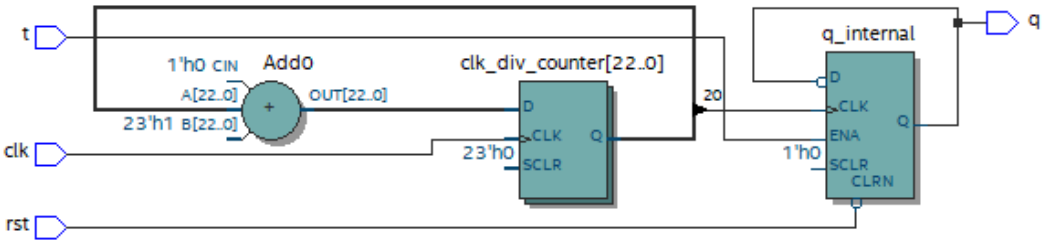
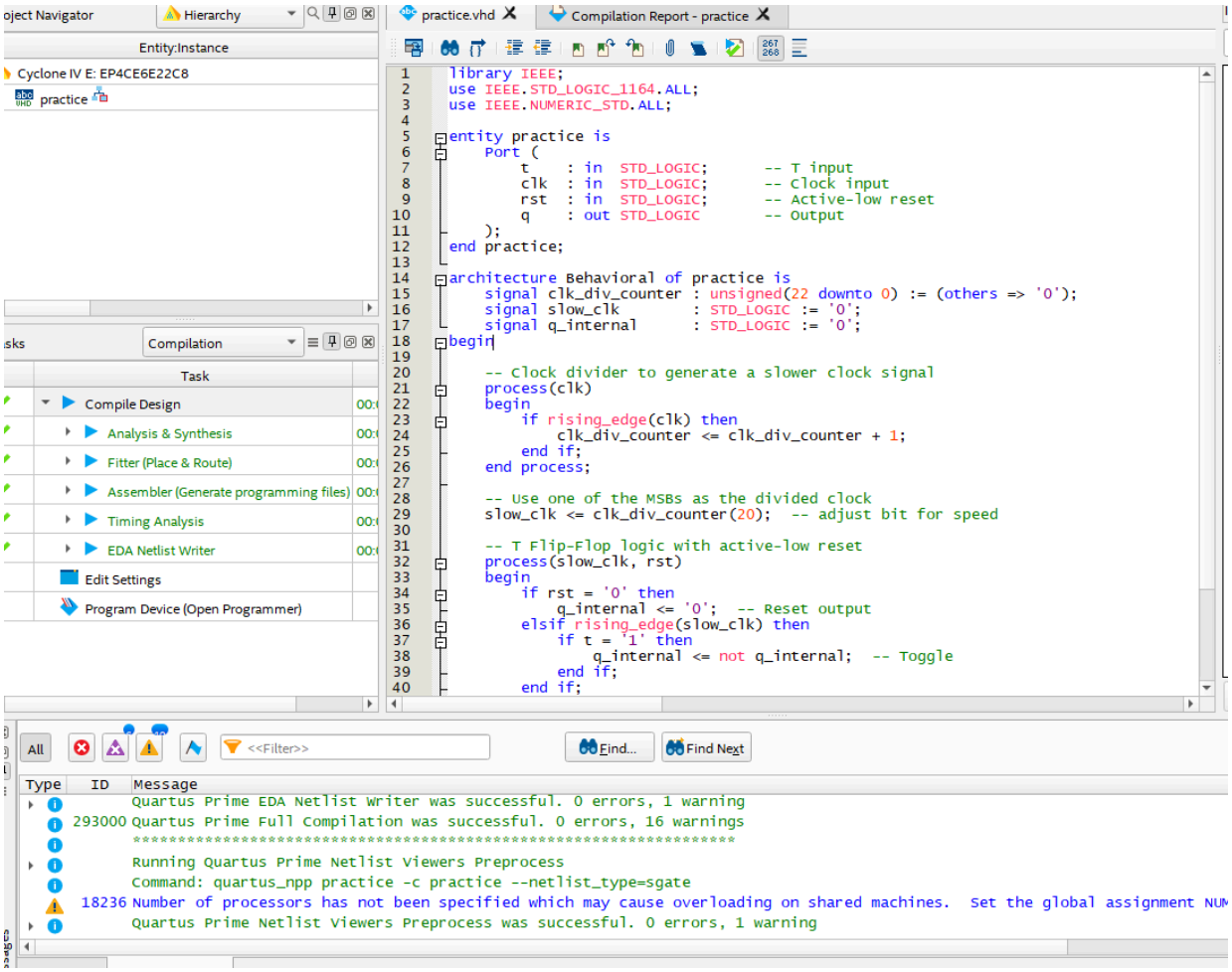
EMMAN ACE G. MENION
BSCPE 3A

