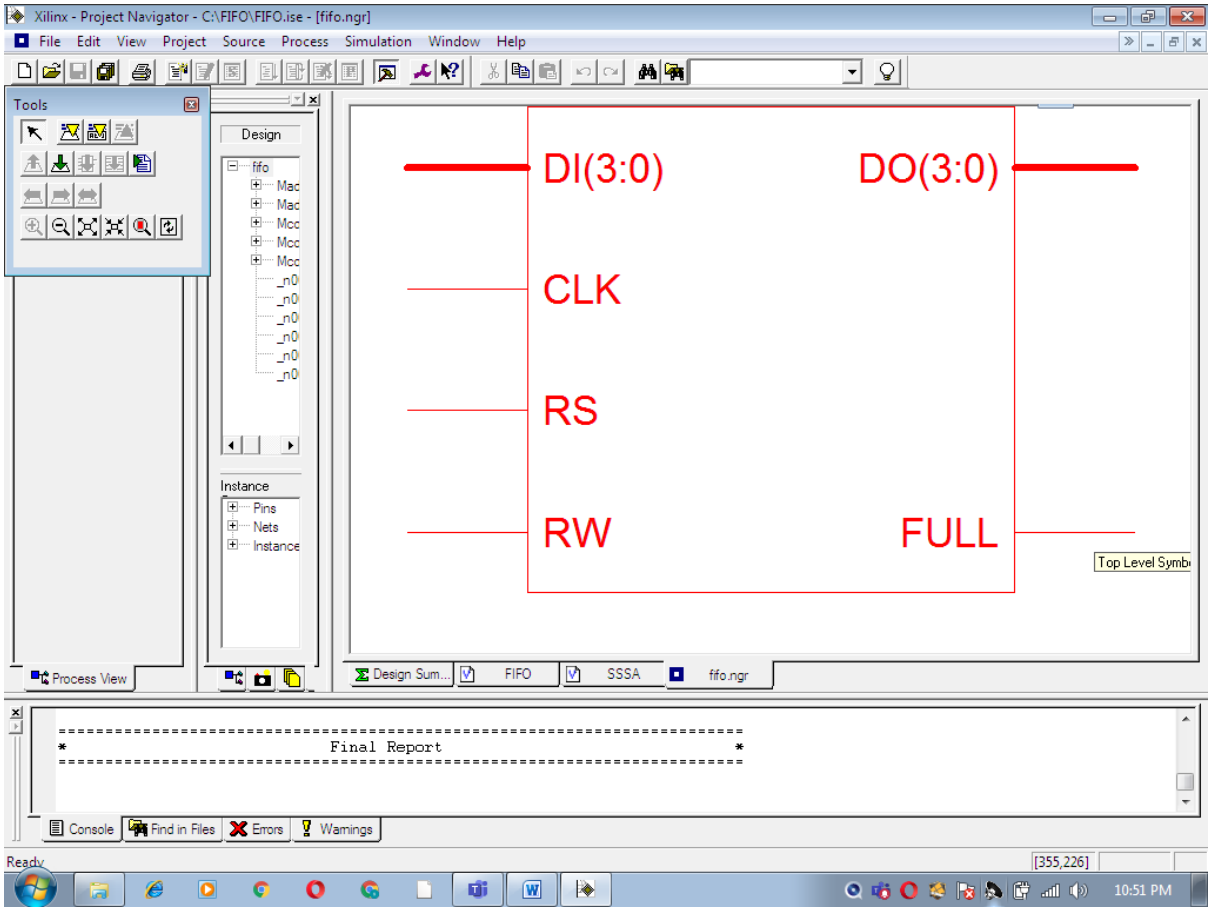
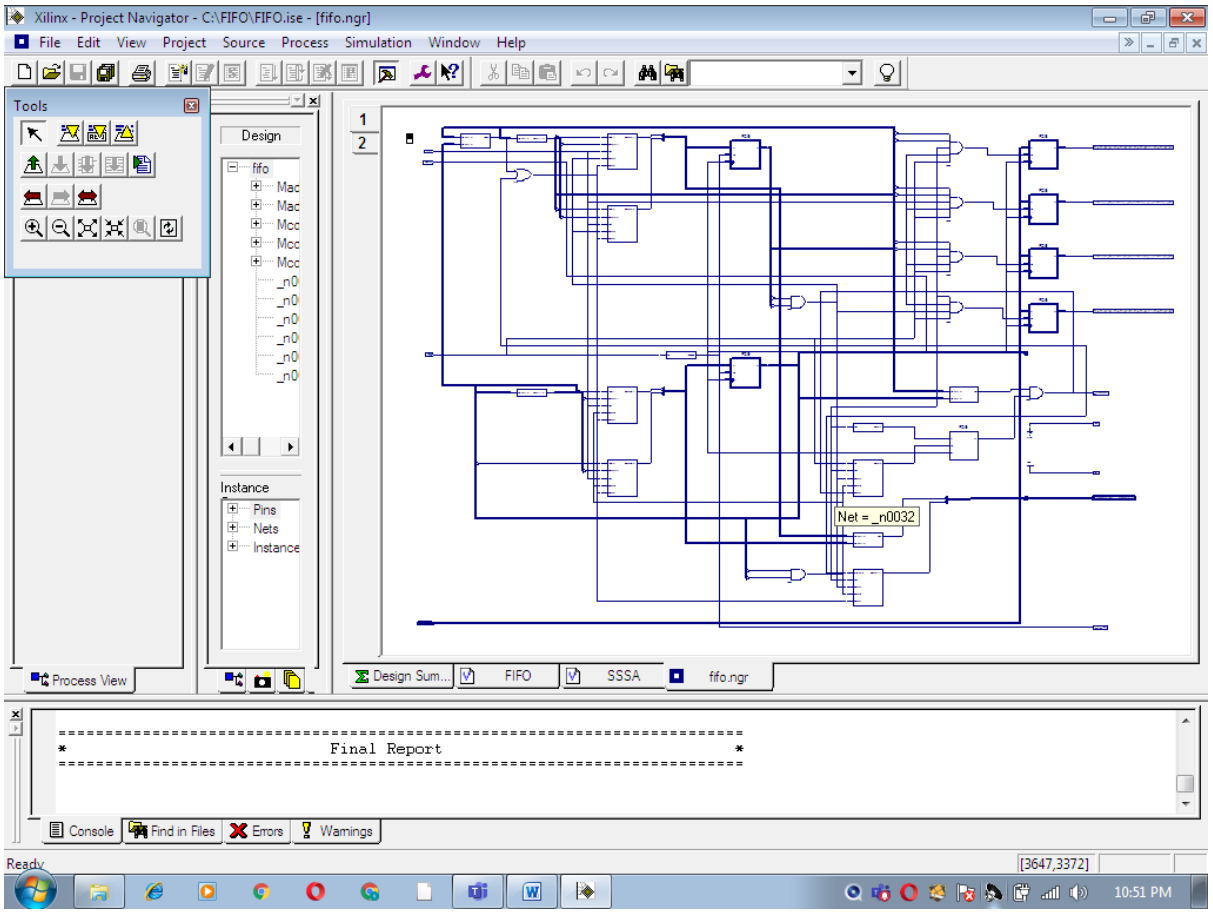
Name:- Swarupa Omkar Maskalle Roll NO. E41038

