Egg Timer Lab

Introduction

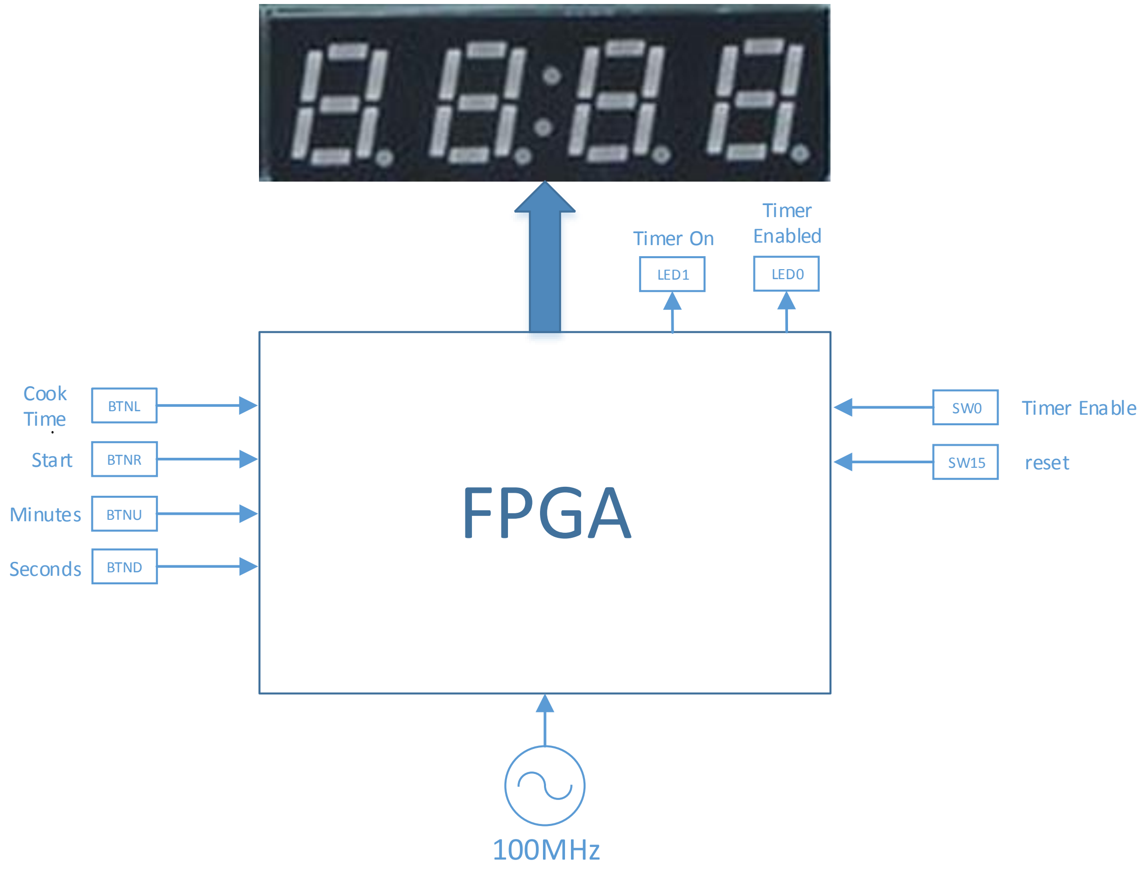
This lab will build upon what you have learned in previous labs. You will combine counters, finite state machined, encoders, registers and logic to build a basic FPGA based egg timer. You will then add your own additional features to improve the egg timer functionality. You will also be considering important aspects of the design process that will be evaluated through Graduate Attributes (GAs).

Objectives

After completing this lab you will be able to design a complex digital system.

High Level Specifications

The basic egg timer high level block diagram is as shown below:



Inputs to the FPGA include: 100MHz clock, BTNL/BTNR/BTNU/BTND push buttons, SW0/SW15 slide switches. Outputs from the FPGA include: LED0/LED1 LEDs and control signals for four seven segment displays.

