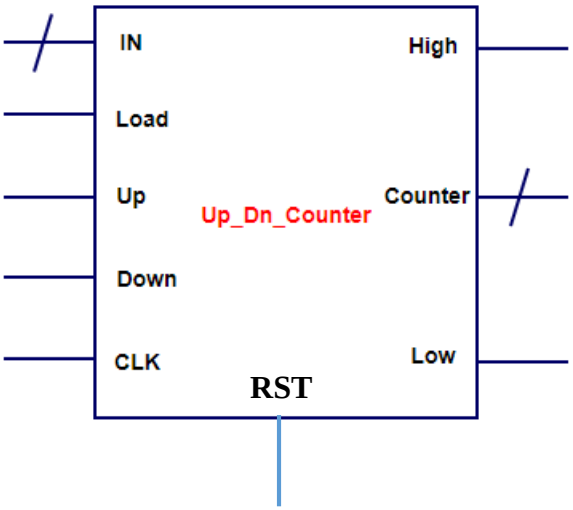
Design & Verfication For 4-bit UP / Down Counter



- 4-bit counter
- Reset Signal (Asynchronous active low)
- Count Value loaded from "IN" port on a positive clock edge when "Load" signal is high, Load has highest priority
- Count value incremented by 1 on a positive clock edge when "Up" signal is high.
- Count value decremented by 1 on a positive clock edge when "Down" signal is high, down has higher priority than "Up" signal
- After counter reaches 15 the next up makes overflow to zero
- After counter reaches 0 the next down makes overflow to 15
- "High" flag active high whenever count value is 15
- "Low" flag active high whenever count value is 0

Verification Plan

Testing items

1] RST → Asynchronous active low “All Output Signals Must be Zero”

2] 4 bit inputs → Valid or Invalid

3] Load → - @ Posedge CLK “ Highest Priority ”

Set to 1 → counter output signal to the input “IN” Signal

Set to 0 → input signal “IN” not loaded , go check Down Signal

4] Down → - @ Posedge CLK “ 2’nd Highest Priority ”

Set to 1 → counter output signal is decremented by 1

Set to 0 → counter output signal isn’t decremented ,go check UP signal

5] UP → - @ Posedge CLK " 3'rd Highest Priority "

Set to 1 → counter output signal is incremented by 1

Set to 0 → counter output signal isn't incremented.

6] counter → Checker: if reset asserted → Check Counter == 0

- @ Posedge CLK According to Priority

Checker: if load is high → check counter == IN

Checker: if Down is high → check counter is decremented

if counter == 0 → check overflow

Checker: if UP is high → check counter is incremented

if counter == 15 → check counter == 0

Checker: if UP & Down & Load are Low → check counter holds the last value

Checker: if counter loaded with invalid date → check counter remains invalid regardless of UP or Down

7] High → Checker: if counter == 15 → Check High == 1 else == 0

8] Low → Checker: if counter == 0 → Check Low == 1 else == 0

Functional Coverage

We need to cover all output signals:

counter → cover all values /* Question for Sherif */

→ Toggling only /* Question for Sherif */

counter\Down → counter overflow from 15 to zero

counter\Up → counter overflow from 0 to 15

High → 1 iff (counter == 15)

Low → 1 iff (counter == 0)

Load \ up \ down → cover priority

Load → cover loading using load signal only

Down → cover loading using Down signal only

Up → cover loading using Up signal only

Redundancy between scoreboard /* Question for Sherif */

/**

Question for Sherif do we need to cover the counter is incremented more than one consecutive time or one increment is enough and same for decrement and load

**/

Sequences & Constrained Randomization

1] Reset Sequence → with RST == 0

2] Priority Sequence → at least 2 control signals is one

2] Load Sequence → with Load == 1 , Down == 0 , Up == 0

3] Down Sequence → with Load == 0 , Down == 1 , Up == 0

4] Up Sequence → with Load == 0 , Down == 0 , Up == 1

Test Scenarios

Scenario 1 : Load Sequence → Up Sequence for random number transaction range [5:10]

-- Goal : testing increment functionality and overflow from 15 to 0

Scenario 2 : Load Sequence → Down Sequence for random number transaction range [5:10]

-- Goal : testing decrement functionality and overflow from 0 to 15

Scenario 3 : Load Sequence → Up → Down → Up → Load …etc.

-- Goal : Mixing the order of sequence

Scenario 4 : Priority Sequence → random for at least 10 times

-- Goal : testing the priority and idle functionality
