**MAIN VHDL CODE**

**library** IEEE**;**

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

**entity** USR\_4 **is**

-----------------------INPUTS AND OUTPUTS DECLARATION------------------------

**port** **(**

CLK **:** **IN** STD\_LOGIC**;** --CLOCK INPUT

RST **:** **IN** STD\_LOGIC**;** --ASYNCHRONOUS RESET INPUT

SIR **:** **IN** STD\_LOGIC**;** --SERIAL INPUT FOR RIGHT SHIFT

SIL **:** **IN** STD\_LOGIC**;** --SERIAL INPUT FOR LEFT SHIFT

D **:** **IN** STD\_LOGIC\_VECTOR **(**3 **DOWNTO** 0**);** --PARALLEL INPUTS FOR PARALLEL LOADING

S **:** **IN** STD\_LOGIC\_VECTOR **(**1 **DOWNTO** 0**);** --INPUTS FOR SELECTING OPERATION

Q **:** **OUT** STD\_LOGIC\_VECTOR **(**3 **DOWNTO** 0**)** --DATA OUTPUTS

**);**

**end** USR\_4**;**

**architecture** Behavioral **of** USR\_4 **is**

-----------------------DECLARATION OF TEMPORARY SIGNAL-----------------------

**SIGNAL** TEMP**:** STD\_LOGIC\_VECTOR **(**3 **DOWNTO** 0**);**

**begin**

**PROCESS** **(**SIR**,**SIL**,**D**,**S**,**CLK**,**RST**)**

**BEGIN**

-------------------------INITIALIZATI0N--------------------------------------

**IF** RST**=**'1' **THEN**

TEMP**<=**"0000"**;** --LOAD 0000 TO TEMPORARY SIGNAL

Q**<=**"0000"**;** --LOAD 0000 TO OUTPUT

**ELSIF** **(**CLK**=**'1' **AND** CLK' **EVENT)** **THEN**

**CASE** S **IS**

--------------------------PARALLEL LOADING-----------------------------------

**WHEN** "11" **=>**

TEMP**<=**D**;**

Q**<=**TEMP**;**

---------------------------SHIFT LEFT----------------------------------------

**WHEN** "01" **=>**

TEMP**<=**D**;**

TEMP**(**3 **DOWNTO** 1**)** **<=** TEMP**(**2 **DOWNTO** 0**);**

TEMP**(**0**)** **<=** SIL**;**

Q**<=**TEMP**;**

---------------------------SHIFT RIGHT---------------------------------------

**WHEN** "10" **=>**

TEMP**<=**D**;**

TEMP**(**2 **DOWNTO** 0**)** **<=** TEMP**(**3 **DOWNTO** 1**);**

TEMP**(**3**)** **<=** SIR**;**

Q**<=**TEMP**;**

---------------------------HOLD----------------------------------------------

**WHEN** **OTHERS** **=>** **NULL;**

**END** **CASE;**

**END** **IF;**

**END** **PROCESS;**

**end** Behavioral**;**

**TEST BENCH**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.ALL;**

**ENTITY** USR\_4\_TB **IS**

**END** USR\_4\_TB**;**

**ARCHITECTURE** behavior **OF** USR\_4\_TB **IS**

-- Component Declaration for the Unit Under Test (UUT)

**COMPONENT** USR\_4

**PORT(**

CLK **:** **IN** std\_logic**;**

RST **:** **IN** std\_logic**;**

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**)**

**);**

**END** **COMPONENT;**

--Inputs

**signal** CLK **:** std\_logic **:=** '0'**;**

**signal** RST **:** std\_logic **:=** '0'**;**

**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'**);**

--Outputs

**signal** Q **:** std\_logic\_vector**(**3 **downto** 0**);**

-- Clock period definitions

**constant** CLK\_period **:** time **:=** 20 ns**;**

**BEGIN**

-- Instantiate the Unit Under Test (UUT)

