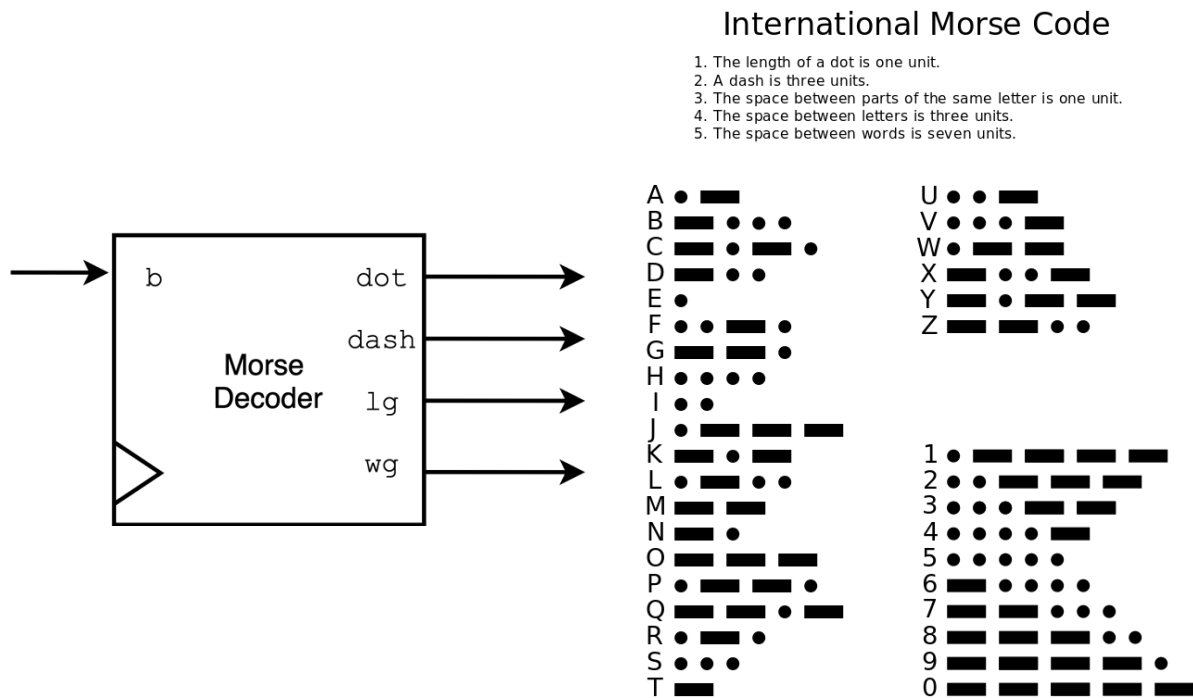


Lab 10

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)

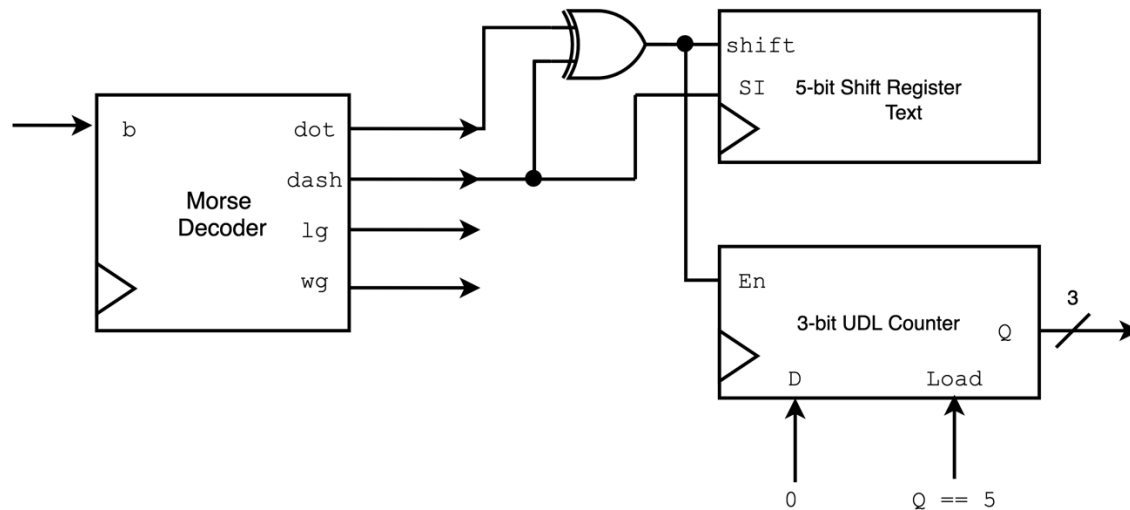
Hints:

- You might want to use a combination of timer and counter to keep track of how many units of time have passed.
- The patterns are defined by the number of unit times, it might be useful to use a range instead. For example, if a dot is one unit of time, then use 0 to 2 units to detect a dot. Similarly, if a dash is three units use 3 to 5 units to detect it.
- Use 50 ms as one unit of time (i.e. a dot is about 50 ms and a dash is about 150 ms)

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 1 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