The seven segment displays display the cook time in minutes (two left hand digits) and seconds (two right hand digits). LED0 is illuminated when the egg timer is enabled. LED1 blinks at a 1 second interval (i.e. one second on and then one second off) when the egg timer is active and counting down until it reaches a count of 0 minutes and 0 seconds.

When BTNL is pressed and held the programmed cook time can be changed by pressing BTNU (each press increments the minutes) and BTND (each press increments the seconds) until BTNL is released. Note the minutes and seconds counts will wrap around to 0 after they hit a maximum of 59. The maximum programmed cook time is therefore 59 minutes and 59 seconds. Also note that when BTNL is pressed, if the timer is currently active and counting down, it will stop and will not resume until the start button is pushed with the timer enable switch turned on.

When BTNR is pressed the egg timer will start counting down (note: the count down time will be displayed on the seven segment displays) and will count down from the programmed cook time value to 0 unless the cook time button (BTNL) is pushed or the timer enable switch is turned off, either of which will cause the programmed cook time to be displayed on the seven segment displays and LED1 to be turned off.

The BTNU/BTND push buttons should be debounced (i.e. do multiple samples of the button signal so that mechanical bouncing of the switch connection is ignored) so that the cook time is only incremented once for every button push.

SW0 enables (switch in forward position) or disables (switch in backward position) the egg timer. When the timer is enabled LED0 is on.

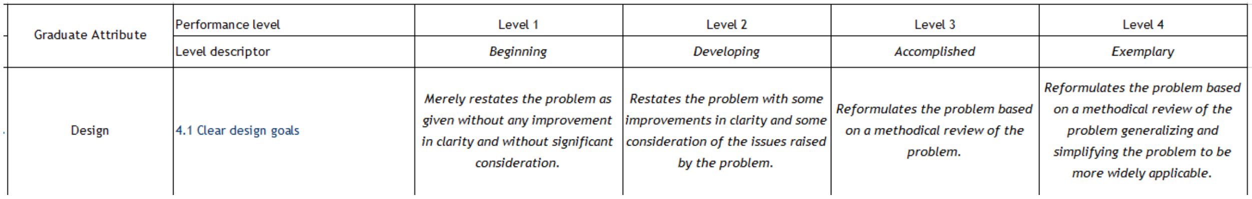
SW15 resets (switch in forward position) the egg timer circuits and sets the programmed cook time to 00:00 minutes and seconds with LED 1 off.

The 100MHz clock provides the master clock from which all other clocks are derived.

Graduate Attributes

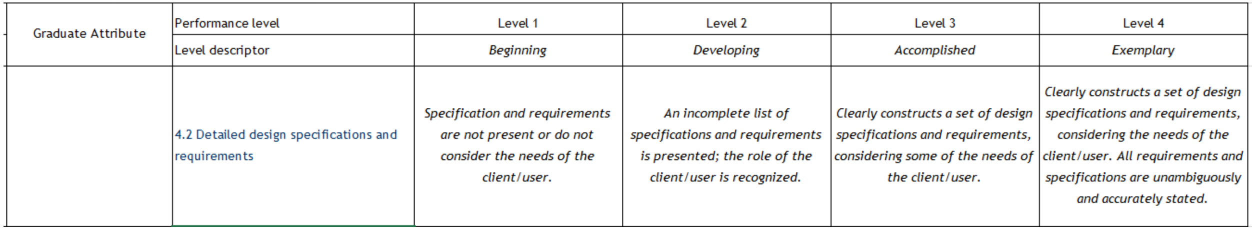
Your design will be assessed on the following GAs (see GA numbers and marking rubics below). You should directly address each of these attributes individually in the text of your final report (i.e. your report should have sections GA 4.1, GA 4.2 etc. that provide information for the GA and/or explain how you addressed the GA)

