

Electrical and Computer Engineering EE3623 Embedded Systems Design Laboratory



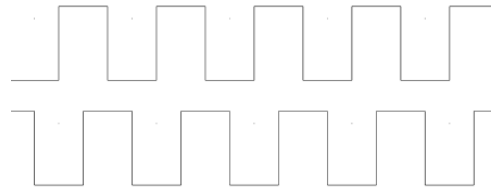
Rotation Measurement using the Rotary Shaft Encoder

Dennis Silage, PhD

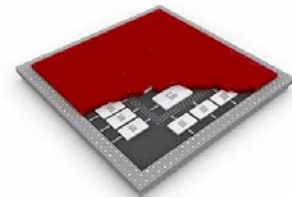
The rotary shaft encoder of the Spartan-3E Starter Board can be used to input an incremental digital signal to a process. In this Laboratory, the shaft rotation signal in a fixed time interval obtained by *slowly* turning the rotary shaft encoder is to be displayed on the LCD. A nested set of Verilog behavioral synthesis modules are required, however the modules *iterativediv.v* in Listing 2.3, *clock.v* in Listing 3.3, *rotary.v* in Listing 3.10 and *lcd.v* in Listing 3.14 of the EDPGA text are available and are an example of *design reuse*.



The frequency generator project in Listing 4.12 also uses the rotary shaft encoder and provides *design reuse* support here.



The number of shaft rotations is counted by each positive edge of a rotary encoder signal. The time interval is to be 10 seconds. The number of shaft rotation signals counted should be then displayed on the LCD on the first line, upper left position. Since Verilog behavioral synthesis modules execute in parallel, no delay in the counting interval is required while the LCD updates its display.



The initial project is to count and display the shaft rotation signal in one direction, for example clockwise, within the 10 second interval. Additional credit would be given if the clockwise shaft rotations measured clockwise were decremented by those measured from a counter-clockwise rotation during the 10 second interval. This requires the use of both rotary shaft encoder signals and their phase relationship as described on page 114 of the EDPGA text.

The suggested design task for this Laboratory would include a 10 second interval *control signal* derived from the *clock.v* module, a shaft rotation count *datapath* module and an overall *controller* module to coordinate the process. You will have to observe how many digits are required to display the shaft rotation and modify *iterativediv.v* appropriately. Leading zeros in the display (such as '018' rendered as '18') would be suppressed as in good embedded design practice.

Verilog Simulation and Task Extension

The Laboratory is now extended for one additional week with a requisite Verilog simulation and an additional rotary shaft encoder project.

1. Both the initial project and this extension are to be simulated to verify the performance of your HDL modules.
2. The extension project is to produce a *moving average* 60-second count of the rotary shaft rotation as a sum of 6 10-second rotation counts from the initial project. A 60-second moving average, unlike the accumulated output every 10 seconds, outputs the last 60-seconds of the rotation count continuously every 10 seconds.

A new 10-second count at the current time T_o is added to the accumulated 60-second count but the oldest 10-second count at the time T_{-60} seconds is first subtracted from the total. The new 60-second count as a sum of the six 10-second counts, T_{-50} through T_o in the Figure, is then displayed on the LCD.

The six 10-second rotation counts must be stored and a pointer is used to point to the *oldest* location where the count to be subtracted is stored, then the *newest* 10-second count is stored there. The pointer is then advanced but with a *wrap-around*. The storage registers are called a *data carousel* (also known as a *register file*) because of the apparent wrap-around that occurs after the first six 10-second counts are stored.

The moving average *smooths out* the rotation count over a 60-second interval.

