Quadrature Encoder Decoding

Updated Feb 26.

Introduction

A quadrature encoder is a mechanical motion-input device that can measure rotation or linear displacement. It has two logic outputs that are offset so that an interface circuit can determine both the direction and amount of motion as described below.

Typical applications are digital controls and position measurement.

In this lab you will use a sequence detector and an up/down counter to track the shaft angle of the EN16-H rotary quadrature encoder included in your ELEX 2117 parts kit[1].

The encoder has two switch contacts labelled A and B and a common connection labelled C:
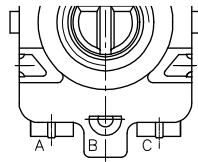
The two switches turn on and off as the shaft rotates:
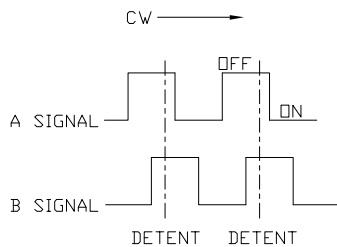
The encoder has 24 detent positions per revolution.

If pull-up resistors are used on A and B so they are low when the switches are ON, then when the shaft moves clockwise (cw) from one detent position to another the Gray-coded sequence of values on AB is: 11, 01, 00, 10, 11,... When moving in the counterclockwise (ccw) direction the sequence is reversed (11, 10, 00, 01, 11, ..):

---

[1] The EN16 datasheet is available on the course web site.
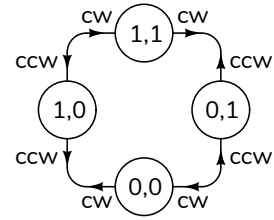
Thus each transition allows us to determine the direction of motion:

- transitions from 10 to 11, 11 to 01, 01 to 00 and 00 to 10 indicate clockwise rotation, while

- transitions from 00 to 01, 01 to 11, 11 to 10 and 10 to 00 indicate counterclockwise rotation, and

- transitions from 00 to 11 or 01 to 10 should not be possible.

A sequence detector can determine the direction of motion: comparing one set of switch inputs to the previous input we can determine when the shaft has moved and in which direction.

This rotary encoder uses mechanical switches and these have contact bounce that can appear as rotation. Debouncing the switch inputs helps prevent this.

State Machine Description

In this lab you will design and implement a sequence detector state machine that implements a "decoder" which determines the direction of rotation. This state machine has two bits of input (**a** and **b**) and two bits of output (**up** and **down**). It can be described as a Moore state machine with four bits of state.

A sequence detector state machine uses a shift register to "remember" past inputs. In this case there are two input bits at a time and we need to look back across two pairs of inputs so the state transition table could be written as:

| current state	input	next state
wxyz	ab	yzab

where **wxyz** and **ab** can be any four- and two-bit values. Referring to the diagram and description above, the **up** and **down** outputs can be derived from the state. Some examples would be:

state	output (**up**, **down**)
1101	1,0
0111	0,1
1100	0,0

Specifications

You must write a module named **qdecoder** that implements the quadrature encoder decoder described above. Your module will be instantiated in a top-level module that uses your decoder's **up** and **down** outputs to increment or decrement a counter. This count will be shown in octal (base 8) on the four-digit 7-segment LED display. Turning clockwise should increment the count and turning counterclockwise should decrement the count. Returning the shaft to its starting position should restore the display to the starting value.

The inputs and outputs of your module are:
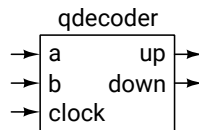
and it is declared as:

```systemverilog
module qdecoder
  ( input logic a, b,
    input logic clock,
    output logic up, down ) ;
```

Components

You will need the same components as in the previous lab with the rotary encoder substituted for the two pushbutton switches.

Procedure

Restore the **lab6.qar** project archive. Design a sequence detector that sets the **up** and **down** outputs based on the **a** and **b** switch inputs and add your code to the **qdecoder.sv** file in the project folder.

If necessary, change the pin assignments to match the CPLD pins you're using.

Compile and test your code and submit the files described below.