4.1 - Design: Clear design goals



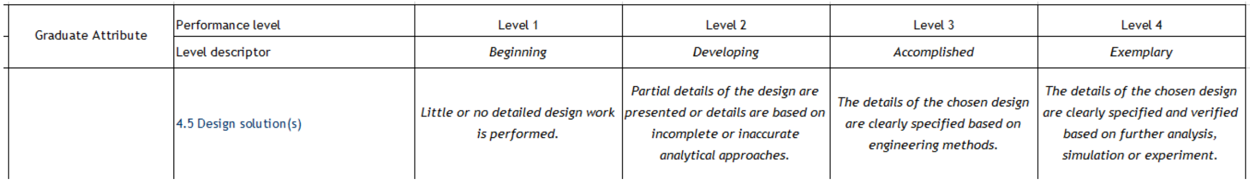
Graduate Attribute	Performance level	Level 1	Level 2	Level 3	Level 4
	Level descriptor	*Beginning*	*Developing*	*Accomplished*	*Exemplary*
Design	4.1 Clear design goals	*Merely restates the problem as given without any improvement in clarity and without significant consideration.*	*Restates the problem with some improvements in clarity and some consideration of the issues raised by the problem.*	*Reformulates the problem based on a methodical review of the problem.*	*Reformulates the problem based on a methodical review of the problem generalizing and simplifying the problem to be more widely applicable.*

4.2 - Design: Detailed design specifications and requirements



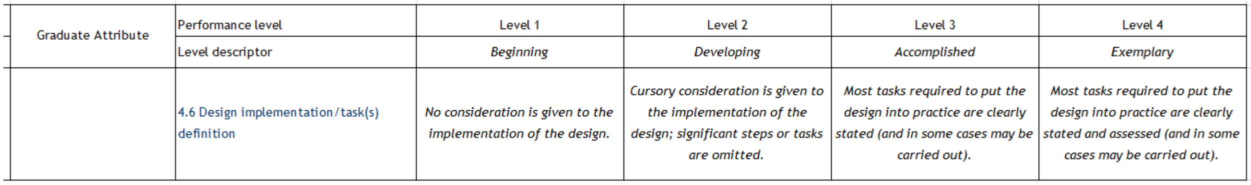
Graduate Attribute	Performance level	Level 1	Level 2	Level 3	Level 4
	Level descriptor	*Beginning*	*Developing*	*Accomplished*	*Exemplary*
	4.2 Detailed design specifications and requirements	*Specification and requirements are not present or do not consider the needs of the client/user.*	*An incomplete list of specifications and requirements is presented; the role of the client/user is recognized.*	*Clearly constructs a set of design specifications and requirements, considering some of the needs of the client/user.*	*Clearly constructs a set of design specifications and requirements, considering the needs of the client/user. All requirements and specifications are unambiguously and accurately stated.*

4.4 - Design: Design solution(s)



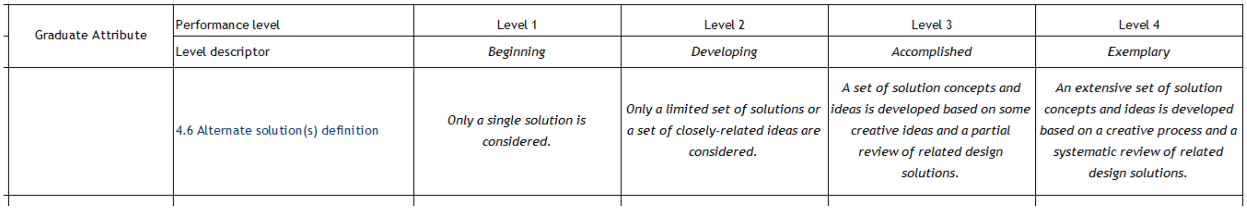
Graduate Attribute	Performance level	Level 1	Level 2	Level 3	Level 4
	Level descriptor	*Beginning*	*Developing*	*Accomplished*	*Exemplary*
	4.5 Design solution(s)	*Little or no detailed design work is performed.*	*Partial details of the design are presented or details are based on incomplete or inaccurate analytical approaches.*	*The details of the chosen design are clearly specified based on engineering methods.*	*The details of the chosen design are clearly specified and verified based on further analysis, simulation or experiment.*

