reset (**CLR**) signals are asynchronous to the system clock (**CLK**) and will first need to be synchronised *before* applied to the rest of the system (as shown by the SYNC components in Fig.3). A typical synchronising circuit is shown in Fig 7. You will be asked to explain in your report how you think this circuit works.

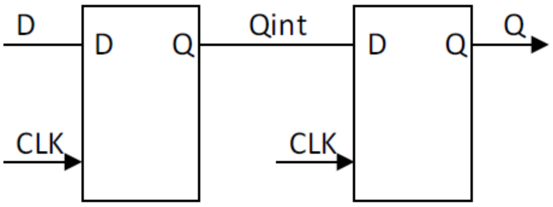

Figure 7. Suggested SYNC circuit

Suggested Design Steps.

The block diagram in Fig. 3 indicates that a number of components are needed. It is advisable to use hierarchy where appropriate to reduce the complexity of the circuits you are designing. In other words, design and test/simulate each component separately.

- Build the clock divide by 2 circuit (DIV2) (see Figs 3)
- Build the data synchronisation circuits (SYNC) (see Fig. 3 & 7)
- Build the Phase_Shifter circuit and develop a test bench to test its operation is consistent with the timing diagram shown in Figure 5.
- Build the Frequency_Counter Circuit and a simple test bench to test its function.
- Combine the above circuits into a TOP_LEVEL circuit description and develop a test bench to verify its operation.
- Using the information provided in Fig 8 and the .UCF file provided, synthesise your design in Xilinx ISE®. Correct any errors or warnings produced during the synthesis and routing of your design before loading the bit file onto the Xilinx device on the Spartan6 board provided.

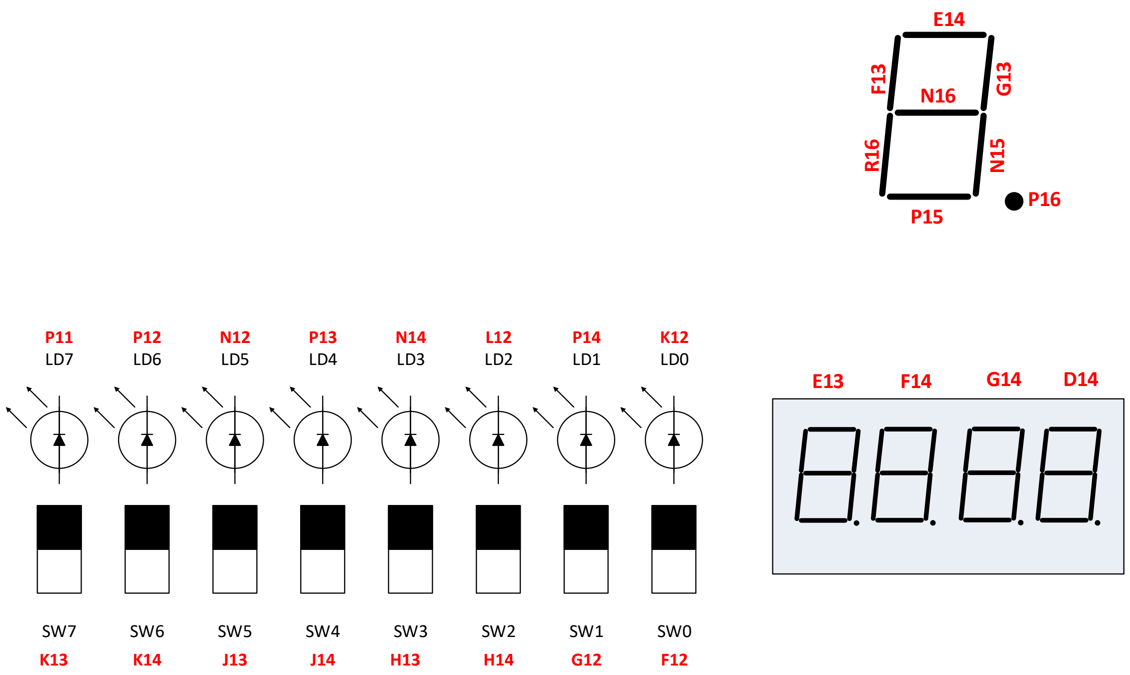

Figure 8. Spartan-6 on-board LEDs, Switches and Display Elements (with appropriate pin numbers)

Marking and Submission

The <u>code and simulation</u> outputs will be checked and marked during the lab sessions by the demonstrator. **<u>Note</u>**: Your design code and simulation outputs <u>must</u> be completed, checked and marked by the demonstrator by the end of the last lab session on KV Week 19 (week beginning 6 Dec.). Your code should be properly and clearly commented (the quality of the comments will be taken into account in the marking).

The mini-project <u>report</u> should be submitted online by

<u>Group 2: 12:00 noon, Tuesday 14 December (KV Week 20)</u>

<u>Group 1: 12:00 noon, Friday 17 December (KV Week 20)</u>

When you submit your report you will also need to include a .zip file containing all your code, your Xilinx ISE® Project and project directories. All work is to be submitted through Moodle.

Your report should include a block diagram of the circuit you have developed and a summary of the implementation statistics i.e., what FPGA resources have been used etc. (available from the Xilinx ISE® software), a discussion highlighting what has been achieved and design decisions that have been made during the process. There will also be some <u>technical questions</u> relating to the design you will have to provide answers for. A <u>template</u> for the report will be provided highlighting what needs to be added in each section.