# **`CSE 620:**

# **Project: HDL Testbench**

**Student Name: Yehia ahmed Ibrahim shahin Student ID: 2301470**

* Include this header with your project.
* In doing this project, you could search the Internet, library, or any other source. ***Please do not copy material from your colleagues***.
* Total: **70 Marks**.

**Introduction:** In this project, you are required to do a set of HDL (VHDL or Verilog) mini-projects.

**Objective:** The objective of the assignment is to get the student experienced with HDL modeling and testbench creation.

**Scope:** Project scope includes developing HDL testbenches to a set of given non-trivial digital system designs.

**Statement:**

1. You are given:
   * The D-flip-flop testbench example, written in VHDL in the VHDL Overview module, included in the VHDL course that could be accessed on: https://sites.google.com/site/csevhdl.
   * An archive of the source code of all VHDL examples included in the VHDL course above. The archive is in the “VHDL Code Examples” folder on the “Course Material” page on the site mentioned above.
2. Select a number of the given example designs in the given archive and develop testbenches to them similar to the D-flip-flop testbench mentioned above.
3. You need to decide whether to do the mini-projects in VHDL or Verilog:
   * **VHDL option**: select 10 example designs.
   * **Verilog option**: select 5 example designs. You need to convert them to Verilog first, and then develop testbenches to them.

**Deadline:** Assignment starts from the first day of the term. The documentation submission and source code email are required before the starting time of the course final exam. No submission will be accepted after that time.

**Deliverables:** A documentation for all mini-projects and an archive of the source code of all developed testbenches. The source code archive is to be emailed to the course instructor. The documentation should include the following (at least) for each design:

1. Design example name and location in the slides,
2. If you are doing the project in Verilog, include and document the code of the design example,
3. Test strategy,
4. Testbench code,
5. Documentation for the testbench code,
6. Simulation results with comments, and
7. Names of all used tools.

**Notes:**

* The final documentation should be well written from the language and organization points of view. It must be precise and concise.
* Support the final documentation with neat diagrams and tables.
* Include a good list of references in the final documentation. Please cite every resource you use.
* To save you time, you do not need to reproduce figures/illustrations from resources. You could copy them as they are, if you wish. In that case, you must associate the copied material with a reference.
* Mini-projects are to be done individually. No groups are permitted.
* Do not copy testbenches and reports from your colleagues. Partially or fully copied testbenches or reports get zero credit.

***End of Assignment***

***Documentation of testbenches of examples***

***Ex1: Comparator (combinational circuit lecture : Slide (22)) :***

***- Test Strategy:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **Input** | | **Delay** | **Expected Output** | |
| **d\_in1** | **d\_in2** | **Q\_eq** | **Q\_neq** |
| Equality at zero | 00000000 | 00000000 | **15** ns | 1 | 0 |
| Equality at LSB | 11110000 | 10100000 | 15ns | 0 | 1 |
| Equality at MSB | 00001011 | 00000000 | 15ns | 0 | 1 |
| Equality at one | ***11111111*** | ***11111111*** | 15ns | 1 | 0 |
| Equality Flipped bits | ***10101010*** | ***10101010*** | 15ns | 1 | 0 |
| inequality Flipped bits | ***10101010*** | ***01010101*** | 15ns | 0 | 1 |

***- Testbench code:***

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**-- components that are used in testbench of comparator**

**PACKAGE** Tb\_comp **IS**

**COMPONENT** comparator **IS**

**Port(** a **:** **IN** std\_logic\_vector **(**7 **DOWNTO** 0**);**

b**:** **IN** std\_logic\_vector **(**7 **DOWNTO** 0**);**

equal\_out **:** **OUT** std\_logic**;**

not\_equal\_out**:** **OUT** std\_logic**);**

**END** **COMPONENT** comparator **;**

**END** **PACKAGE** Tb\_comp **;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** WORK**.**Tb\_comp**.ALL** **;**

**-- the test bench of comparator**

**ENTITY** Tb\_Comparator **IS**

**END** **ENTITY** Tb\_Comparator **;**

**-- the behaviour of the testbench**

**ARCHITECTURE** behav **OF** Tb\_Comparator **IS**

**-- define the prototype of comparator component**

**For** dut **:** comparator **USE** **ENTITY** WORK**.**comparator**(**Behavioral**);**

**-- define the input and output signals**

**SIGNAL** d\_in1 **:** STD\_LOGIC\_VECTOR **(**7 **DOWNTO** 0 **);**

**SIGNAL** d\_in2 **:** STD\_LOGIC\_VECTOR **(** 7 **DOWNTO** 0 **);**

**SIGNAL** d\_eq **:**STD\_LOGIC **;**

**SIGNAL** d\_neq **:** STD\_LOGIC **;**

**BEGIN**

dut **:** comparator **PORT** **MAP** **(** d\_in1 **,** d\_in2 **,** d\_eq **,** d\_neq**)** **;**

**-- testbench process**

P1 **:** **PROCESS** **IS**

**BEGIN**

-- Equality at zero

d\_in1 **<=** "00000000" **;**

d\_in2 **<=** "00000000" **;**

**WAIT** **FOR** 15 ns **;**

**ASSERT** d\_eq **=** '1' **and** d\_neq **=** '0'

**REPORT** "Problem with Equality at zero"

**SEVERITY** error **;**

**-- Equality at lsb**

d\_in1 **<=** "11110000" **;**

d\_in2 **<=** "10100000" **;**

**WAIT** **FOR** 15 ns **;**

**ASSERT** d\_eq **=** '0' **and** d\_neq **=** '1'

**REPORT** "Problem with equality at lsb"

**SEVERITY** error **;**

**-- Equality at msb**

d\_in1 **<=** "00001011" **;**

d\_in2 **<=** "00000000" **;**

**WAIT** **FOR** 15 ns **;**

**ASSERT** d\_eq **=** '0' **and** d\_neq **=** '1'

**REPORT** "Problem with equality at LSB"

**SEVERITY** error **;**

**--Equality at one**

d\_in1 **<=** "11111111" **;**

d\_in2 **<=** "11111111" **;**

**WAIT** **FOR** 15 ns **;**

**ASSERT** d\_eq **=** '1' **and** d\_neq **=** '0'

**REPORT** "Problem with equality ar one"

**SEVERITY** error**;**

**-- Equality Flipped bits**

d\_in1 **<=** "10101010" **;**

d\_in2 **<=** "10101010" **;**

**WAIT** **FOR** 15 ns **;**

**ASSERT** d\_eq **=** '1' **and** d\_neq **=** '0'

**REPORT** "Problem with Equality Flipped bits"

**SEVERITY** error**;**

**-- inequality Flipped bits**

d\_in1 **<=** "10101010" **;**

d\_in2 **<=** "01010101" **;**

**WAIT** **FOR** 15 ns **;**

**ASSERT** d\_eq **=** '0' **and** d\_neq **=** '1'

**REPORT** "Problem with inequality Flipped bits"

**SEVERITY** error**;**

**WAIT;**

**END** **PROCESS** P1 **;**

**END** **ARCHITECTURE** behav**;**

- Implement testbench module (Tb\_Comparator) and call the Tb\_comp component is rooted of comparator.

