**MAIN VHDL CODE**

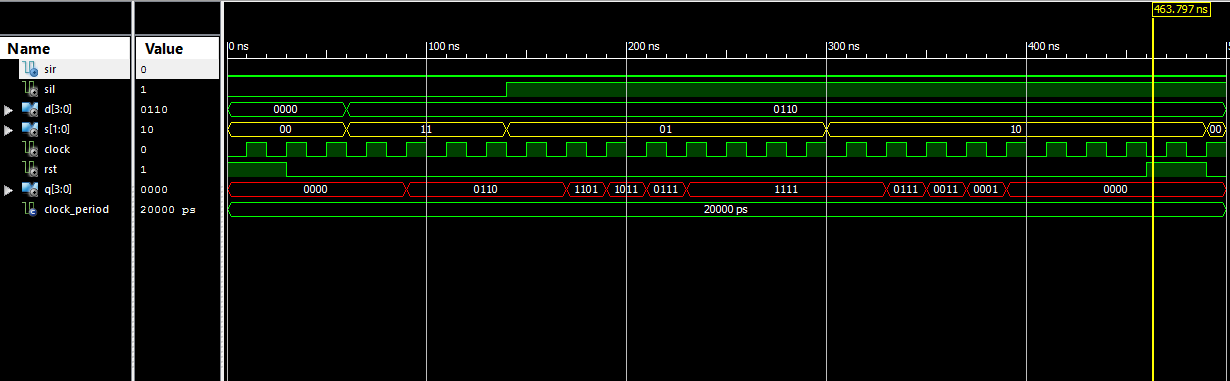
|  |
| --- |
| **library** IEEE**;**  **use** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **entity** Uni\_Shift\_Reg **is**  **port** **(** SIR **:** **in** std\_logic**;** --This is the Serial Input used when Right Shifting  SIL **:** **in** std\_logic**;** --This is the Serial Input used when Left Shifting  D **:** **in** std\_logic\_vector **(**3 **downto** 0**);** --There are Parallel loading inputs (D(0)D(1)D(2) and D(3)  S **:** **in** std\_logic\_vector **(**1 **downto** 0**);** --These are operation selecttion inputs (S(1) and S(0))  Q **:** **out** std\_logic\_vector **(**3 **downto** 0**);**--These are outputs of the register (Q(0)Q(1)Q(2) and Q(3)  clock **:** **in** std\_logic**;** --This is clock input  RST **:** **in** std\_logic --This is reset input  **);**  **end** Uni\_Shift\_Reg**;**  **architecture** Behavioral **of** Uni\_Shift\_Reg **is**  **signal** t**:** std\_logic\_vector **(**3 **downto** 0**);** --This is the temporary signal used inside the Process  **begin**    **process** **(**SIR**,**SIL**,**D**,**S**,**clock**,**RST**)**    **begin**    **if** RST**=**'1' **then**  t**<=**"0000"**;** --Initialize temparory signal to 0000  Q**<=**"0000"**;** --Initialize output to 0000    **elsif** **(**clock**=**'1' **and** clock' **event)** **then**    **case** S **is**  --Parallel Loading  **when** "11" **=>**  t**(**0**)<=**D**(**0**);** -- When S at 11, parallel  t**(**1**)<=**D**(**1**);** -- input will load in to  t**(**2**)<=**D**(**2**);** -- the temporary signal  t**(**3**)<=**D**(**3**);** --  -- and then  Q**(**0**)<=**t**(**0**);** -- temporary signal will  Q**(**1**)<=**t**(**1**);** -- load in to the output  Q**(**2**)<=**t**(**2**);** --  Q**(**3**)<=**t**(**3**);** --    --Shift Left  **when** "01" **=>**  t**(**0**)<=**D**(**0**);** -- When S at 01, parallel  t**(**1**)<=**D**(**1**);** -- input will load in to  t**(**2**)<=**D**(**2**);** -- the temporary signal  t**(**3**)<=**D**(**3**);** --  -- and then  t**(**0**)<=**SIL**;** -- t(0),t(1) and t(2) will  t**(**1**)<=**t**(**0**);** -- shift left and SIL will  t**(**2**)<=**t**(**1**);** -- assign to t(0)  t**(**3**)<=**t**(**2**);** --  -- and then  Q**(**0**)<=**t**(**0**);** -- temporary signal will  Q**(**1**)<=**t**(**1**);** -- load in to the output  Q**(**2**)<=**t**(**2**);** --  Q**(**3**)<=**t**(**3**);** --    --Shift Right  **when** "10" **=>**  t**(**0**)<=**D**(**0**);** -- When S at 10, parallel  t**(**1**)<=**D**(**1**);** -- input will load in to  t**(**2**)<=**D**(**2**);** -- the temporary signal  t**(**3**)<=**D**(**3**);** --  -- and then  t**(**0**)<=**t**(**1**);** -- t(3),t(2) and t(1) will  t**(**1**)<=**t**(**2**);** -- shift right and SIR will  t**(**2**)<=**t**(**3**);** -- assign to t(3)  t**(**3**)<=**SIR**;** --  -- and then  Q**(**0**)<=**t**(**0**);** -- temporary signal will  Q**(**1**)<=**t**(**1**);** -- load in to the output  Q**(**2**)<=**t**(**2**);** --  Q**(**3**)<=**t**(**3**);** --    --Hold  **when** "00" **=>**  t**(**0**)<=**t**(**0**);** -- When S at 00, temporary  t**(**1**)<=**t**(**1**);** -- input will not change  t**(**2**)<=**t**(**2**);** --  t**(**3**)<=**t**(**3**);** --  -- and then  Q**(**0**)<=**t**(**0**);** -- temporary signal will  Q**(**1**)<=**t**(**1**);** -- load in to the output  Q**(**2**)<=**t**(**2**);** --  Q**(**3**)<=**t**(**3**);** --    **when** **others** **=>** **null;**    **end** **case;**    **end** **if;**    **end** **process;**  **end** Behavioral**;** |