uut**:** USR\_4 **PORT** **MAP** **(**

CLK **=>** CLK**,**

RST **=>** RST**,**

SIR **=>** SIR**,**

SIL **=>** SIL**,**

D **=>** D**,**

S **=>** S**,**

Q **=>** Q

**);**

-- Clock process definitions

CLK\_process **:process**

**begin**

CLK **<=** '0'**;**

**wait** **for** CLK\_period**/**2**;**

CLK **<=** '1'**;**

**wait** **for** CLK\_period**/**2**;**

**end** **process;**

-- Stimulus process

**PROCESS**

**BEGIN**

--------------------------INITIALIZE TEST---------------------------------

RST**<=**'1'**;**

**WAIT** **FOR** 100 NS**;**-- HOLD AT INITIAL STATE FOR 100 NS.

--------------------------TEST PARALLEL LOADING---------------------------

RST**<=**'0'**;**

S**<=**"11"**;**

D**<=**"1111"**;**

**WAIT** **FOR** 100 NS**;**

--------------------------TESTING SHIFT RIGHT-----------------------------

S**<=**"10"**;**

SIR**<=** '0'**;**

**WAIT** **FOR** 100 NS**;**

--------------------------TESTING HOLD------------------------------------

S**<=**"00"**;**

**WAIT** **FOR** 100 NS**;**

--------------------------TESTING SHIFT LEFT------------------------------

S**<=**"01"**;**

SIL**<=** '1'**;**

**WAIT** **FOR** 100 NS**;**

--------------------------TESTING RESET-----------------------------------

RST**<=**'1'**;**

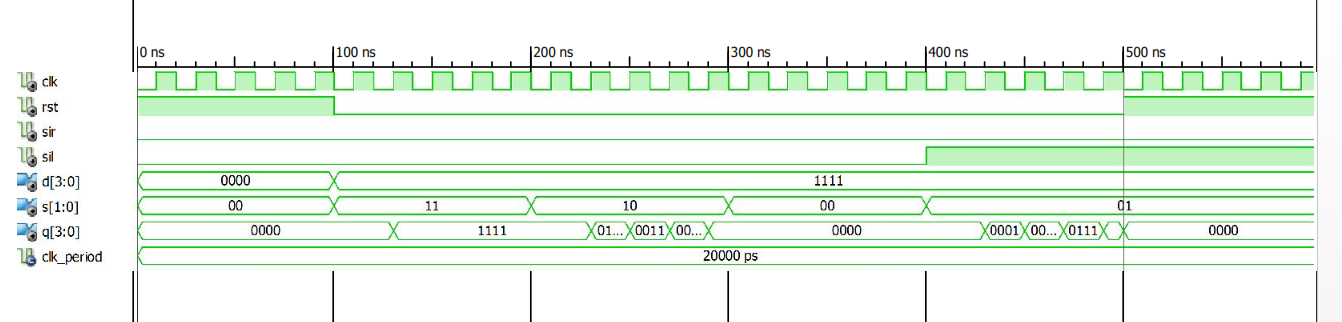
**WAIT** **FOR** 100 NS**;**-- HOLD AT INITIAL STATE FOR 100 NS.

**END** **PROCESS;**

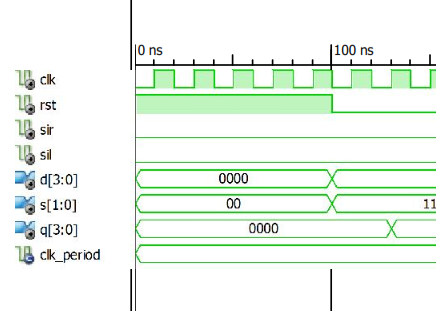
**END;**

**SIMULATION RESULTS**

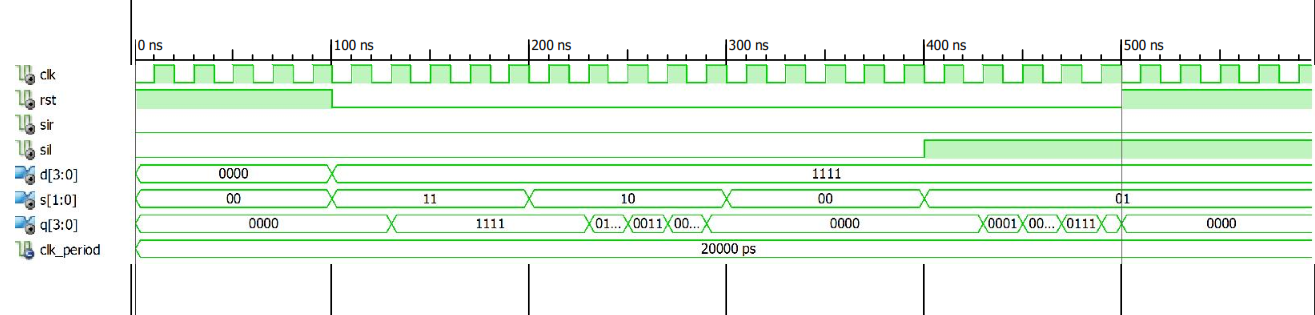
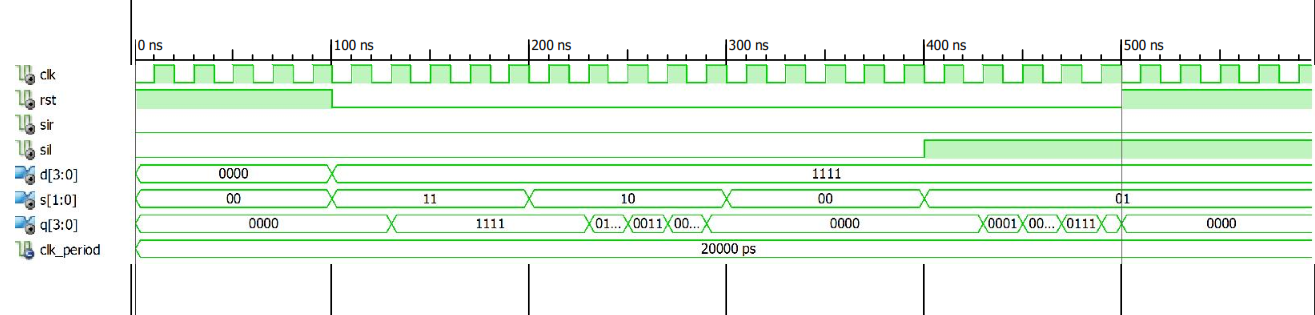
**FULL WAVEFORM**