- using six test case to test the comparator:

- Equality at zero

- Equality at LSB

- Equality at MSB

- Equality at one

- Equality Flipped bits

- Inequality Flipped bits

***- Simulation result:***

A screenshot of a computer

Description automatically generated

**- comment:**

- The code of comparator component is executed successfully.

***Ex 2:( Sequential circuit lecture: Slide (19)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | Delay | D\_out  expected | Comment |
| Clk | D\_in | Res |
| Test the reset | 0->1 | 1 | 1 | 15ns | 0 | guarantee the effect of reset will not change with different input value |
| Test the reset | 0->1 | 0 | 1 | 15ns | 0 |
| odd parity | 0->1 | 1 | 0 | 15ns | 1 | Test odd Parity |
| Continue odd parity | 0->1 | 0 | 0 | 15ns | 1 | Test adding zero that will not change the output |
| even parity | 0->1 | 1 | 0 | 15ns | 0 | Test odd parity |
| Change to odd parity | 0->1 | 1 | 0 | 15ns | 1 | Test adding one that will change the output |
| Change to even parity | 0->1 | 1 | 0 | 15ns | 0 | Test adding one that will change the output |
| Continue odd parity | 0->1 | 0 | 0 | 15ns | 0 | Test adding zero that will not change the output |

***- Testbench code:***

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**PACKAGE** Tb\_oddP **IS**

**COMPONENT** tb\_fsm **IS**

**PORT(** clk**,** reset**:** **IN** std\_logic**;**

x**:** **IN** std\_logic**;**

y**:** **OUT** std\_logic**);**

**END** **COMPONENT** tb\_fsm **;**

**END** **PACKAGE** Tb\_oddP **;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**USE** WORK**.**Tb\_oddP**.ALL;**

**ENTITY** tb\_fsm\_com **IS**

**END** **ENTITY** tb\_fsm\_com **;**

**ARCHITECTURE** behav **OF** tb\_fsm\_com **IS**

**FOR** DUT **:** tb\_fsm **USE** **ENTITY** WORK**.**fsm **(**mealy\_3p**)** **;**

**SIGNAL** d\_in **,** clk **,** res **:** STD\_LOGIC **;**

**SIGNAL** d\_out **:** STD\_LOGIC **;**

**BEGIN**

DUT **:**tb\_fsm **PORT** **MAP** **(**clk**,**res**,**d\_in**,**d\_out**)** **;**

p1 **:** **PROCESS** **IS**

**BEGIN**

-- reset the clk = 0 ;

-- reset the design unit

res **<=** '1' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 10 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "Problem in testing the reset pin when d\_in = 1"

**SEVERITY** error **;**

-- guarantee the effect of reset will not change with different input value

res **<=** '1' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 22 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "Problem in testing the reset pin when d\_in = 0 "

**SEVERITY** error **;**

--Test odd Parity

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **until** **rising\_edge(**clk**)** **;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test odd Parity "

**SEVERITY** error**;**

-- Test adding zero that will not change the output

res**<=** '0' **;**

d\_in **<=** '0' **;**

**WAIT** **until** **rising\_edge(**clk**)** **;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test adding zero that will not change the output"

**SEVERITY** error **;**

-- Test odd parity

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **until** **rising\_edge(**clk**)** **;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem Test odd parity"

**SEVERITY** error **;**

-- Test adding one that will change the output

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **until** **rising\_edge(**clk**)** **;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test adding one that will change the output"

**SEVERITY** error **;**

-- Test adding one that will change the output

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **until** **rising\_edge(**clk**)** **;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem est adding one that will change the output"

**SEVERITY** error **;**

-- Test adding zero that will not change the output

res**<=** '0' **;**

d\_in **<=** '0' **;**

**WAIT** **until** **rising\_edge(**clk**)** **;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem Test adding zero that will not change the output "

**SEVERITY** error **;**

**WAIT** **;**

**END** **PROCESS** p1 **;**

clock**:** **PROCESS** **IS**

**BEGIN**

clk **<=**'0'**,** '1' **AFTER** 10 ns**;**

**WAIT** **FOR** 20 ns**;**

**END** **PROCESS** clock**;**

**END** **ARCHITECTURE** behav **;**

***- Simulation result:***

A screenshot of a computer

Description automatically generated

***-comment :***

- The code of odd parity (fsm\_mealy\_3p) component is executed successfully.

***Ex 3:( Sequential circuit lecture: Slide (24)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | Delay | D\_out  expected | Comment |
| Clk | D\_in | Res |
| Test the reset | 0->1 | 1 | 1 | 15ns | 0 | guarantee the effect of reset will not change with different input value |
| Test the reset | 0->1 | 0 | 1 | 15ns | 0 |
| odd parity | 0->1 | 1 | 0 | 15ns | 1 | Test odd Parity |
| Continue odd parity | 0->1 | 0 | 0 | 15ns | 1 | Test adding zero that will not change the output |
| even parity | 0->1 | 1 | 0 | 15ns | 0 | Test odd parity |
| Change to odd parity | 0->1 | 1 | 0 | 15ns | 1 | Test adding one that will change the output |
| Change to even parity | 0->1 | 1 | 0 | 15ns | 0 | Test adding one that will change the output |
| Continue odd parity | 0->1 | 0 | 0 | 15ns | 0 | Test adding zero that will not change the output |

***- Testbench code:***

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**PACKAGE** Tb\_moore\_pk **IS**

**COMPONENT** Tb\_moore **IS**

**PORT(** clk**,** reset**:** **IN** std\_logic**;**

x**:** **IN** std\_logic**;**

y**:** **OUT** std\_logic**);**

**END** **COMPONENT** Tb\_moore**;**

**END** **PACKAGE** Tb\_moore\_pk**;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**USE** WORK**.**Tb\_moore\_pk**.ALL;**

**ENTITY** Tb\_mo **IS**

**END** **ENTITY** Tb\_mo **;**

**ARCHITECTURE** behav **OF** Tb\_mo **IS**

**FOR** DUT **:** Tb\_moore **USE** **ENTITY** WORK**.**fsm **(**moore\_2p**);**

**SIGNAL** clk **,** res **,** d\_in **,** d\_out **:** STD\_LOGIC **;**

**BEGIN**

DUT **:** Tb\_moore **PORT** **MAP** **(**clk **,** res **,** d\_in **,** d\_out**);**

p1 **:** **PROCESS** **IS**

**BEGIN**

-- testing the reset signal

res **<=** '1' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 12 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "Problem in effect of reset signal ."