CPLD I/O

The default connections to the CPLD are shown in the following diagram:
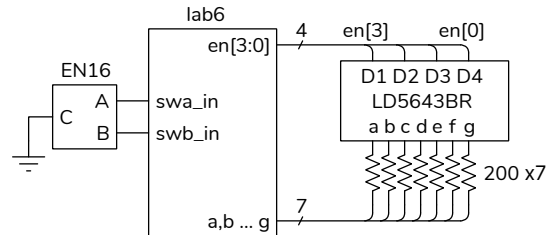
The pin assignments for the LED are the same as in the previous lab and internal pull-up resistors are configured on the **A** (**swa_in**) and **B** (**swb_in**) switch inputs. Connect the encoder's terminal **C** to ground. The pin assignments are shown below:

To	Assignment Name	Value
a	Location	PIN_33
b	Location	PIN_44
c	Location	PIN_38
d	Location	PIN_34
dp	Location	PIN_36
e	Location	PIN_30
en[3]	Location	PIN_35
en[2]	Location	PIN_50
en[1]	Location	PIN_48
en[0]	Location	PIN_42
f	Location	PIN_52
g	Location	PIN_40
clock	Location	PIN_12
swa_in	Location	PIN_99
swb_in	Location	PIN_97
swa_in	Weak Pull-Up Resistor	On
swb_in	Weak Pull-Up Resistor	On

The wiring to the 7-segment LED and rotary encoder is shown below:
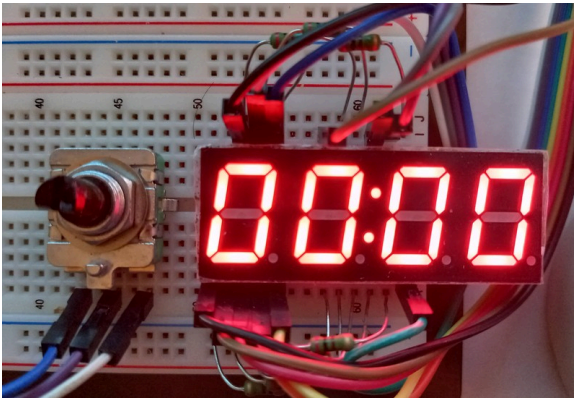
and the connections to the CPLD pin headers are the same as in the previous lab.

Submission

To get credit for completing this lab, submit the following to the Assignment folder for this lab on the course website:

1. A PDF document containing:

   (a) A block diagram (schematic) of your sequence detector.

   (b) A listing of your `qdecoder.sv` file.

   (c) A screen capture of your compilation report (**Window > Compilation Report**) similar to:

   (d) The schematic created by **Tools > Netlist Viewers > RTL Viewer**, double-clicking on the `qdecoder:q0` block to expand it, and then **File > Export**... . It might look similar to:
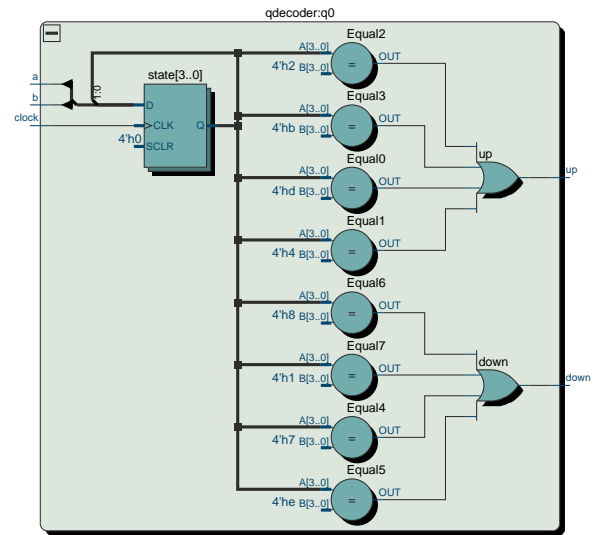
Note that this is the schematic of your decoder only, not of the complete design.

2. A video of your breadboard showing:

   (a) an initial count value (typically 0000 on power-on)

   (b) the count increasing by at least 100 (base 8) as the shaft turns clockwise

   (c) the count decreasing by at least 200 (base 8) as the shaft is turned counterclockwise

   (d) the count increasing back to the original value as the shaft returns to the original angle

Turn the encoder with a screwdriver that has a piece of tape attached as shown in the sample video to make it easy to see the shaft angle.