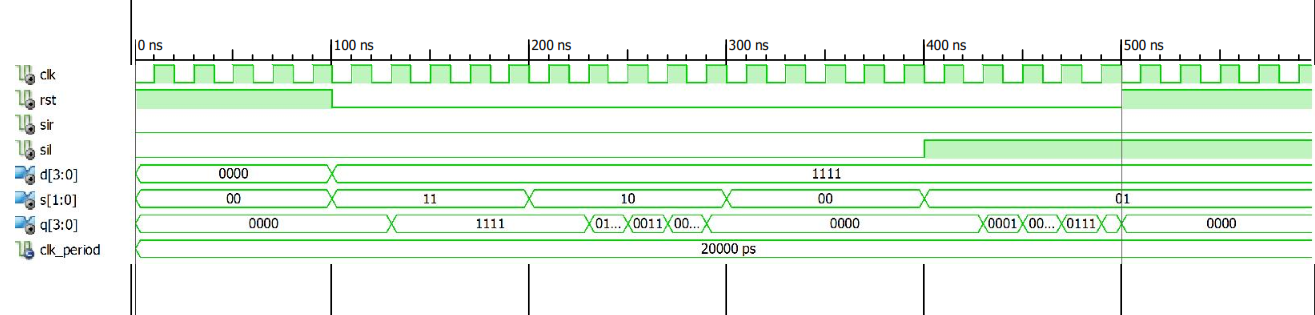
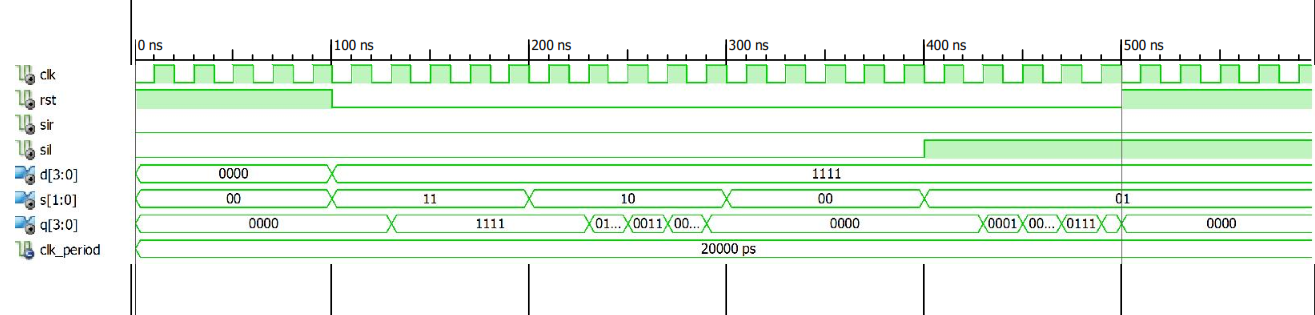
**Initialization (UNTILL 100NS)**