**SEVERITY** warning **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

res **<=** '1' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "Problem in effect of reset signal ."

**SEVERITY** warning **;**

-- testing the output

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

--Test odd Parity

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 12 ns**;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test odd Parity "

**SEVERITY** error**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding zero that will not change the output

res**<=** '0' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test adding zero that will not change the output"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test odd parity

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 12 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem Test odd parity"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding one that will change the output

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test adding one that will change the output"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding one that will change the output

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem est adding one that will change the output"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding zero that will not change the output

res**<=** '0' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem Test adding zero that will not change the output "

**SEVERITY** error **;**

**WAIT** **;**

**END** **PROCESS** p1 **;**

ck **:** **PROCESS** **IS**

**BEGIN**

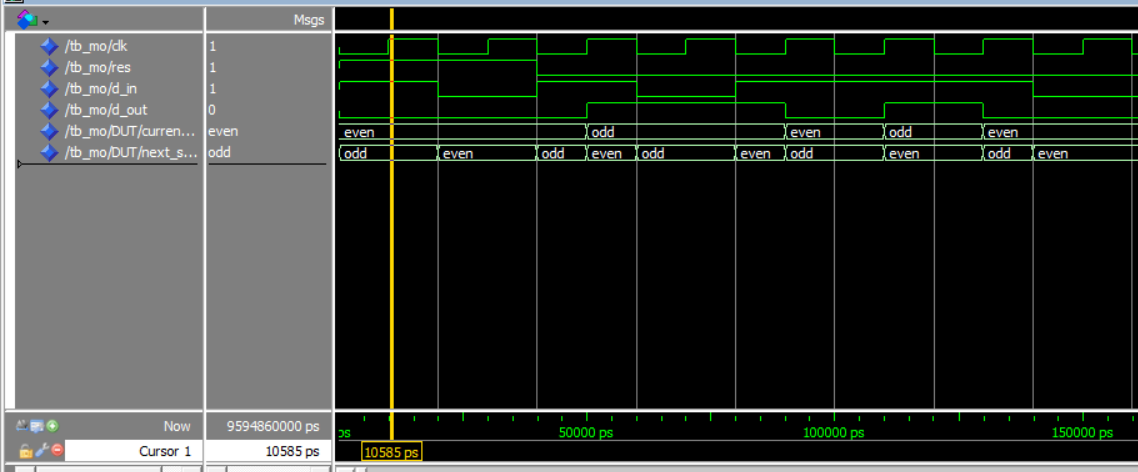
clk **<=** '0' **,** '1' **AFTER** 10 ns **;**

**WAIT** **FOR** 20 ns **;**

**END** **PROCESS** ck **;**

**END** **ARCHITECTURE** behav **;**

***- Simulation result:***



* **Comment:**
* The code of odd parity (fsm\_moore\_2p) component is executed successfully.

***Ex 4:( Sequential circuit lecture: Slide (26)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | Delay | D\_out  expected | Comment |
| Clk | D\_in | Res |
| Test the reset | 0->1 | 1 | 1 | 15ns | 0 | guarantee the effect of reset will not change with different input value |
| Test the reset | 0->1 | 0 | 1 | 15ns | 0 |
| odd parity | 0->1 | 1 | 0 | 15ns | 1 | Test odd Parity |
| Continue odd parity | 0->1 | 0 | 0 | 15ns | 1 | Test adding zero that will not change the output |
| even parity | 0->1 | 1 | 0 | 15ns | 0 | Test odd parity |
| Change to odd parity | 0->1 | 1 | 0 | 15ns | 1 | Test adding one that will change the output |
| Change to even parity | 0->1 | 1 | 0 | 15ns | 0 | Test adding one that will change the output |
| Continue odd parity | 0->1 | 0 | 0 | 15ns | 0 | Test adding zero that will not change the output |

***- Testbench code:***

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**PACKAGE** Tb\_moore\_pk **IS**

**COMPONENT** Tb\_moore3 **IS**

**PORT(** clk**,** reset**:** **IN** std\_logic**;**

x**:** **IN** std\_logic**;**

y**:** **OUT** std\_logic**);**

**END** **COMPONENT** Tb\_moore3**;**

**END** **PACKAGE** Tb\_moore\_pk**;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**USE** WORK**.**Tb\_moore\_pk**.ALL;**

**ENTITY** Tb\_mo **IS**

**END** **ENTITY** Tb\_mo **;**

**ARCHITECTURE** behav **OF** Tb\_mo **IS**

**FOR** DUT **:** Tb\_moore3 **USE** **ENTITY** WORK**.**fsm **(**moore\_3p**);**

**SIGNAL** clk **,** res **,** d\_in **,** d\_out **:** STD\_LOGIC **;**

**BEGIN**

DUT **:** Tb\_moore3 **PORT** **MAP** **(**clk **,** res **,** d\_in **,** d\_out**);**

p1 **:** **PROCESS** **IS**

**BEGIN**

-- testing the reset signal

res **<=** '1' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 12 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "Problem in effect of reset signal ."

**SEVERITY** warning **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

res **<=** '1' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "Problem in effect of reset signal ."