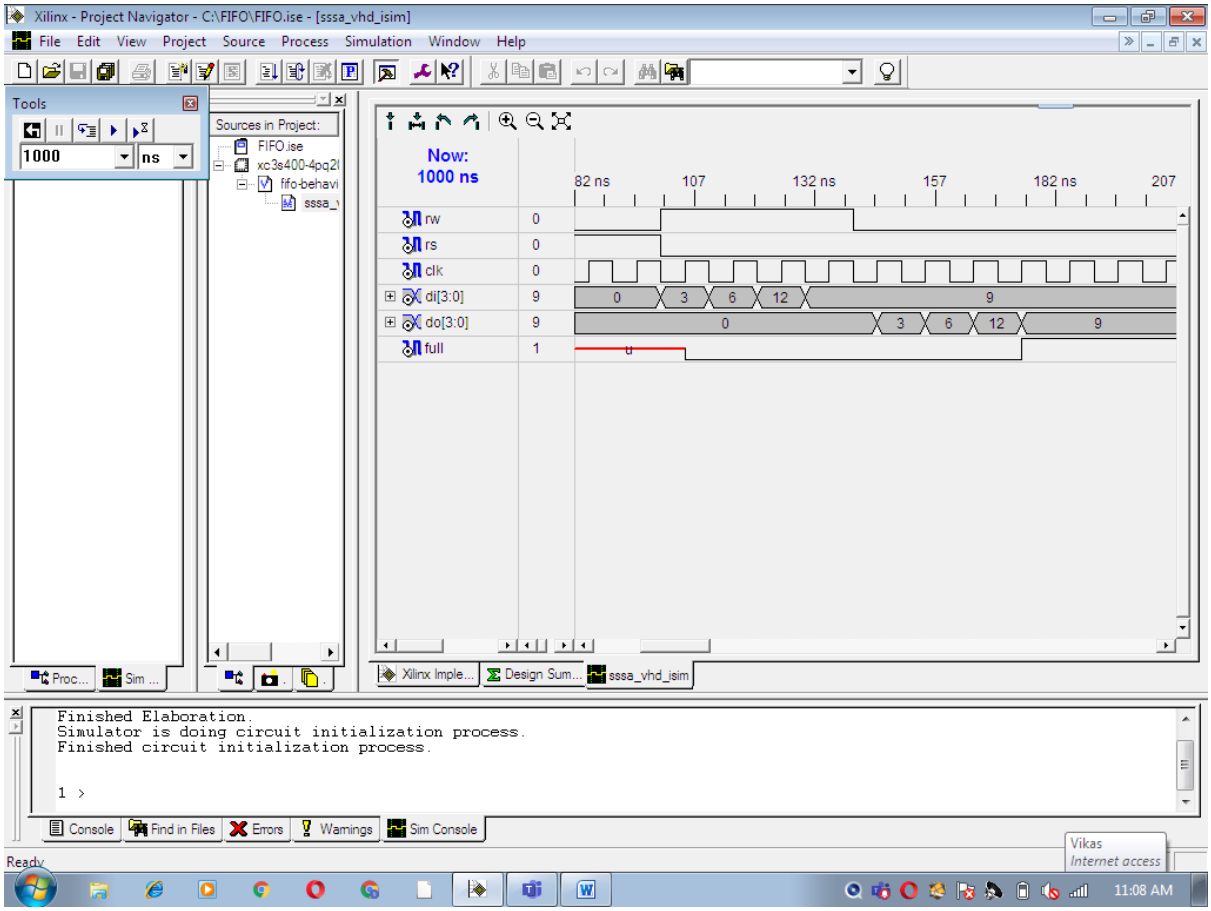
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity FIFO is
   Port ( DI : in std_logic_vector(3 downto 0);
          DO : out std_logic_vector(3 downto 0);
          RW : in std_logic;
          FULL : out std_logic;
          RS : in std_logic;
          CLK : in std_logic);
end FIFO;

