HW#10 디지털 회로 설계 및 언어 월수 9:00~10:15 2015104027 박정진

**Design LD Driver (minor 0.1)**

Design LD Driver with

1 12-bits binary counter I\_Out (Predefined value of I\_out is 2000)

1 11-bits binary counter Counter\_1000

2 External input(main input) SW\_ON, LD\_ON

1 Flip-Flop LD\_ON\_reg

1 Master Clock 100MHZ (period = 10ns)

Operation

/\*

If design the circuit which satisfy the condition that take 10 seconds to fully increase, it takes too long time to simulate it, so I scale it down to 10ms. If you want to design the circuit acting like original assignment condition, just count 10e^6. It means change the condition that on state S1, if Counter == 999999 then increase I\_out, and on state S2, if Counter == 499999 then decrease I\_out.

However it is not efficient that storing 10e^6 decimal because it requires a lot of bits. Another way to count 10e^6 more efficient is design asynchronous counter. But in condition all register operates by master clock edge, it is not proper with given condition.

\*/

If SW\_ON=1, initiate the system operation by clearing I\_out and Counter\_1000 then

If LD\_ON=1, start increasing I\_out by counting Counter\_1000.

If Counter\_1000 == 11’d999, I\_Out I\_out+2

Else Counter\_1000 Counter\_1000+1’b1

I\_out increase linearly and time to fully increase is 10ms. when I\_out reaches 2000, stop increasing , keep its value before SW\_ON or LD\_ON change to 0.

If LD\_ON = 0, start decreasing I\_out value to twice about the rate of increase, and stop when I\_out becomes less equal than 1. (maintain the system switch on which means that remain the state on S2)

During increasing or decreasing, if LD\_ON switches to opposite value, increase or decrease value from that time and the rate of increase or decrease is same as before that we designated.

If SW\_ON = 0 after on any state except initial, change value of LD\_ON to 0 (For this reason, we need to declare reg type LD\_ON\_reg to store LD\_ON) and decrease I\_out value to twice the rate of increase, finally stop the system. On this state, operation is independent of LD\_ON.

If SW\_ON=1, LD\_ON=0 on initial state, the operation cycle repeats by clearing I\_out and Counter\_1000.

If SW\_ON = 0 on initial state, the system remains in the initial state

States

initial state :

increase I\_out :

decrease I\_out :

keep I\_out :

State diagram

A close up of a map

Description automatically generated

Register Transfer Operation

S0 : LD\_ON\_reg LD\_ON

if (SW\_ON) then I\_out 0, Counter\_1000 0

S1 : if (SW\_ON)

then LD\_ON\_reg LD\_ON

if (LD\_ON\_reg)

if(I\_out < 2000)

if(Counter\_1000 == 999) then I\_Out I\_out+2, Counter\_1000 0

else then Counter\_1000 Counter\_1000+1’b1

else then Counter\_1000 0

else then LD\_ON\_reg 1’b0

S2 : if (SW\_ON)

then LD\_ON\_reg LD\_ON

if (LD\_ON\_reg) then Counter\_1000 0

else

if(I\_out > 1)

if(Counter\_1000 == 499) then I\_Out I\_out-2, Counter\_1000 0

else then Counter\_1000 Counter\_1000+1’b1

else then LD\_ON\_reg 0

if(I\_out > 1)

if(Counter\_1000 == 499) then I\_Out I\_out-2, Counter\_1000 0

else then Counter\_1000 Counter\_1000+1’b1

S3 : if (SW\_ON)

