```
1
2   module Timing_Unit
3    #(parameter cnt_size =4)
4     (
5           input clk,rst,cnt_en,clear,
6           output reg [cnt_size-1:0] T
7           // output reg T0,T1,T2,T3,T4,T5,T6
8           );
9
10  // reg [cnt_size-1:0] T;
11
12   wire l_t;
13
14
15   always @(posedge clk)
16    begin
17          if(!rst|clear)
18                  T = 1'b0;
19          else if(1'b1)   //          else if(cnt_en)
20                  T = T+1'b1;
21    end
22
23  /*
24   always @(T)
25   begin
26          T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;
27   case(T)
28          0: begin T0=1; end
29          1: begin T1=1; T0=0; end
30          2: begin T2=1; T1=0; end
31          3: begin T3=1; T2=0; end
32          4: begin T4=1; T3=0; end
33          5: begin T5=1; T4=0; end
34          6: begin T6=1; T5=0; end
35          default: begin T0=0;T1=0;T2=0;T3=0;T4=0;T5=0;T6=0; end
36   endcase
37   end
38   */
39  endmodule
40
41  /*
42  module Clocking_Logic
43   (
44          // input Start,Stop,Wait,Done,
45          input cnt_en,
46          input Mclk,
47          output clk
48          );
49
50  always @(posedge Mclk)
51   begin
52    if(cnt_en)
53                  clk = Mclk;
54    else  // if (Stop | Wait)
55              clk = clk;
56   end
57  endmodule
58  */
59
```