The N8 controller application will require that you divide down a 50 MHz source clock. It is possible to get “close enough” to the desired divided clock frequency by dividing in powers of 2. This kind of counter is much simpler than counters that will require exact division and avoids subtle counter problems that can occur.

The following Verilog code illustrates how simple it is to make a $2^n$ counter. A simulation follows for this particular example. Notice that there is no need to have an end or reset condition since the counter automatically cycles to zero when it reaches the maximum count.

/*
The following is a power of 2 clock divider. Clockout = Clockin / (2^(n+1))
For example, if you wish to divide clockin by 16 set n = 3
*/
module p2divider(clockin, clockout);

input clockin;
output wire clockout;

parameter n = 3;
reg [n:0] count;

always@(posedge clockin)
begin
  count <= count + 1;
end

assign clockout = count[n];
endmodule