VHDL CODE FOR T FLIP FLOP:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity practice is
   Port (
      t    : in  STD_LOGIC;      -- T input
      clk  : in  STD_LOGIC;      -- Clock input
      rst  : in  STD_LOGIC;      -- Active-low reset
      q    : out STD_LOGIC       -- Output
   );
end practice;

architecture Behavioral of practice is
   signal clk_div_counter : unsigned(22 downto 0) := (others => '0');
   signal slow_clk        : STD_LOGIC := '0';
   signal q_internal      : STD_LOGIC := '0';
begin

   -- Clock divider to generate a slower clock signal
   process(clk)
   begin
      if rising_edge(clk) then
         clk_div_counter <= clk_div_counter + 1;
      end if;
   end process;

   -- Use one of the MSBs as the divided clock
   slow_clk <= clk_div_counter(20);  -- adjust bit for speed

   -- T Flip-Flop logic with active-low reset
   process(slow_clk, rst)
   begin
      if rst = '0' then
         q_internal <= '0';  -- Reset output
      elsif rising_edge(slow_clk) then
         if t = '1' then
            q_internal <= not q_internal;  -- Toggle
         end if;
      end if;
   end process;

   q <= q_internal;

end Behavioral;
```

Clear	T	Clock	Qn+1	$\overline{Qn+1}$
1	0	0	0	$\overline{Qn}$
0	1	⎍	$\overline{Qn}$	Qn


t
clk
rst
1'h0 CIN
Add0
A[22..0]
23'h1 B[22..0]
OUT[22..0]
clk_div_counter[22..0]
D
CLK
Q
23'h0
SCLR
20
q_internal
D
CLK
ENA
Q
1'h0
SCLR
CLRN
q


oject Navigator
Hierarchy
Entity:Instance
Cyclone IV E: EP4CE6E22C8
practice
practice.vhd
Compilation Report - practice
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.NUMERIC_STD.ALL;
4
5 entity practice is
6 Port (
7 t : in STD_LOGIC; -- T input
8 clk : in STD_LOGIC; -- Clock input
9 rst : in STD_LOGIC; -- Active-low reset
10 q : out STD_LOGIC -- Output
11);
12 end practice;
13
14 architecture Behavioral of practice is
15 signal clk_div_counter : unsigned(22 downto 0) := (others => '0');
16 signal slow_clk : STD_LOGIC := '0';
17 signal q_internal : STD_LOGIC := '0';
18 begin
19
20 -- Clock divider to generate a slower clock signal
21 process(clk)
22 begin
23 if rising_edge(clk) then
24 clk_div_counter <= clk_div_counter + 1;
25 end if;
26 end process;
27
28 -- Use one of the MSBs as the divided clock
29 slow_clk <= clk_div_counter(20); -- adjust bit for speed
30
31 -- T Flip-Flop logic with active-low reset
32 process(slow_clk, rst)
33 begin
34 if rst = '0' then
35 q_internal <= '0'; -- Reset output
36 elsif rising_edge(slow_clk) then
37 if t = '1' then
38 q_internal <= not q_internal; -- Toggle
39 end if;
40 end if;
sks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
All
<<Filter>>
Find...
Find Next
Type
ID
Message
Quartus Prime EDA Netlist Writer was successful. 0 errors, 1 warning
293000 Quartus Prime Full Compilation was successful. 0 errors, 16 warnings
Running Quartus Prime Netlist Viewers Preprocess
Command: quartus_npp practice -c practice --netlist_type=sgate
18236 Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM
Quartus Prime Netlist Viewers Preprocess was successful. 0 errors, 1 warning