then LD\_ON\_reg LD\_ON

if (LD\_ON\_reg = 0) then Counter\_1000 0

ASM Chart

A close up of a map

Description automatically generated

Verilog

A screen shot of a smart phone

Description automatically generatedA screenshot of a video game

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a cell phone

Description automatically generated

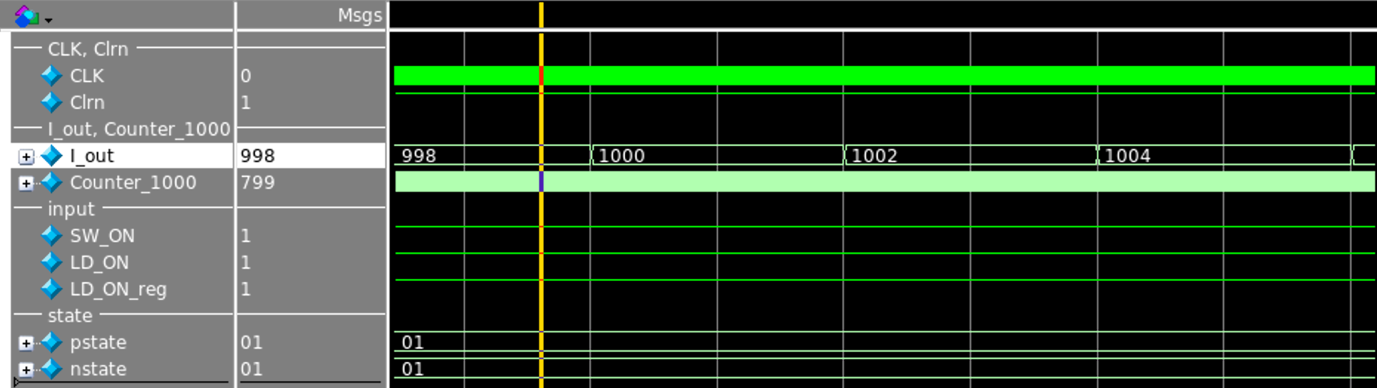
Testbench

A screenshot of a cell phone

Description automatically generated

RTL Simulation

1) Increasing I\_out



SW\_ON = 1

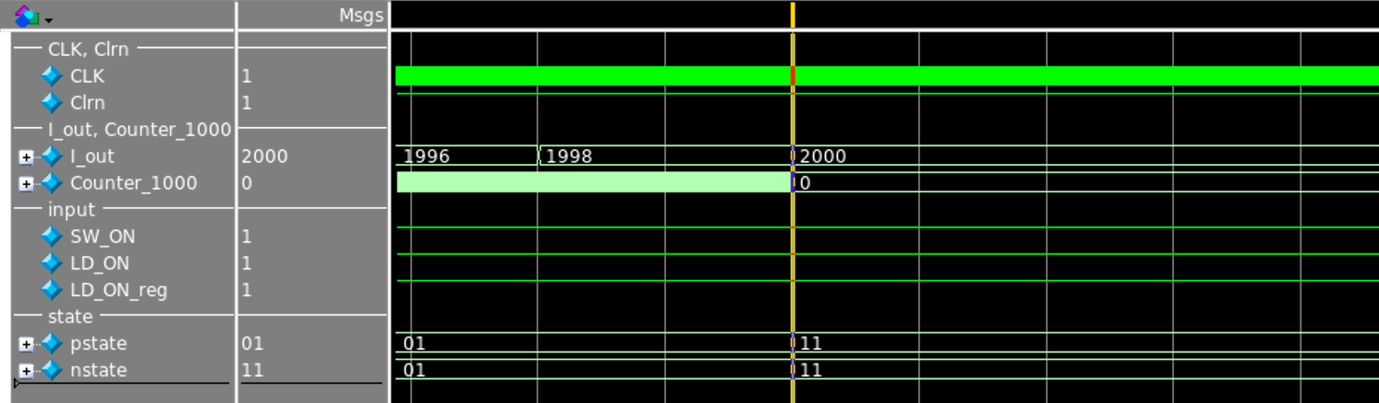
LD\_ON = 1

LD\_ON\_reg = 1

State : S1 (01)