**SEVERITY** warning **;**

-- testing the output

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

--Test odd Parity

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 12 ns**;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test odd Parity "

**SEVERITY** error**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding zero that will not change the output

res**<=** '0' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test adding zero that will not change the output"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test odd parity

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 12 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem Test odd parity"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding one that will change the output

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '1'

**REPORT** "problem Test adding one that will change the output"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding one that will change the output

res**<=** '0' **;**

d\_in **<=** '1' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem est adding one that will change the output"

**SEVERITY** error **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

-- Test adding zero that will not change the output

res**<=** '0' **;**

d\_in **<=** '0' **;**

**WAIT** **FOR** 13 ns**;**

**ASSERT** d\_out **=** '0'

**REPORT** "problem Test adding zero that will not change the output "

**SEVERITY** error **;**

**WAIT** **;**

**END** **PROCESS** p1 **;**

ck **:** **PROCESS** **IS**

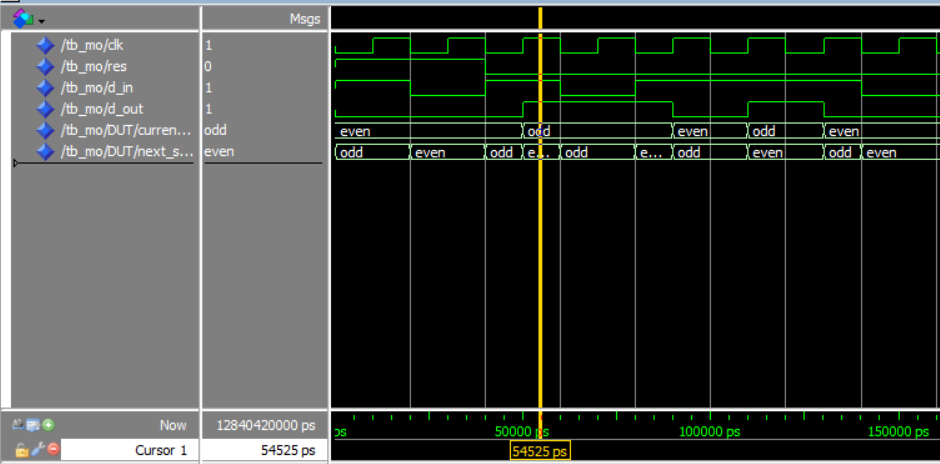
**BEGIN**

clk **<=** '0' **,** '1' **AFTER** 10 ns **;**

**WAIT** **FOR** 20 ns **;**

**END** **PROCESS** ck **;**

**END** **ARCHITECTURE** behav **;**

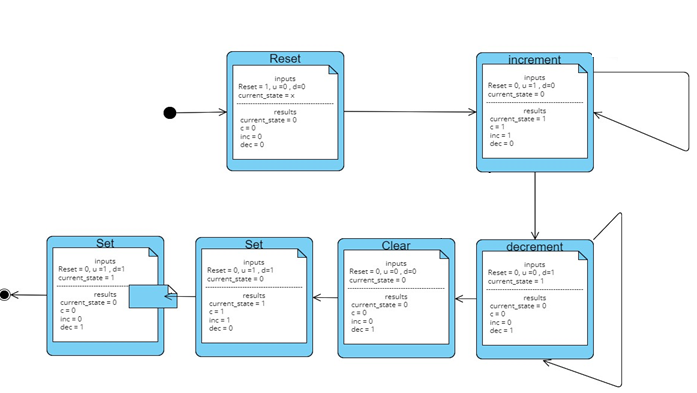
***- Simulation result:***

***-comment:***

- The code of odd parity (fsm\_moore\_3p) component is executed successfully.

***Ex 5:(RTL synthesis lecture: Slide (50)) :***

***- Test Strategy:***



***- Testbench code:***

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**PACKAGE** Tb\_PACK **IS**

**COMPONENT** tb\_com **IS**

**PORT(** u**,** d**,** reset**:** **IN** bit**;**

clk**:** **IN** std\_logic**;**

inc**,** dec**,** c**:** **OUT** bit**);**

**END** **COMPONENT** tb\_com **;**

**END** **PACKAGE** Tb\_PACK **;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** WORK**.**Tb\_PACK**.ALL** **;**

**ENTITY** tb\_inc **IS**

**END** **ENTITY** tb\_inc **;**

**ARCHITECTURE** behav **OF** tb\_inc **IS**

**FOR** DUT **:** tb\_com **USE** **ENTITY** WORK**.**source **(**fsm**)** **;**

**type** Tb\_state **IS** **(**RES\_S **,** INC\_S **,** DEC\_S **,** CLR\_S **,** SET\_S **,**FIN\_S **);**

**SIGNAL** u **,** d **,** reset **:** BIT **;**

**SIGNAL** clk **:** STD\_LOGIC **;**

**SIGNAL** inc **,** dec **,** c **:** BIT **;**

**SIGNAL** T\_st **:** Tb\_state **:=** RES\_S**;**

**BEGIN**

---------------------------------------DUT instance ---------------------------

DUT **:** tb\_com **PORT** **MAP** **(**u **,** d**,** reset **,** clk **,** inc **,** dec **,**c **);**

--------------------------------------------------------------------------------

-------------------------------------test strategy------------------------------

test **:** **PROCESS** **IS**

**VARIABLE** current\_state **:** natural **RANGE** 0 **TO** 1 **:=** 0 **;**

**VARIABLE** flag **:** natural **RANGE** 0 **TO** 1 **:=** 0 **;**

**BEGIN**

**CASE** T\_st **IS**

**WHEN** RES\_S **=>**

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

reset **<=** '1' **;**

u **<=** '0' **;**

d **<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** c **=** '0' **and** inc **=** '0' **and** dec **=**'0'

**REPORT** "Problem in reset implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 0 **;**

T\_st **<=** INC\_S **;**

**WAIT** **FOR** 2 ns **;**

**WHEN** INC\_S **=>**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

u**<=** '1' **;**

d**<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** c **=**'1' **and** inc **=** '0' **and** dec **=**'0'

**REPORT** "Problem in increment implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 1 **;**

T\_st **<=** DEC\_S **;**

**WAIT** **FOR** 2 ns **;**

**WHEN** DEC\_S **=>**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

u**<=** '0' **;**

d**<=** '1' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** c **=**'0' **and** inc **=** '0' **and** dec **=**'0'

**REPORT** "Problem in decrement implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 0 **;**

T\_st **<=** CLR\_S **;**

**WAIT** **FOR** 2 ns **;**

**WHEN** CLR\_S **=>**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

u**<=** '0' **;**

d**<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**IF** current\_state **=** 0 **THEN**

**ASSERT** c **=**'0' **and** inc **=** '0' **and** dec **=**'0'

**REPORT** "Problem in clear implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 0 **;**

**ELSE**

**ASSERT** c **=**'1' **and** inc **=** '1' **and** dec **=**'0'

**REPORT** "Problem in clear implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 0 **;**

**END** **IF** **;**

T\_st **<=** SET\_S **;**

**WAIT** **FOR** 2 ns **;**

**WHEN** SET\_S **=>**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

u**<=** '1' **;**

d**<=** '1' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**IF** current\_state **=** 0 **THEN**