architecture Behavioral of FIFO is

begin

process(RS,CLK)
        type memory is Array( 0 to 3)of std_logic_vector(3 downto 0);
                        variable mem: memory;

                        variable  r_p:integer range 0 to 3;
                          variable  w_p:integer range 0 to 3;
                          variable  overwrite :boolean;
begin
if RS='1' then
DO<="0000";
elsif(CLK'event and CLK='1') then
if  RW='1'then

                        if  (overwrite=False OR w_p/=r_p) then
                        mem (w_p):=DI;
                                if w_p=3  then
                                        w_p:=0 ;
                                        else

                                        w_p:=w_p+1;
                                        overwrite:=False;
                                end if;
                                end if;

                               elsif       RW='0' then
                        if          (overwrite=False OR w_p/=r_p) then
                DO<=mem(r_p);
                                 if (r_p=3 ) then
                                        r_p:=0 ;
                                        overwrite:=true;
```

```
                                    else
                                r_p:=r_p+1;
                        end if;
                        end if;
                        end if;

                        if      w_p=r_p         then

                        if overwrite=true then
                FULL<='1';

                         else
                                FULL<='0';

                        end if;

                                 else
                                 FULL<='0';
                                end if;
                                end if;
                        end process;

end Behavioral;

ENTITY SSSA_vhd IS
END SSSA_vhd;

ARCHITECTURE behavior OF SSSA_vhd IS

        -- Component Declaration for the Unit Under Test (UUT)
        COMPONENT fifo
        PORT(
                DI : IN std_logic_vector(3 downto 0);
                RW : IN std_logic;
                RS : IN std_logic;
                CLK : IN std_logic;
                DO : OUT std_logic_vector(3 downto 0);
                FULL : OUT std_logic
                );
        END COMPONENT;
        --Inputs
        SIGNAL RW :  std_logic := '0';
        SIGNAL RS :  std_logic := '0';
        SIGNAL CLK :  std_logic := '0';
        SIGNAL DI :  std_logic_vector(3 downto 0) := (others=>'0');

        --Outputs
        SIGNAL DO :  std_logic_vector(3 downto 0);
        SIGNAL FULL :  std_logic;
        constant        CLK_period : time := 10 ns;

        BEGIN

        -- Instantiate the Unit Under Test (UUT)
        uut: fifo PORT MAP(
```

```
                DI => DI,
                DO => DO
        RW => RW,
        FULL => FULL,
        RS => RS,
        CLK => CLK
        );
        process
        begin
                CLK<='0'        ;
        wait for CLK_period/2;
                        CLK<='1'        ;
        wait for CLK_period/2;
        end process;
         tb : PROCESS
        BEGIN
                 RS<='1';
        wait for 100 ns;
        RS<='0';
        RW<='1';
        DI<="0011";

                -- Wait 100 ns for global reset to finish
                wait for 10 ns;
                RW<='1';
        DI<="0110";

-- Wait 100 ns for global reset to finish
                wait for 10 0
RW<='1';
        DI<="1100";

                -- Wait 100 ns for global reset to finish
                wait for 10 ns;
                RW<='1';
        DI<="1001";

                -- Wait 100 ns for global reset to finish
                wait for 10 ns;
                -- Place stimulus here
                                        RW<='0';
                -- Wait 100 ns for global reset to finish
                wait for 10 ns;
                RW<='0';
                -- Wait 100 ns for global reset to finish
                wait for 10 ns;
                RW<='0';
                -- Wait 100 ns for global reset to finish
                wait for 10 ns;
                WAIT FOR CLK_period*10;
                wait; -- will wait forever
        END PROCESS;

END;
```


Xilinx - Project Navigator - C:\FIFO\FIFO.ise - [fifo.ngr]
File Edit View Project Source Process Simulation Window Help
Tools
Design
fifo
Instance
Pins
Nets
Instance
DI(3:0)
DO(3:0)
CLK
RS
RW
FULL
Top Level Symb
Process View
Design Sum...
FIFO
SSSA
fifo.ngr
Final Report
Console
Find in Files
Errors
Warnings
Ready
[355,226]
10:51 PM


Xilinx - Project Navigator - C:\FIFO\FIFO.ise - [fifo.ngr]
File Edit View Project Source Process Simulation Window Help
Tools
Design
fifo
Instance
Pins
Nets
Instance
1
2
Net = _n0032
Process View
Design Sum...
FIFO
SSSA
fifo.ngr
Final Report
Console
Find in Files
Errors
Warnings
Ready
[3647,3372]
10:51 PM



==========

=

* Synthesis Options Summary *

==========

=

---- Source Parameters

Input File Name : "fifo.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "fifo"

Output Format : NGC

Target Device : xc3s400-4-pq208

---- Source Options

Top Module Name : fifo

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

FSM Style : lut

RAM Extraction : Yes

RAM Style : Auto

ROM Extraction : Yes

ROM Style : Auto

Mux Extraction : YES

Decoder Extraction : YES

Priority Encoder Extraction : YES

Shift Register Extraction : YE

```
Logical Shifter Extraction        : YES
XOR Collapsing                    : YES
Resource Sharing                  : YES
Multiplier Style                  : auto
Automatic Register Balancing      : No