I\_out increase in 2 unit time blocks

2) Keeping I\_out



SW\_ON = 1

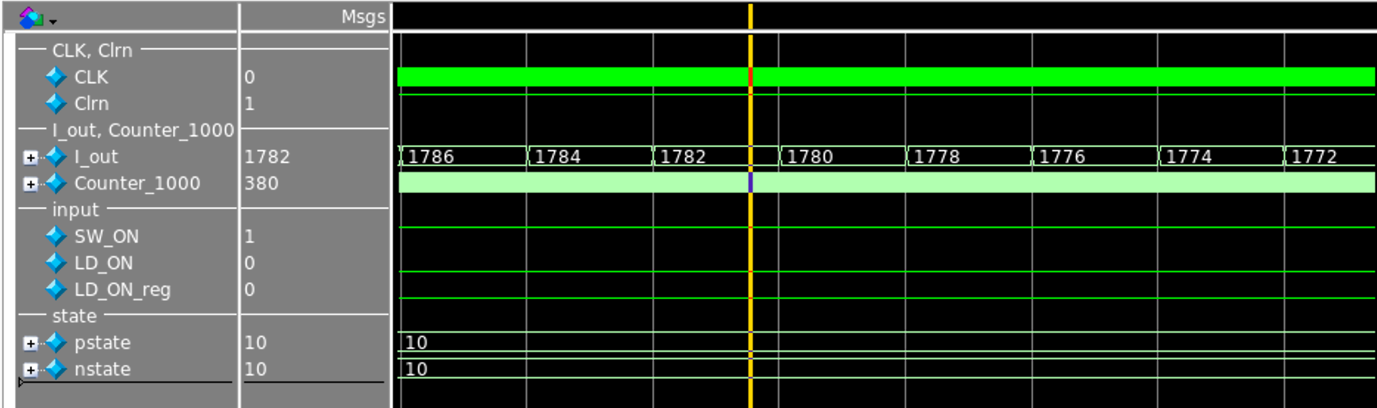
LD\_ON = 1

LD\_ON\_reg = 1

State : S3 (11)

Keeping I\_out (from 10ms)

3) Decreasing I\_out



SW\_ON = 1

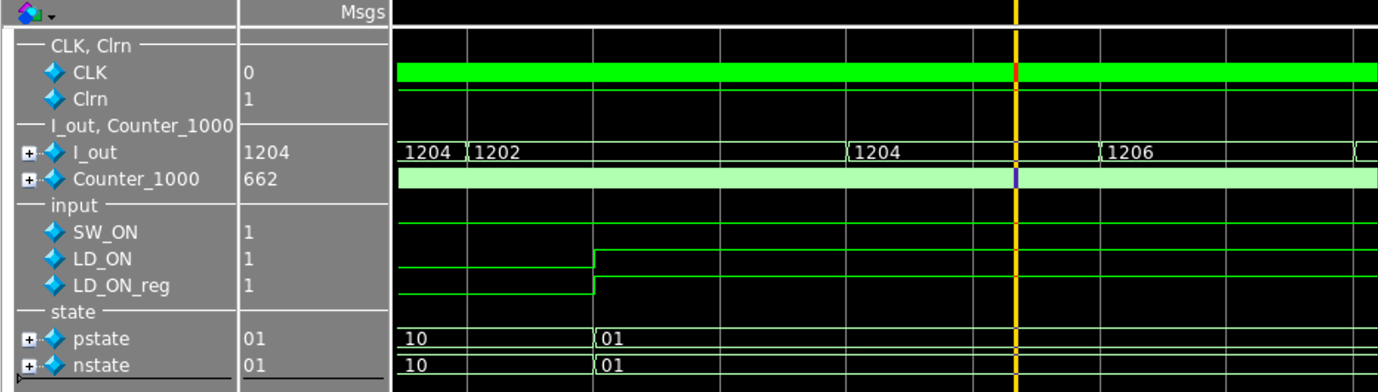
LD\_ON = 0

LD\_ON\_reg = 0

State : S2 (10)

I\_out decrease in 1 unit time block

4) Increasing I\_out again



SW\_ON = 1

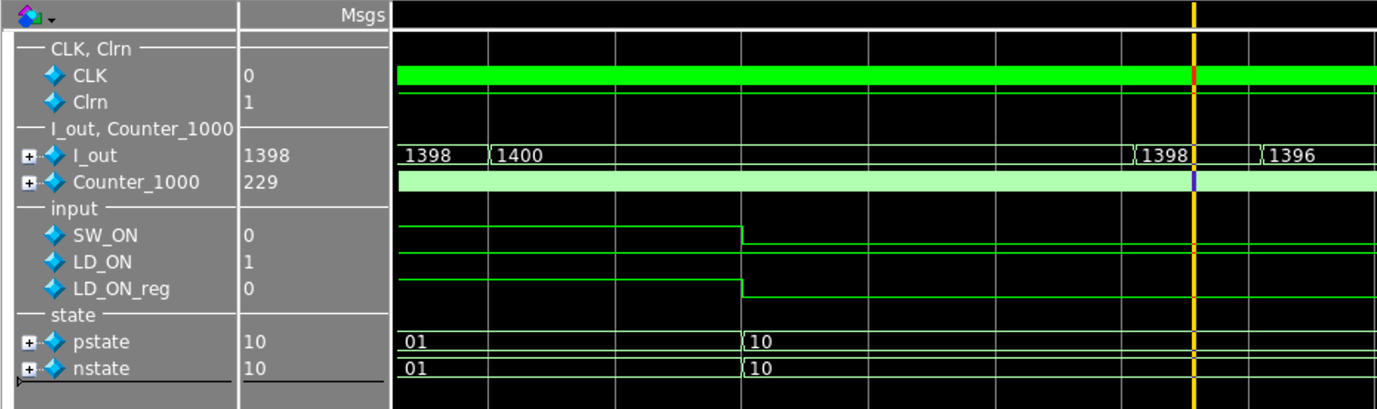
LD\_ON = 1

LD\_ON\_reg = 1

State : S1 (01)