**ASSERT** c **=**'1' **and** inc **=** '0' **and** dec **=**'1'

**REPORT** "Problem in set implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 1 **;**

**ELSE**

**ASSERT** c **=**'0' **and** inc **=** '1' **and** dec **=**'0'

**REPORT** "Problem in set implementation and effect on the output"

**SEVERITY** warning **;**

current\_state **:=** 0 **;**

**END** **IF** **;**

**IF** flag **=** 0 **THEN**

T\_st **<=** SET\_S **;**

flag **:=** 1 **;**

**ELSE**

T\_st **<=** FIN\_S **;**

**END** **IF** **;**

**WAIT** **FOR** 2 ns **;**

**WHEN** FIN\_S **=>**

**WAIT** **;**

**END** **CASE** **;**

**END** **PROCESS** test **;**

------------------------clock generation --------------------------------------

ck **:** **PROCESS** **IS**

**BEGIN**

clk **<=** '0' **,** '1' **AFTER** 10 ns**;**

**WAIT** **FOR** 20 ns **;**

**END** **PROCESS** ck **;**

**END** **ARCHITECTURE** behav**;**

***- Simulation result:***

A screenshot of a computer

Description automatically generated

**Comment:**

* Inc , dec are pulse output signal because change after the input change and after current state updated.
* The code of inc,dec component is executed successfully.
* ***Ex 6:(johnson sequential lecture: Slide (33)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | | Delay | Ang | Comment |
| Clk | Cw | ccw | Res |
| Test the reset | 0->1 | 1 | 0 | 1 | 15ns | 0000 | guarantee the effect of reset will not change with different input value |
| Test the reset | 0->1 | 0 | 1 | 1 | 15ns | 0000 |
| Cw | 0->1 | 1 | 0 | 0 | 15ns | 0001 | Test clockwise |
| Staying | 0->1 | 0 | 0 | 0 | 15ns | 0001 | Staying in the same state |
| ccw | 0->1 | 0 | 1 | 0 | 15ns | 0000 | Test counterclockwise from staying state |
| Ccw , cw | 0->1 | 1 | 1 | 0 | 15ns | 1000 | Conflict state, test staying in the previous state |
| ccw | 0->1 | 0 | 1 | 0 | 15ns | 1000 | Test counterclockwise from staying state |
| cw | 0->1 | 1 | 0 | 0 | 15ns | 0000 | Test clockwise |

***- Testbench code:***

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**PACKAGE** Tb\_pk **IS**

**COMPONENT** Tb\_com **IS**

**PORT(** cw**,** ccw**,** reset**,** clk**:** **IN** std\_logic**;**

angle **:** **OUT** std\_logic\_vector **(**2 **DOWNTO** 0**));**

**END** **COMPONENT** Tb\_com **;**

**END** **PACKAGE** Tb\_pk **;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** work**.**const\_johnson\_encoding**.ALL;**

**USE** WORK**.**Tb\_pk**.ALL** **;**

**ENTITY** tb\_jo **IS**

**END** **ENTITY** tb\_jo **;**

**ARCHITECTURE** behav **OF** tb\_jo **IS**

**FOR** DUT **:** Tb\_com **USE** **ENTITY** WORK**.**angle **(**angle**);**

**SIGNAL** cw**,** ccw**,** reset**,** clk **:** std\_logic**;**

**SIGNAL** angle **:** std\_logic\_vector **(**2 **DOWNTO** 0**);**

**BEGIN**

DUT **:** Tb\_com **PORT** **MAP** **(**cw**,** ccw**,** reset**,** clk **,**angle **);**

test **:** **PROCESS** **IS**

**BEGIN**

--------------------------------------------------------------------------------------

----------------------------------------rst =1 -------------------------------------

reset **<=** '1' **;**

cw **<=** '0' **;**

ccw **<=** '1' **;WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "0000"

**REPORT** "Problem in effect the reset on outputs."

**SEVERITY** warning **;**

------------------------------------------------------------------------------------

------------------------------rst =1 -----------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '1'**;**

cw **<=** '0' **;**

ccw **<=** '1' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "0000"

**REPORT** "Problem in effect the reset on the outputs"

**SEVERITY** warning **;**

-------------------------------------------------------------------------------------

-----------------------------------cw = 1 , ccw =0 ----------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

reset **<=** '0' **;**

cw **<=** '1' **;**

ccw **<=** '0'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "0001"

**REPORT** "problem in clock wise transition "

**SEVERITY** warning **;**

------------------------------------------------------------------------------------

--------------------------------cw = 0 , ccw = 0 -----------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

cw **<=** '0' **;**

ccw **<=** '0'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "0001"

**REPORT** "problem in staying the state"

**SEVERITY** warning **;**

-------------------------------------------------------------------------------------

---------------------------------- cw = 0 , ccw = 1 ---------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

cw **<=** '0' **;**

ccw **<=** '1'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "0000"

**REPORT** "problem in counterclockwise the state"

**SEVERITY** warning **;**

-------------------------------------------------------------------------------------

------------------------------ cw = 1 , ccw = 1 -------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

cw **<=** '1' **;**

ccw **<=** '1'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "0000"

**REPORT** "problem in staying state in confilct cw , ccw "

**SEVERITY** warning **;**

---------------------------------------------------------------------------------------

------------------------------ cw = 0 , ccw = 1 ---------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

cw **<=** '0' **;**

ccw **<=** '1'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "1000"

**REPORT** "problem in counterclockwise the state"

**SEVERITY** warning **;**

---------------------------------------------------------------------------------------

------------------------------cw = 1 , ccw = 0 ----------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

reset **<=** '0' **;**

cw **<=** '1' **;**

ccw **<=** '0'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 2 ns **;**

**ASSERT** angle **=** "1000"

**REPORT** "problem in clockwise the state"

**SEVERITY** warning **;**

---------------------------------------------------------------------------------------

**WAIT** **;**

**END** **PROCESS** test **;**

ck **:** **PROCESS** **IS**

**BEGIN**

clk **<=** '0' **,** '1' **AFTER** 10 ns **;**

**WAIT** **FOR** 20 ns**;**

**END** **PROCESS** ck **;**

**END** **ARCHITECTURE** behav **;**

* A screen shot of a computer

  Description automatically generated**Simulation:**

**Comment :**

* Bug : there are 8 state that should represent in 4-bit but the design code has 3-bit angle not 4-bit.
* ***Ex 7:(shift register sequential lecture: Slide (33)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | | | | Q | Comment |
| Clk | Lin | Rin | s | d | Res |
| Test the reset | 0->1 | X | X | X | X | 1 | 0000 | guarantee the effect of reset will not change with different input value |
| Load | 0->1 | X | X | 11 | d | 0 | d | Load the data |
| Shift left (logic) | 0->1 | X | 0 | 01 | XXXX | 0 | D[2:0]&0 | Shift left (logic) |
| Shift left (arth) | 0->1 | X | 1 | 01 | XXXX | 0 | D[2:0]&1 | Shift left (arth) |
| Shift right (arth) | 0->1 | 1 | X | 10 | XXXX | 0 | D[2:0]&1 | Shift right (arth) |
| Shift left (logic) | 0->1 | 0 | X | 10 | XXXX | 0 | D[2:0]&0 | Shift right (logic) |
| Hold | 0->1 | X | X | 00 | XXXX | 0 | q | Hold the data |