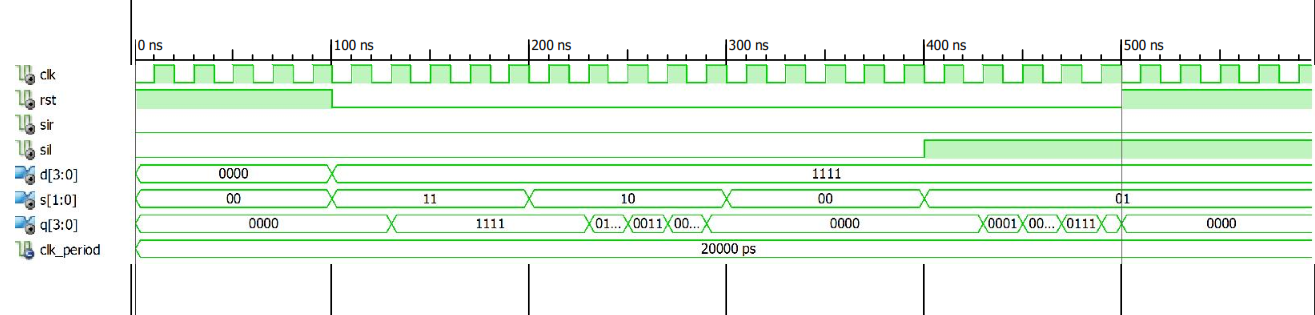
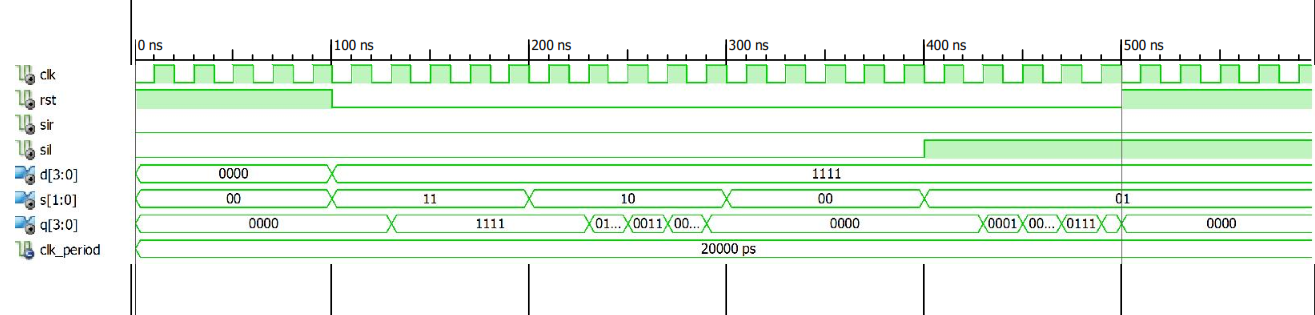
**Parallel Loading**



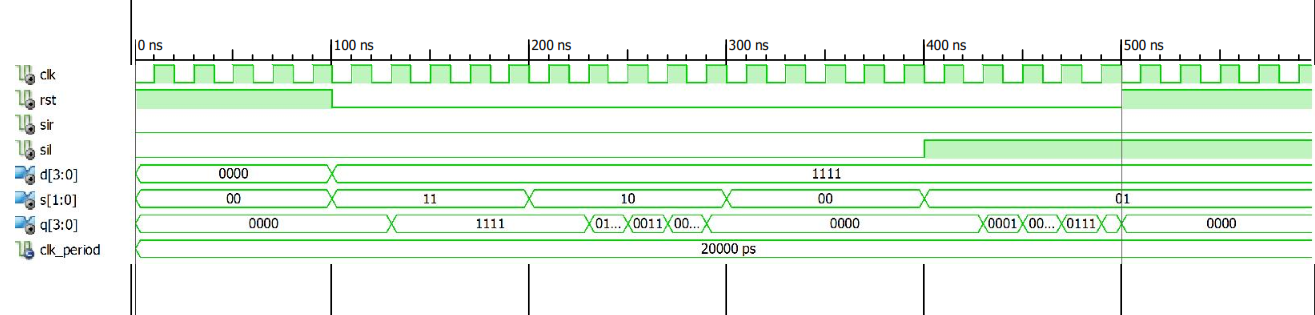
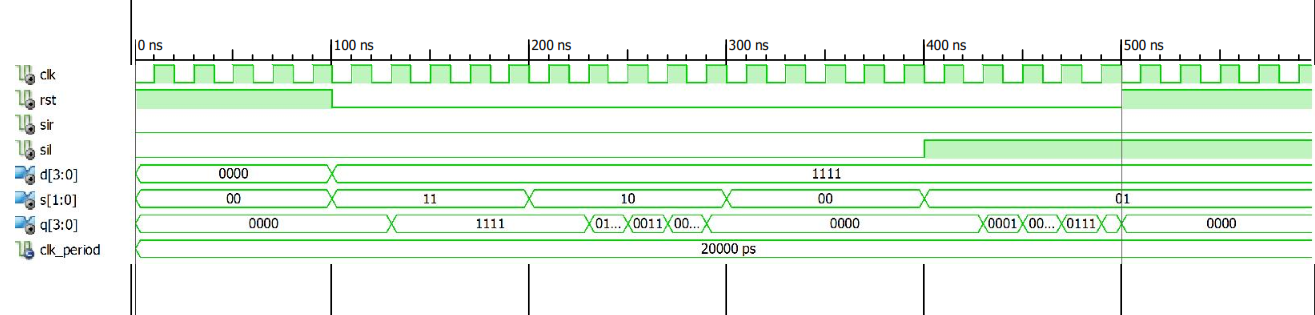
**Right Shift**



**Hold**



**Left Shift**



**Reset**