4.5 - Design: Design implementation / task(s) definition



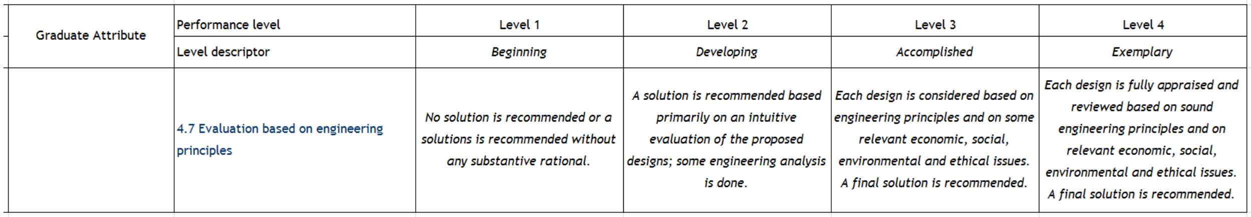
Graduate Attribute	Performance level	Level 1	Level 2	Level 3	Level 4
	Level descriptor	*Beginning*	*Developing*	*Accomplished*	*Exemplary*
	4.6 Design implementation/task(s) definition	*No consideration is given to the implementation of the design.*	*Cursory consideration is given to the implementation of the design; significant steps or tasks are omitted.*	*Most tasks required to put the design into practice are clearly stated (and in some cases may be carried out).*	*Most tasks required to put the design into practice are clearly stated and assessed (and in some cases may be carried out).*

4.6 - Design: Alternate solution(s) definition and evaluation



Graduate Attribute	Performance level	Level 1	Level 2	Level 3	Level 4
	Level descriptor	*Beginning*	*Developing*	*Accomplished*	*Exemplary*
	4.6 Alternate solution(s) definition	*Only a single solution is considered.*	*Only a limited set of solutions or a set of closely-related ideas are considered.*	*A set of solution concepts and ideas is developed based on some creative ideas and a partial review of related design solutions.*	*An extensive set of solution concepts and ideas is developed based on a creative process and a systematic review of related design solutions.*