***- Testbench code:***

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**PACKAGE** Tb\_Pk **IS**

**COMPONENT** Tb\_com **IS**

**GENERIC(** width**:** positive **:=** 4**);**

**PORT(** clk**,** clr**,** l\_in**,** r\_in**,** s0**,** s1**:** **IN** std\_logic**;**

d**:** **IN** std\_logic\_vector **(**width**-**1 **DOWNTO** 0**);**

q**:** **INOUT** std\_logic\_vector **(**width**-**1 **DOWNTO** 0**));**

**END** **COMPONENT** Tb\_com **;**

**END** **PACKAGE** Tb\_pk **;**

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** WORK**.**Tb\_pk**.ALL** **;**

**ENTITY** Tb\_sh **IS**

**END** **ENTITY** Tb\_sh **;**

**ARCHITECTURE** behav **OF** Tb\_sh **IS**

**FOR** DUT **:** Tb\_com **USE** **ENTITY** WORK**.**sr **(**sr**)** **;**

**CONSTANT** Width **:** INTEGER **:=** 4 **;**

**SIGNAL** clk**,** clr**,** l\_in**,** r\_in**,** s0**,** s1 **:**std\_logic**;**

**SIGNAL** d **:** std\_logic\_vector **(**width**-**1 **DOWNTO** 0**);**

**SIGNAL** q**:** std\_logic\_vector **(**width**-**1 **DOWNTO** 0**);**

**BEGIN**

DUT **:** Tb\_com **GENERIC** **MAP** **(**width **=>** 4**)** **PORT** **MAP** **(** clk**,** clr**,** l\_in**,** r\_in**,** s0**,** s1 **,** d **,**q**);**

test **:** **PROCESS** **IS**

**BEGIN**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

clr **<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "0000"

**REPORT** "Problem in effect of reset."

**SEVERITY** warning **;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

clr **<=** '1' **;**

d **<=** "1010" **;**

s0 **<=** '1' **;**

s1 **<=** '1' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "1010"

**REPORT** "problem in load option "

**SEVERITY** warning**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

s0 **<=** '1' **;**

s1 **<=** '0' **;**

r\_in **<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "0100"

**REPORT** "problem in shift left logic option "

**SEVERITY** warning**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

s0 **<=** '1' **;**

s1 **<=** '0' **;**

r\_in **<=** '1' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "1001"

**REPORT** "problem in shift left arth option "

**SEVERITY** warning**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

s0 **<=** '0' **;**

s1 **<=** '1' **;**

l\_in **<=** '1' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "1100"

**REPORT** "problem in shift right arth option "

**SEVERITY** warning**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

s0 **<=** '0' **;**

s1 **<=** '1' **;**

l\_in **<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "0110"

**REPORT** "problem in shift right logic option "

**SEVERITY** warning**;**

**WAIT** **UNTIL** **falling\_edge** **(**clk**);**

s0 **<=** '0' **;**

s1 **<=** '0' **;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**);**

**WAIT** **FOR** 3 ns **;**

**ASSERT** q **=** "0110"

**REPORT** "problem in shift right logic option "

**SEVERITY** warning**;**

**WAIT** **;**

**END** **PROCESS** test **;**

ck **:** **PROCESS** **IS**

**BEGIN**

clk **<=** '0' **,** '1' **AFTER** 10 ns **;**

**WAIT** **FOR** 20 ns **;**

**END** **PROCESS** ck **;**

**END** **ARCHITECTURE** behav **;**

* ***Simulation:***

A screen shot of a computer

Description automatically generated

* **Comment:**
* The code of shift register component is executed successfully.

* ***Ex 8:(ROM sequential lecture: Slide (53)) :***

***- Test Strategy:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Inputs | | data | Comment |
| en | addr |
| Disable | 0 | XXXX | ZZZZZZ | Test the disable operation |
| Apply address | 1 | 000 | 000000 | Read the data that has this address |
| Apply address | 1 | 101 | 011001 | Read the data that has this address |
| Apply address | 1 | 001 | 000001 | Read the data that has this address |
| Apply address | 1 | 010 | 000100 | Read the data that has this address |
| Apply address | 1 | 100 | 010000 | Read the data that has this address |
| Apply address | 1 | 111 | 110001 | Read the data that has this address |

***- Testbench code:***

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** IEEE**.**numeric\_STD**.ALL;**

**PACKAGE** Tb\_pk **IS**

**COMPONENT** Tb\_com **IS**

**GENERIC(** n **:** integer **:=** 3**;**

m **:** integer **:=** 6**);**

**PORT(** enable**:** **IN** std\_logic**;**

address**:** **IN** unsigned **(**n**-**1 **DOWNTO** 0**);**

data**:** **OUT** unsigned **(**m**-**1 **DOWNTO** 0**));**

**END** **COMPONENT** Tb\_com **;**

**END** **PACKAGE** Tb\_pk **;**

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** IEEE**.**numeric\_STD**.ALL;**

**USE** WORK**.**Tb\_pk**.ALL;**

**ENTITY** Tb\_rom **IS**

**END** **ENTITY** Tb\_rom **;**

**ARCHITECTURE** behav **OF** Tb\_rom **IS**

**CONSTANT** n **:** INTEGER **:=** 3 **;**

**CONSTANT** m **:**INTEGER **:=** 6 **;**

**SIGNAL** addr**:** unsigned **(**n**-**1 **DOWNTO** 0**);**

**SIGNAL** data**:** unsigned **(**m**-**1 **DOWNTO** 0**);**

**SIGNAL** en **:** STD\_LOGIC **;**

**SIGNAL** dt **:** STD\_LOGIC\_VECTOR **(**m**-**1 **DOWNTO** 0**)** **;**

**FOR** DUT **:** Tb\_com **USE** **ENTITY** WORK**.**rom **(**flag\_arch**)** **;**

**BEGIN**

DUT **:** Tb\_com **GENERIC** **MAP** **(** n **=>** 3 **,** m **=>**6**)** **PORT** **MAP** **(**en **,** addr **,** data**);**

test **:** **PROCESS** **IS**

**BEGIN**

**WAIT** **FOR** 2 ns **;**

en **<=** '0' **;**

dt **<=** STD\_LOGIC\_VECTOR **(**data**);**

**WAIT** **FOR** 4 ns **;**

**ASSERT** dt **=** "ZZZZZZ"**;**

**REPORT** "problem in output when ROM is disabled"

**SEVERITY** warning **;**

**WAIT** **FOR** 4 ns **;**

en **<=** '1' **;**

addr **<=** "000" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** data **=** "000000"

