```
architecture UNI_Shift_Register_arch of UNI_Shift_Register is SIGNAL temp: STD_LOGIC_VECTOR (3 downto 0):="0000"; begin

PROCESS(rst, clk, mode, Sin, Pin) BEGIN

IF rst='1' THEN Pout <= "0000"; Sout <= '0';

ELSIF FALLING EDGE(clk) THEN

CASE mode IS

WHEN "00" =>

temp(3 downto 1) <= temp(2 downto 0); temp(0) <= Sin;

Sout <= temp(3); Pout <= "0000";

WHEN "01" =>

temp(3 downto 1) <= temp(2 downto 0); temp(0) <= Sin;

Pout <= temp;

Sout <= '0'

WHEN "10" => temp <= Pin; Sout <= temp(3);

temp(3 downto 1) <= temp(2 downto 0); Pout <= "0000";

WHEN OTHERS =>

Pout <= Pin; Sout <= '0';
```

```
END CASE;

END IF; END PROCESS;

end UNI_Shift_Register_arch;

tbb

-Inputs

signal rst std_logic := '0',

signal clk: std_logic := '1';

signal mode: std_logic_vector(1 downto 0) = (others => '0'); signal Sin: std_logic := '0';

signal Pin: std_logic_vector(3 downto 0) = "1010";

--Outputs

signal Sout: std_logic;

signal Pout: std_logic_vector(3 downto 0);

- Clock period definitions constant cik period: time: 10 ns;

BEGIN

- Instantiate the Unit Under Test (UUT)
```

```
uut: UNI_Shift_Register PORT MAP (

rst => rst,

clk => clk, mode => mode, Sin => Sin, Pin => Pin, Sout => Sout, Pout => Pout );

Clock process definitions

cik_process process

begin

clk<=NOT(cik); wait for clk_period/2; end process;

- Stimulus process stim_proc_mode: process begin mode<="00";

wait for 80 ns;

mode<="01"; wait for 50 ns;

mode<="10"; wait for 50 ns;

mode<="11";
wait for 20 ns; end process;

stim_proc_Sin process begin wait for 10 ns;

Sin<='1';

wait for 10 ns;
```

```
Sin<='0';

wait for 10 ns;

Sin<='1';

wait for 10 ns;

Sin<='0';

wait for 10 ns;

Sin<= '0';

wait for 40 ns;

Sin<='1';

wait for 10 ns;

Sin<='0';

wait for 10 ns;

Sin<='1';

wait for 10 ns;

Sin<='0';

wait for 10 ns;
```

```
Sin<= '0';

wait;

end process;

stim_proc_rst:process

begin

rst<='0';

wait for 300 ns;

rst<='1';

wait for 10 ns;

end process;

END;
```