ButtonToLEDs.v

```
1  module Counter26BitEnableReset (
2      input clock,
3      input clear,
4      input enable,
5      input reset,
6      output reg[25:0] out);
7
8      always @(negedge clock, posedge clear)
9         if (clear)
10           out <= 0;
11        else if (reset)
12           out <= 0;
13        else if (enable)
14           out <= out + 1;
15
16 endmodule
17
18 module Comparator26Bit (
19     input [25:0] in_1,
20     input [25:0] in_2,
21     output out);
22
23     assign out = (in_1 == in_2);
24
25 endmodule
26
27 module ButtonToPulse (
28     input clock,
29     input clear,
30     input button,
31     output pulse);
32
33     wire[25:0] out;
34
35     Counter26BitEnableReset counter(clock, clear, ~button, button, out);
36     Comparator26Bit comparator(out, 26'b1, pulse);
37
38 endmodule
```

```
39
40  module Counter8BitEnable (
41      input clock,
42      input clear,
43      input enable,
44      output reg[7:0] out);
45
46      always @(posedge clear, negedge clock)
47          if (clear)
48              out <= 0;
49          else if (enable)
50              out <= out + 1;
51
52  endmodule
53
54  module ButtonToLEDs (
55      input clock,
56      input clear,
57      input button,
58      output[7:0] out);
59
60      wire x;
61
62      ButtonToPulse pulse(clock, clear, button, x);
63      Counter8BitEnable counter(clock, clear, x, out);
64
65  endmodule
66
67
68
69
70
71
```