I\_out increase in 2 unit time blocks again

5) Switching Off



SW\_ON = 0

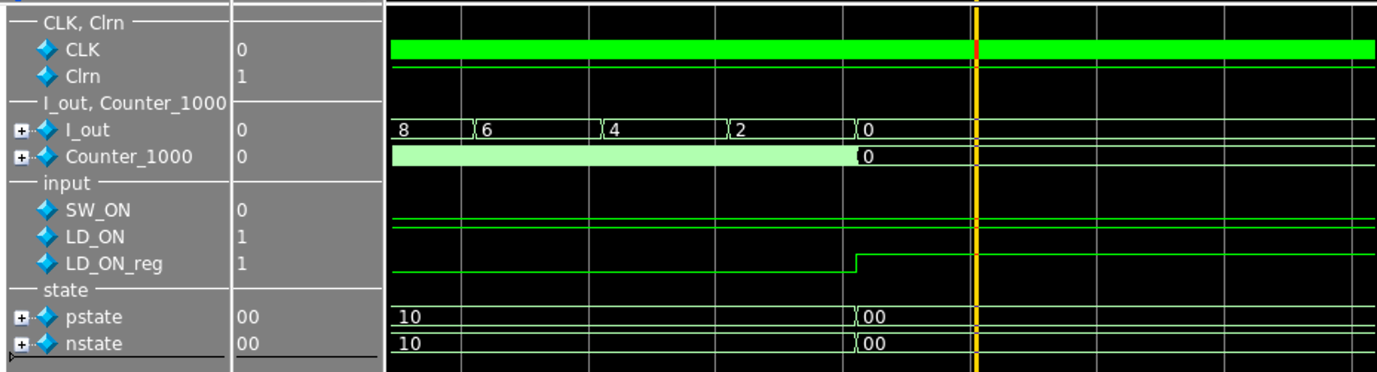
LD\_ON = 1

LD\_ON\_reg = 0

State : S2 (10) (decreasing I\_out first)

I\_out decrease in 1 unit time block

6) After end of decreasing, system is on initial state



SW\_ON = 0

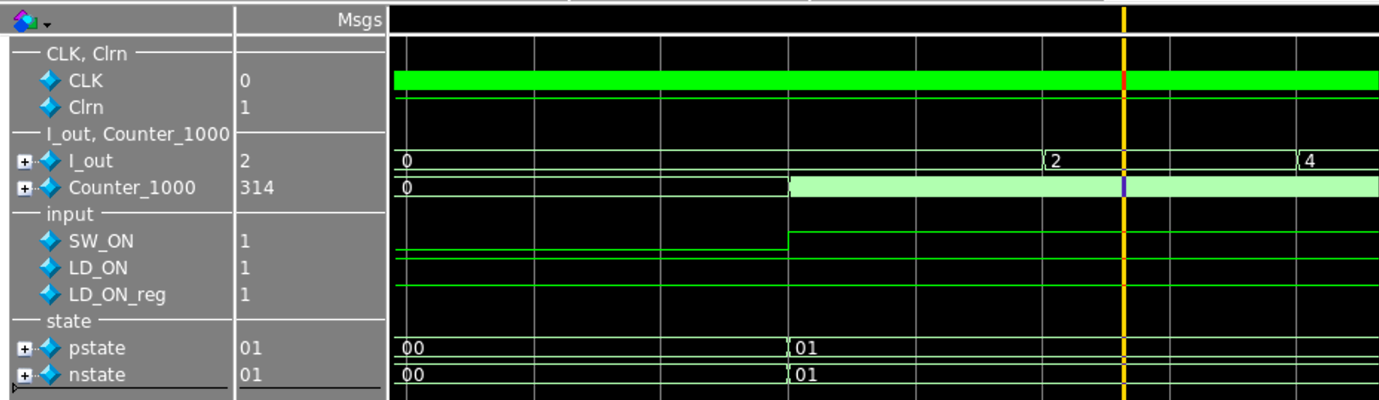
LD\_ON = 1

LD\_ON\_reg = 1

State : S0 (00) (initial state)

Stay on initial state

7) Restart



SW\_ON = 1

LD\_ON = 1

LD\_ON\_reg = 1

State : S1 (01)

I\_out increase in 2 unit time blocks