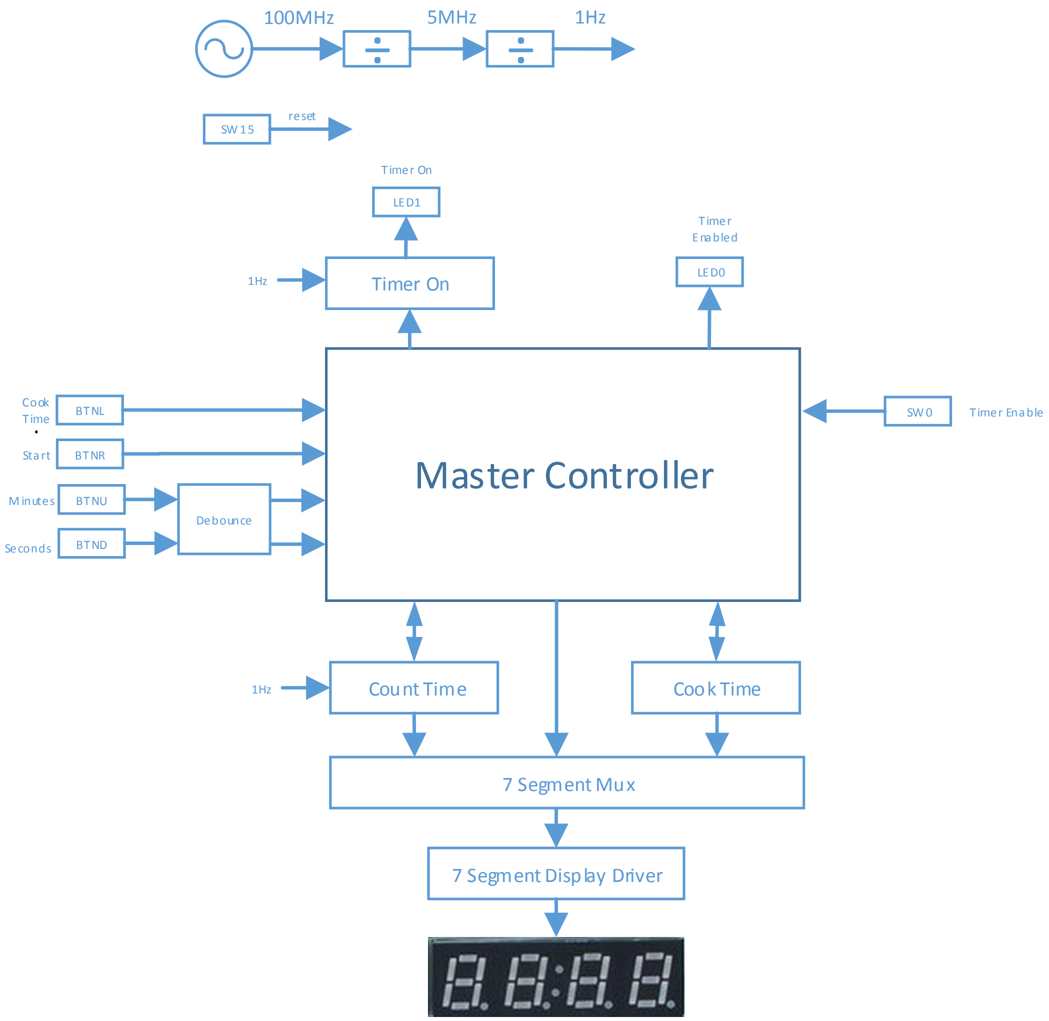
4.7 - Design: Evaluation based on engineering principles



Graduate Attribute	Performance level	Level 1	Level 2	Level 3	Level 4
	Level descriptor	*Beginning*	*Developing*	*Accomplished*	*Exemplary*
	4.7 Evaluation based on engineering principles	*No solution is recommended or a solutions is recommended without any substantive rational.*	*A solution is recommended based primarily on an intuitive evaluation of the proposed designs; some engineering analysis is done.*	*Each design is considered based on engineering principles and on some relevant economic, social, environmental and ethical issues. A final solution is recommended.*	*Each design is fully appraised and reviewed based on sound engineering principles and on relevant economic, social, environmental and ethical issues. A final solution is recommended.*

Design

As with all designs, there are lots of detailed design specifications that need to be worked out and it is important that you do some planning before you start coding the design (note: make sure to consider the graduate attributes from the beginning of the design process). The first step is to come up with an initial block design (note: you may change the block design as you start coding to improve or simplify the design) for the circuits and identify the inputs and outputs and high level operation of each block. A possible block diagram and block operation descriptions for the egg timer are shown below:



Clocking

The 100MHz clock is divided down to 5MHz and then divided again to generate a 1Hz control signal (i.e. 1Hz is not a new clock just a control signal that enables operations at 1Hz, all flip-flops in your design should be using the 5MHz clock). The 5MHz clock and 1Hz control signal are distributed to the other blocks of the design as required (note: most blocks will use the 5MHz clock but only some blocks will require the 1Hz control signal, for example the Timer On block).

Reset

The reset signal is distributed to the other blocks of the design as required.

Debounce

The debounce circuit receives the BTNU/BTND signals, as well as, the 5MHz clock and generates debounced versions of the signals. When the buttons are pushed it may take up to 10mS for the signal to settle so the debounce circuit needs to generate a low speed internal signal for sampling the button presses and check that there are no changes on the individual input signals before updating the individual output signals (i.e. there is a separate debounce circuit for each button input).

