**module ud\_counter(**

**C,**

**R,**

**UD, //high for UP counter and low for Down counter**

**COUNT**

**);**

**input C, R, UD;**

**output [3:0] COUNT;**

**//PLACE YOUR CODE HERE**

**reg [3:0] COUNT;**

**always @(posedge C or posedge R)**

**begin**

**if(R) //Active high R**

**COUNT <= 0;**

**else**

**if(UD) //Up mode selected**

**COUNT <= COUNT + 1; //Increment Counter**

**else //Down mode selected**

**COUNT <= COUNT - 1; //Decrement Counter**

**end**

**endmodule**

**module test\_ud\_counter;**

**reg clock;**

**reg reset;**

**reg mode;**

**wire[3:0] out;**

**ud\_counter ud (.C(clock), .R(reset),**

**.COUNT(out), .UD(mode));**

**always**

**#5 clock = ~clock;**

**initial**

**begin**

**clock = 1'b0;**

**reset = 1'b1;**

**mode = 1'b0;**

**#7**

**reset = 1'b0;**

**#50**

**mode = 1'b1;**

**#80**

**mode = 1'b0;**

**#400 $finish;**

**end**

**initial**

**$monitor($time, " UD = %b, COUNT = %b", mode, out);**

**endmodule**

|  |  |
| --- | --- |
|  |  |