**REPORT** "problem in address : 000 "

**SEVERITY** warning **;**

**WAIT** **FOR** 2 ns **;**

en **<=** '1' **;**

addr **<=** "101" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** data **=** "011001"

**REPORT** "problem in address : 101 "

**SEVERITY** warning **;**

**WAIT** **FOR** 2 ns **;**

en **<=** '1' **;**

addr **<=** "001" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** data **=** "000001"

**REPORT** "problem in address : 001 "

**SEVERITY** warning **;**

**WAIT** **FOR** 2 ns **;**

en **<=** '1' **;**

addr **<=** "010" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** data **=** "000100"

**REPORT** "problem in address : 010 "

**SEVERITY** warning **;**

**WAIT** **FOR** 2 ns **;**

en **<=** '1' **;**

addr **<=** "100"**;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** data **=** "010000"

**REPORT** "problem in address : 100 "

**SEVERITY** warning **;**

**WAIT** **FOR** 2 ns **;**

en **<=** '1' **;**

addr **<=** "111" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** data **=** "110001"

**REPORT** "problem in address : 111 "

**SEVERITY** warning **;**

**WAIT** **;**

**END** **PROCESS** test **;**

**END** **ARCHITECTURE** behav **;**

* A screen shot of a computer

  Description automatically generated**-Simulation:**
* **Comment :**
* The code of ROM component is executed successfully.
* ***Ex 9:(RAM sequential lecture: Slide (62)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | | | Data\_out | Comment |
| R | W | Addr\_in | Addr\_out | Data\_in |
| Write | 0 | 1 | 0000 | XXXX | 1010 | XXXX | Test the write operation on ram |
| Write | 0 | 1 | 1111 | XXXX | 0011 | XXXX | Test the write operation on ram |
| Read | 1 | 0 | XXXX | 0000 | XXXX | 1010 | Test the read the data that were written in previous case |
| Read | 1 | 0 | XXXX | 1111 | XXXX | 0011 | Test the read the data that were written in previous case |
| Write and read | 1 | 1 | 0011 | 0000 | 1100 | 1010 | Test write and read operation together |
| Read | 1 | 0 | XXXX | 0011 | XXXX | 1100 | Test read data that was written in previous case |

***Testbench code:***

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** IEEE**.**NUMERIC\_STD**.ALL;**

**PACKAGE** Tb\_pk **IS**

**COMPONENT** Tb\_com **IS**

**GENERIC(** n**:** positive **:=** 4**;**

m**:** positive **:=** 4**);**

**PORT(** r**,** w**:** **IN** std\_logic**;**

address\_in**:** **IN** unsigned **(**n**-**1 **DOWNTO** 0**);**

address\_out**:** **IN** unsigned **(**n**-**1 **DOWNTO** 0**);**

data\_in**:** **IN** std\_logic\_vector **(**m**-**1 **DOWNTO** 0**);**

data\_out**:** **OUT** std\_logic\_vector **(**m**-**1 **DOWNTO** 0**));**

**END** **COMPONENT** Tb\_com **;**

**END** **PACKAGE** Tb\_pk **;**

**LIBRARY** IEEE**;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** IEEE**.**NUMERIC\_STD**.ALL;**

**USE** WORK**.**Tb\_pk**.ALL** **;**

**ENTITY** Tb\_ram **IS**

**END** **ENTITY** Tb\_ram **;**

**ARCHITECTURE** Tb\_behav **OF** Tb\_ram **IS**

**CONSTANT** n **:** positive **:=** 4 **;**

**CONSTANT** m **:** positive **:=** 4 **;**

**FOR** DUT **:** Tb\_com **USE** **ENTITY** WORK**.**dual\_port\_ram **(**behav**);**

**SIGNAL** r**,** w**:**std\_logic**;**

**SIGNAL** addr\_in **,** addr\_out**:**unsigned **(**n**-**1 **DOWNTO** 0**);**

**SIGNAL** da\_in **,** da\_out **:** std\_logic\_vector **(**m**-**1 **DOWNTO** 0**);**

**BEGIN**

DUT **:** Tb\_com **GENERIC** **MAP** **(**n **=>**4 **,** m**=>**4**)** **PORT** **MAP** **(**r**,**w**,**addr\_in **,** addr\_out **,** da\_in **,** da\_out**)** **;**

test **:** **PROCESS** **IS**

**BEGIN**

----------------------------------write case---------------------------------------------

**WAIT** **FOR** 2 ns **;**

w **<=** '1' **;**

r **<=** '0' **;**

addr\_in **<=** "0000" **;**

da\_in **<=** "1010" **;**

**WAIT** **FOR** 2 ns **;**

----------------------------------write case ---------------------------------------------

**WAIT** **FOR** 2 ns **;**

w **<=** '1' **;**

r **<=** '0' **;**

addr\_in **<=** "1111" **;**

da\_in **<=** "0011" **;**

**WAIT** **FOR** 2 ns **;**

----------------------------------read case ---------------------------------------------

**WAIT** **FOR** 2 ns **;**

w **<=** '0' **;**

r **<=** '1' **;**

addr\_out **<=** "0000" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** da\_out **=** "1010"

**REPORT** "problem in writing or read the data in address : 0000"

**SEVERITY** warning **;**

----------------------------------read case ---------------------------------------------

**WAIT** **FOR** 2 ns **;**

w **<=** '0' **;**

r **<=** '1' **;**

addr\_out **<=** "1111" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** da\_out **=** "0011"

**REPORT** "problem in writing or read the data in address : 1111"

**SEVERITY** warning **;**

----------------------------------write and read case s ---------------------------------------------

**WAIT** **FOR** 2 ns **;**

w **<=** '1' **;**

r **<=** '1' **;**

addr\_in **<=** "0011" **;**

addr\_out **<=** "0000" **;**

da\_in **<=** "1100" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** da\_out **=** "1010"

**REPORT** "problem in writing or reading the data in address 0000"

**SEVERITY** warning **;**

----------------------------------read case ---------------------------------------------

**WAIT** **FOR** 2 ns **;**

w **<=** '0' **;**

r **<=** '1' **;**

addr\_out **<=** "0011" **;**

**WAIT** **FOR** 2 ns **;**

**ASSERT** da\_out **=** "1100"

**REPORT** "problem in writing or read the data in address : 0011"

**SEVERITY** warning **;**

**WAIT** **;**

**END** **PROCESS** test **;**

**END** **ARCHITECTURE** Tb\_behav **;**

* A screen shot of a computer

  Description automatically generated**Simulation :**
* **Comment :**

The code of RAM component is executed successfully.