Master Controller

The master controller receives the button press and debounced button press signals, as well as, the timer enable switch signal and the 5MHz clock. The master controller is responsible for a number of functions as follows:

- If the timer enable signal is high it turns on the timer enabled LED0

- If the timer enable signal is high and the timer counts down to 0 the timer is disabled and LED1 is tuned off
- Based on the button presses, control signals need to be generated for the Cook Time block to set the programed cook time
- Based on the button presses, control signals need to be generated for the Count Time block to set the count time
- Based on the button presses, control signals need to be generated for the 7 Segment Mux block to select which data to display on the 7 segment display.

Timer On

The Timer On block receives a signal from the master controller and the 1 Hz control signal (note: it also requires the 5MHz clock). When the master controller signal is high it turns on the LED1 at 0.5 Hz rate with 50% duty cycle (i.e. on for 1s and off for 1s …). When the master controller signal is low it turns off LED1.

Count Time

Part of the Count Time block is similar to the real time clock you did in a previous lab. You need to take the 1Hz control and divide it down further to generate a 1/60 Hz control signal (or you could generate a separate 1/60Hz control signal from the 5MHz clock) that will be fed into a minutes counter that count down from a maximum of 59. In addition, you need to have the count time counters programmable and the program values need to be incremented or decremented based on control signals from the master controller. The program values must stay within the range of the minutes and seconds count time counters. You need to feed a signal back from the count time block to the master controller so it knows when the count time has reached 0. Finally, based on a control signal from the master controller, you need to decide which data to output to the 7 Segment Mux; the programmed cook time or the count time.

7 Segment/LED15 Mux

The 7 Segment Mux block receives 7 segment data signals from the Count Time and Cook Time blocks. It also receives a control signal from the Master Controller. When the Master Controller control signal is low it passes the 7 segment data from the Count Time block to its output. When the control signal is high it passes the data from the Cook Time block to its output. The output data drives the 7 Segment Display Driver.

7 Segment Display Driver

The 7 Segment Display Driver block is similar to the one used in the real time clock you did in a previous lab. It takes in the display data and a 5MHz clock and generated the control signals to drive the four digit 7 Segment display. Note: For debug purposes you may want to display faster time (i.e. run the Time block at a higher frequency) to make sure everything is working before switching to the slower operation of minutes and seconds.

At this point you may be thinking, that's a lot of circuit design and wouldn't it be easier to do this with a microcontroller and a software program. You would be right, but the point is to build a circuit with hardware as a learning exercise and to realize, if we had different application that needed to run at very high speed a hardware based solution like you're building here would be much faster than a microcontroller based solution. In reality, most systems today are a combination of microcontrollers and high speed hardware.

When you have the basic egg timer working, demonstrate the working circuit and code to the TA. Include the basic egg timer code as part of your final report (see below).

Innovation

Once you have the basic egg timer working it’s time for you to add your own additional design features to improve the egg timer and make it more usable. These may be simply operational changes or the use of other parts of the FPGA board (e.g. memory blocks (internal and external), VGA output, mono audio output, more 7 segment displays, LEDs (including tricolor LEDs), buttons or switches, UART/USB/Ethernet interfaces or PMOD connectors with external PMOD modules (https://store.digilentinc.com/brands/Digilent-Pmod.html).

Vivado has many build in IP blocks to help you use these features including a microcontroller (microblaze) with automatic SDK generation (https://www.xilinx.com/video/hardware/creating-a-simple-microblaze-design-in-ip-integrator.html). Note: the optional tutorials can help with using features like the microblaze controller and PMOD connectors.

When you have your custom egg timer working, demonstrate the working circuit and code to the TA. Also, submit a final report that details you custom design including graduate attributes, block diagrams, design and operational principles and design choices, Verilog/software code, verification (e.g. simulations and test benches), hardware testing, final hardware configuration, lessons learned, conclusions and references.

Conclusion

In this lab, you learned how to design a complex digital system with multiple inputs and outputs and created your own custom enhancements to the basic design.