```
1   `timescale 1ns / 1ps
2   /**********************************************************************************
3   *
4   * Engineer: Justin Maeder; 015906629
5   * Email:    Justin_maeder@hotmail.com
6   * Filename: ticker.v
7   * Date:     February 13, 2019
8   * Version:  14.7
9   * Description: The ticker module takes the 100MHz Nexys 4 clock as well as a 2-bit
10  *              constant value (k), and outputs a desired clock.
11  *
12  **********************************************************************************/
13  module ticker(clk,reset,k,tick);
14
15     // Input & Output Declarations
16     input clk,reset;
17     input [1:0] k;
18     output tick;
19
20     // Hold 20-bit values of flops
21     reg  [1:0] count,ncount;
22
23     // Tick = "1" if if count meets desired constant
24     assign tick = (count == k);
25
26     // Combinational block to increment ncount until tick = "1", then reset ncount.
27     always @ (*) begin
28        if (tick) ncount = 2'b0;
29        else      ncount = count + 2'b1;
30     end
31
32     // Sequential block to hold state as count increments
33     always @ (posedge clk, posedge reset) begin
34        if (reset) count <= 2'b0;
35        else       count <= ncount;
36     end
37
38  endmodule
```