---- Target Options
Add IO Buffers                    : YES
Global Maximum Fanout             : 500
Add Generic Clock Buffer(BUFG)    : 8
Register Duplication              : YES
Equivalent register Removal       : YES
Slice Packing                     : YES
Pack IO Registers into IOBs       : au
---- General Options
Optimization Goal                 : Speed
Optimization Effort               : 1
Keep Hierarchy                    : NO
Global Optimization               : AllClockNets
RTL Output                        : ONLY
Write Timing Constraints          : NO
Hierarchy Separator               : /
Bus Delimiter                     : <>
Case Specifier                    : maintain
Slice Utilization Ratio           : 100
Slice Utilization Ratio Delta     : 5

---- Other Options
lso                               : fifo.lso
Read Cores                        : YES
cross_clock_analysis              : NO
verilog2001                       : YES
safe_implementation               : No
Optimize Instantiated Primitives  : NO
use_clock_enable                  : Yes
use_sync_set                      : Yes
use_sync_reset                    : Yes
enable_auto_floorplanning         : No

=========================================================================
=


=========================================================================
=
*                     HDL Compilation                   *
=========================================================================
=
Compiling vhdl file "C:/FIFO/FIFO.vhd" in Library work.
Entity <FIFO> compiled.
Entity <FIFO> (Architecture <Behavioral>) compiled.
=========================================================================
=
*                     HDL Analysis                      *
=========================================================================
```

```
Analyzing Entity <fifo> (Architecture <Behavioral>).
Entity <fifo> analyzed. Unit <fifo> generated.


=========================================================================
=
*                     HDL Synthesis                       *
=========================================================================
=

Synthesizing Unit <fifo>.
    Related source file is "C:/FIFO/FIFO.vhd".
    Found 4-bit register for signal <DO>.
    Found 1-bit register for signal <FULL>.
    Found 4-bit 4-to-1 multiplexer for signal <$n0015> created at line 69.
    Found 1-bit 4-to-1 multiplexer for signal <$n0018>.
    Found 2-bit adder for signal <$n0019> created at line 62.
    Found 2-bit adder for signal <$n0020> created at line 75.
    Found 2-bit comparator equal for signal <$n0021> created at line 81.
    Found 2-bit comparator not equal for signal <$n0022> created at line 56.
    Found 2-bit comparator equal for signal <$n0026> created at line 56.
    Found 16-bit register for signal <mem>.
    Found 1-bit register for signal <overwrite<0>>.
    Found 2-bit register for signal <r_p>.
    Found 2-bit register for signal <w_p>.
    Summary:
        inferred  26 D-type flip-flop(s).
        inferred   2 Adder/Subtractor(s).
        inferred   3 Comparator(s).
        inferred   5 Multiplexer(s).
Unit <fifo> synthesized.

=========================================================================
=
*                  Advanced HDL Synthesis                 *
=========================================================================
=

Advanced RAM inference ...
Advanced multiplier inference ...
Advanced Registered AddSub inference ...
Dynamic shift register inference ...

=========================================================================
=
HDL Synthesis Report

Macro Statistics
# Adders/Subtractors            : 2
 2-bit adder                   : 2
# Registers                     : 9
 1-bit register                : 2
 2-bit register                : 2
 4-bit register                : 5
# Comparators                   : 3
```

```
 2-bit comparator equal           : 2
 2-bit comparator not equal       : 1
# Multiplexers                   : 2
 1-bit 4-to-1 multiplexer         : 1
 4-bit 4-to-1 multiplexer         : 1

=========================================================================
=

=========================================================================
=
*                     Low Level Synthesis                               *
=========================================================================
=
=========================================================================
=
*                       Final Report                                    *
=========================================================================
=
Final Results
RTL Top Level Output File Name     : fifo.ngr
Keep Hierarchy                     : NO

Design Statistics
# IOs                              : 12

Cell Usage :
# BELS                             : 2
#      GND                         : 1
#      VCC                         : 1
# FlipFlops/Latches                : 26
#      FDCE                        : 4
#      FDE                         : 22
=========================================================================
=
CPU : 5.92 / 8.26 s | Elapsed : 6.00 / 8.00 s

-->

Total memory usage is 122500 kilobytes

Number of errors   :    0 (   0 filtered)
Number of warnings :    0 (   0 filtered)
Number of infos    :    0 (   0 filtered)
```