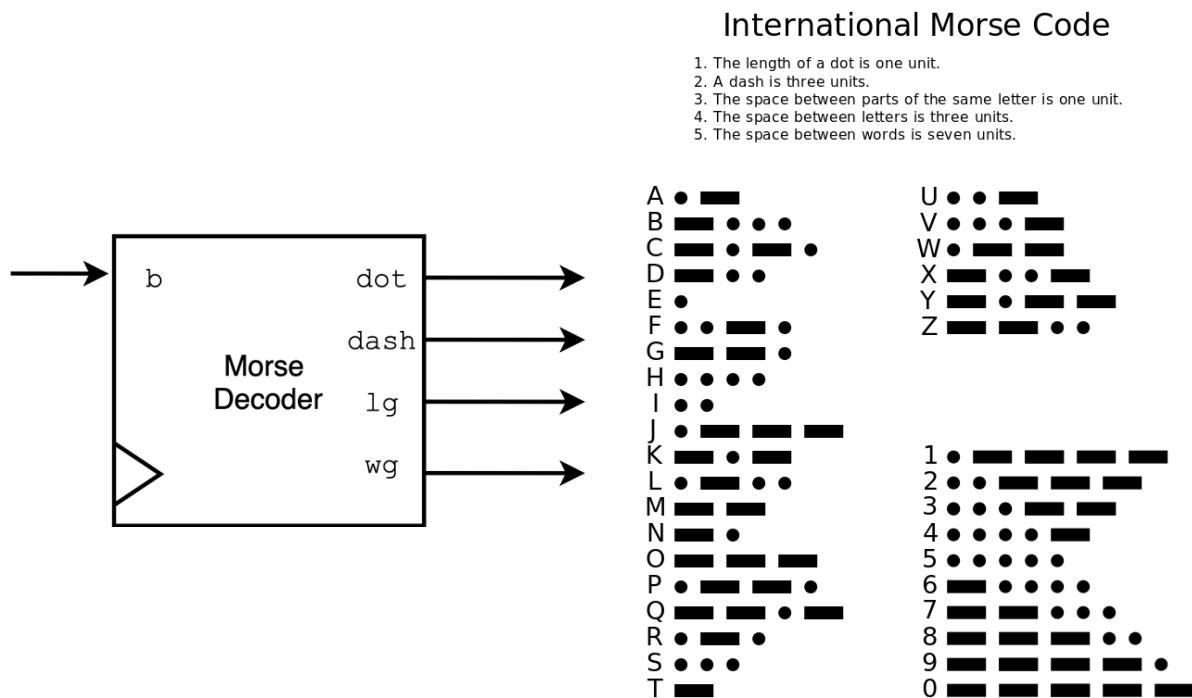


Lab 10

Part1: Morse Decoder

In this lab you will design a Morse Decoder that takes an input *b* and generates the following signals:

- *dot*: when the input *b* is asserted for one unit
- *dash*: when the input *b* is asserted for 3 units
- *lg*: letter gap, indicating the end of letter
- *wg*: word gap, indicating the end of a word (of multiple letters)



You are free to use whatever components in your design, just make sure to name your top file `morse_decoder.v` (we will use this file in future lab projects)

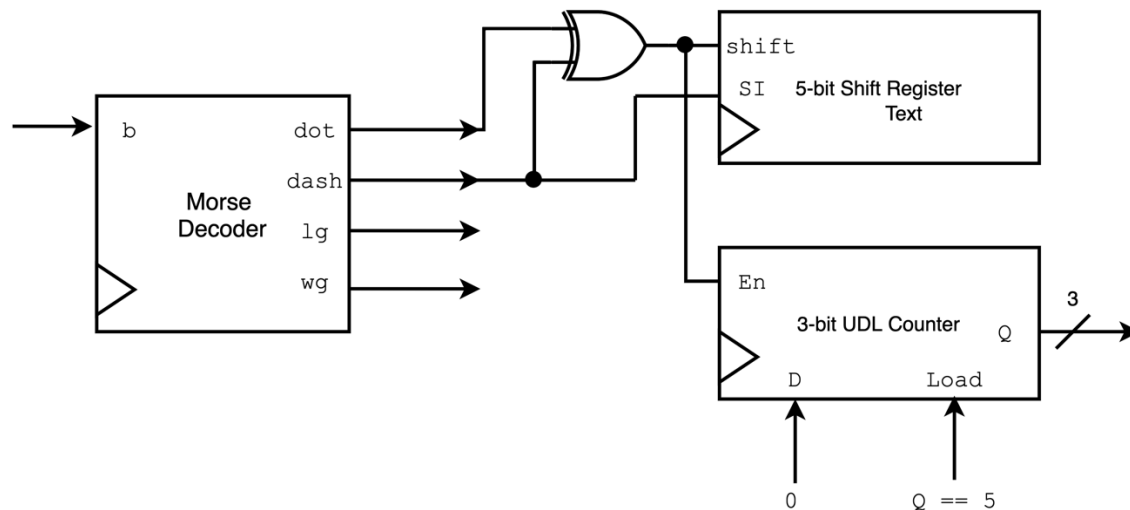
Hints:

- You might want to use a combination of timer and counter to keep track of how many units of time have passed
- Online resources indicate that 50 ms is a typical unit of time; however, in my demo I found 100 ms easier to work with.

There is no FPGA implementation for this part

Part2: Morse Decoder Application

Verify the functionality of your decoder by implementing the following circuit.



FPGA Implementation:

- Connect input b to one of the push buttons
- Display the content of the shift register on the first 5 seven-segment digits.
- Display the content of the counter on the 8th seven-segment digit.
- It is important to keep track of the number of segments in a Morse code, because the number range between 2 and 5.
- The D, Load inputs reset the value of the counter to 0 after shifting 5 dits/dashes.

Submission check list:

- [] All Verilog code you generated or modified
- [] All testbenches written
- [] Embed all screenshot of your testbench output in your README.md
- [] Embed all block diagram or state diagrams generated in your README.md
- [] Short videos demonstrating each of the parts you implemented on the FPGA