**TEST BENCH**

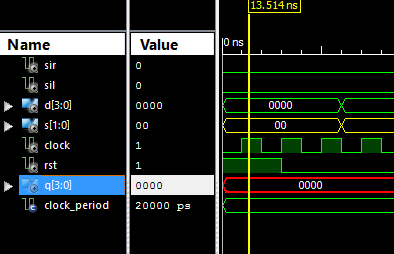
|  |
| --- |
| **LIBRARY** ieee**;**  **USE** ieee**.**std\_logic\_1164**.ALL;**  **ENTITY** Uni\_Shift\_Reg\_TB **IS**  **END** Uni\_Shift\_Reg\_TB**;**    **ARCHITECTURE** behavior **OF** Uni\_Shift\_Reg\_TB **IS**    -- Component Declaration for the Unit Under Test (UUT)    **COMPONENT** Uni\_Shift\_Reg  **PORT(**  SIR **:** **IN** std\_logic**;**  SIL **:** **IN** std\_logic**;**  D **:** **IN** std\_logic\_vector**(**3 **downto** 0**);**  S **:** **IN** std\_logic\_vector**(**1 **downto** 0**);**  Q **:** **OUT** std\_logic\_vector**(**3 **downto** 0**);**  clock **:** **IN** std\_logic**;**  RST **:** **IN** std\_logic  **);**  **END** **COMPONENT;**    --Inputs  **signal** SIR **:** std\_logic **:=** '0'**;**  **signal** SIL **:** std\_logic **:=** '0'**;**  **signal** D **:** std\_logic\_vector**(**3 **downto** 0**)** **:=** **(others** **=>** '0'**);**  **signal** S **:** std\_logic\_vector**(**1 **downto** 0**)** **:=** **(others** **=>** '0'**);**  **signal** clock **:** std\_logic **:=** '0'**;**  **signal** RST **:** std\_logic **:=** '0'**;**  --Outputs  **signal** Q **:** std\_logic\_vector**(**3 **downto** 0**);**  -- Clock period definitions  **constant** clock\_period **:** time **:=** 20 ns**;**    **BEGIN**    -- Instantiate the Unit Under Test (UUT)  uut**:** Uni\_Shift\_Reg **PORT** **MAP** **(**  SIR **=>** SIR**,**  SIL **=>** SIL**,**  D **=>** D**,**  S **=>** S**,**  Q **=>** Q**,**  clock **=>** clock**,**  RST **=>** RST  **);**  -- Clock process definitions  clock\_process **:process**  **begin**  clock **<=** '0'**;**  **wait** **for** clock\_period**/**2**;**  clock **<=** '1'**;**  **wait** **for** clock\_period**/**2**;**  **end** **process;**    -- Stimulus process  stim\_proc**:** **process**  **begin**  -- test RST and HOLD  RST**<=**'1'**;**  **wait** **for** 30ns**;**  RST**<=**'0'**;**  S**<=**"00"**;**  **wait** **for** 30ns**;**  -- test parallel loading    S**<=**"11"**;**  D**<=**"0110"**;**  **wait** **for** 80ns**;**  -- test shift right  S**<=**"01"**;**  SIL**<=**'1'**;**  **wait** **for** 160ns**;**  -- test shift left  S**<=**"10"**;**  SIR**<=**'0'**;**  **wait** **for** 160ns**;**    **end** **process;**  **END;** |

**SIMULATION RESULTS**

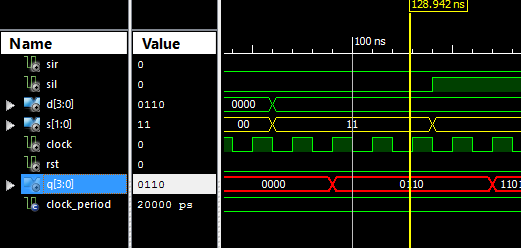
**FULL WAVEFORM**



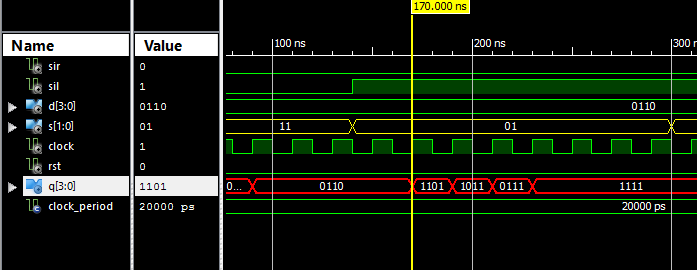
**RESET and HOLD**



**PARALLEL LOADING**



**SHIFT LEFT**



**SHIFT RIGHT**