* ***Ex 10:(Binary counter sequential lecture: Slide (38)) :***

***- Test Strategy:***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | Inputs | | | | | dout | Comment |
| Clk | Clr | Up\_do | Preset | din |
| Test the reset | 0->1 | 1 | X | X | XXXX | 0000 | Test the effect of reset on outputs |
| Load | 0->1 | 0 | X | 1 | 1010 | 1010 | Test the load the input to output |
| Count up two cycle | 0->1 | 0 | 1 | 0 | XXXX | 1011  1100 | Test the output will be incremented within more clock cycle |
| 0->1 |
| Count down two cycle | 0->1  0->1  0->1 | 0 | 0 | 0 | XXXX | 1011  1010  1001 | Test the output will be decremented within more clock cycle |
| Count up to exceed the max | 0->1 | 0 | 1 | 0 | XXXX | 0000 | test what happened when exceed the maximum number |
| Count down to exceed the min | 0->1 | 0 | 0 | 1 | XXXX | 0000 | Test what happened when exceed the minimum number |

***- Testbench code:***

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL** **;**

**USE** IEEE**.**NUMERIC\_STD**.ALL** **;**

**PACKAGE** Tb\_pk **IS**

**COMPONENT** Tb\_com **IS**

**GENERIC(** n**:** positive **:=** 4**);**

**PORT(** clk**,** clear**,** up\_down**:** **IN** std\_logic**;**

preset**:** **IN** std\_logic**;**

d\_in**:** **IN** unsigned **(**n**-**1 **DOWNTO** 0**);**

d\_out**:** **OUT** unsigned **(**n**-**1 **DOWNTO** 0**));**

**END** **COMPONENT** Tb\_com **;**

**END** **PACKAGE** Tb\_pk **;**

**LIBRARY** IEEE **;**

**USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**USE** IEEE**.**NUMERIC\_STD**.ALL;**

**USE** WORK**.** Tb\_pk**.ALL** **;**

**ENTITY** Tb\_co **IS**

**END** **ENTITY** Tb\_co **;**

**ARCHITECTURE** behav **OF** Tb\_co **IS**

**CONSTANT** n **:** positive **:=**4 **;**

**FOR** DUT **:** Tb\_com **USE** **ENTITY** WORK**.**bin\_counter **(**behav**);**

**SIGNAL** clk**,** clear**,** up\_down **,** preset **:** std\_logic**;**

**SIGNAL** d\_in **,** d\_out **:** unsigned **(**n**-**1 **DOWNTO** 0**)** **;**

**BEGIN**

DUT **:** Tb\_com **GENERIC** **MAP** **(**n**=>**4**)** **PORT** **MAP** **(**clk**,** clear**,** up\_down **,** preset **,** d\_in **,** d\_out**);**

test **:** **PROCESS** **IS**

**VARIABLE** c **:** positive **:=**2 **;**

**BEGIN**

-------------------------------------------------- reset -------------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

clear **<=** '1'**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**WAIT** **FOR** 4 ns **;**

**ASSERT** d\_out **=** "0000"

**REPORT** "problem in reset effect on the output"

**SEVERITY** warning **;**

-------------------------------------------------- preset -------------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

clear **<=** '0' **;**

preset **<=** '1'**;**

d\_in **<=** "1010"**;**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**WAIT** **FOR** 4 ns **;**

**ASSERT** d\_out **=** "1010"

**REPORT** "problem in preset option ."

**SEVERITY** warning **;**

-------------------------------------------------- count up (2) cycle -------------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

clear **<=** '0' **;**

preset **<=** '0'**;**

up\_down **<=** '1' **;**

**FOR** j **IN** c**-**1 **DOWNTO** 0 **LOOP**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**END** **LOOP** **;**

**WAIT** **FOR** 4 ns **;**

**ASSERT** d\_out **=** "1100"

**REPORT** "problem in count up (2) cycle option ."

**SEVERITY** warning **;**

-------------------------------------------------- count down (3) -------------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

clear **<=** '0' **;**

preset **<=** '0'**;**

up\_down **<=** '0' **;**

**FOR** j **IN** c **DOWNTO** 0 **LOOP**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**END** **LOOP** **;**

**WAIT** **FOR** 4 ns **;**

**ASSERT** d\_out **=** "1001"

**REPORT** "problem in count down (3) cycle option ."

**SEVERITY** warning **;**

-------------------------------------------------- count down to min -------------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

clear **<=** '0' **;**

preset **<=** '0'**;**

up\_down **<=** '0' **;**

**FOR** j **IN** **to\_integer(**d\_out**)** **DOWNTO** 0 **LOOP**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**END** **LOOP** **;**

**WAIT** **FOR** 4 ns **;**

**ASSERT** d\_out **=** "1111"

**REPORT** "problem in count down to min option ."

**SEVERITY** warning **;**

-------------------------------------------------- count up to max -------------------------------------------------

**WAIT** **UNTIL** **falling\_edge** **(**clk**)** **;**

clear **<=** '0' **;**

preset **<=** '0'**;**

up\_down **<=** '0' **;**

**FOR** j **IN** 0 **TO** **to\_integer(**d\_out**)-**1 **LOOP**

**WAIT** **UNTIL** **rising\_edge** **(**clk**)** **;**

**END** **LOOP** **;**

**WAIT** **FOR** 4 ns **;**

**ASSERT** d\_out **=** "0000"

**REPORT** "problem in count up to max option ."

**SEVERITY** warning **;**

**WAIT** **;**

**END** **PROCESS** test **;**

ck **:** **PROCESS** **IS**

**BEGIN**

clk **<=** '0' **,** '1' **AFTER** 10 ns **;**

**WAIT** **FOR** 20 ns **;**

**END** **PROCESS** ck **;**

**END** **ARCHITECTURE** behav **;**

* A screen shot of a computer

  Description automatically generated**Simulation :**
* **Comment :**

The code of binary counter component is executed successfully.

**Tools:**

* **Modelism**