Create a directory, then create a library where we put the rle\_enc.v file in.

This is for count bits where we check if new bitstream.

* If yes then reset the shift\_buff[0] (the ID bit) to 0 or 1 (depending on what we will be counting).
  + Set the new\_bitstream to 0 so that it knows a new bitstream is coming in.
  + Iterate bit\_count
* If no then
  + if the shift\_buffer at index 0 is the value type we are looking for then bitcount increases
    - Iterate through the bit count
  + Else the bitstream was done counting

| COUNT\_BITS: begin    if(new\_bitstream) begin  value\_type <= shift\_buf[0];  new\_bitstream <= 0;  bit\_count = bit\_count ++; //If this doesn't work then use bit\_count + 1  end  else begin  if(shift\_buf[0] == value\_type) begin  bit\_count = bit\_count + 1; //If this doesn't work then use bit\_count + 1  end  else begin  new\_bitstream <= 1;  end  end end |
| --- |

If not newbitstream then shift the bits in the buffer then iterate the shift count to see how many consecutive bits there are.

| SHIFT\_BITS: begin  if(!new\_bitstream) begin  shift\_buf = shift\_buf >> 1;  shift\_count ++ // If this doesn't work then use bit\_count + 1  end end |
| --- |

For the other bubbles it's just the case statement

Test code:



| Always begin  #1 clkt = !clkt; end  initial begin   rest = 1;  #2 rest = 0; end  always @ (posedge wr\_reqt) begin  send\_readyt = 1; end  Always @ (negedge wr\_reqt) begin  send\_readyt = 0; end  always @ (posedge rd\_reqt) begin  recv\_readyt = 1; end  always @ (negedge rd\_reqt) begin  recv\_readyt = 0; end  initial begin clkt = 0;   demo1 = 16'b1111111100000000;  in\_datat = demo1[7:0];  #40 in\_datat = demo1[15:8];  #50;    demo2 = 8'b11110000;  in\_datat = demo2[7:0];  #55;   demo3 = 24'b111000111111000000011111;  in\_datat = demo3[7:0];  #45 in\_datat = demo3[15:8];  #45 in\_datat = demo3[23:16]; end |
| --- |