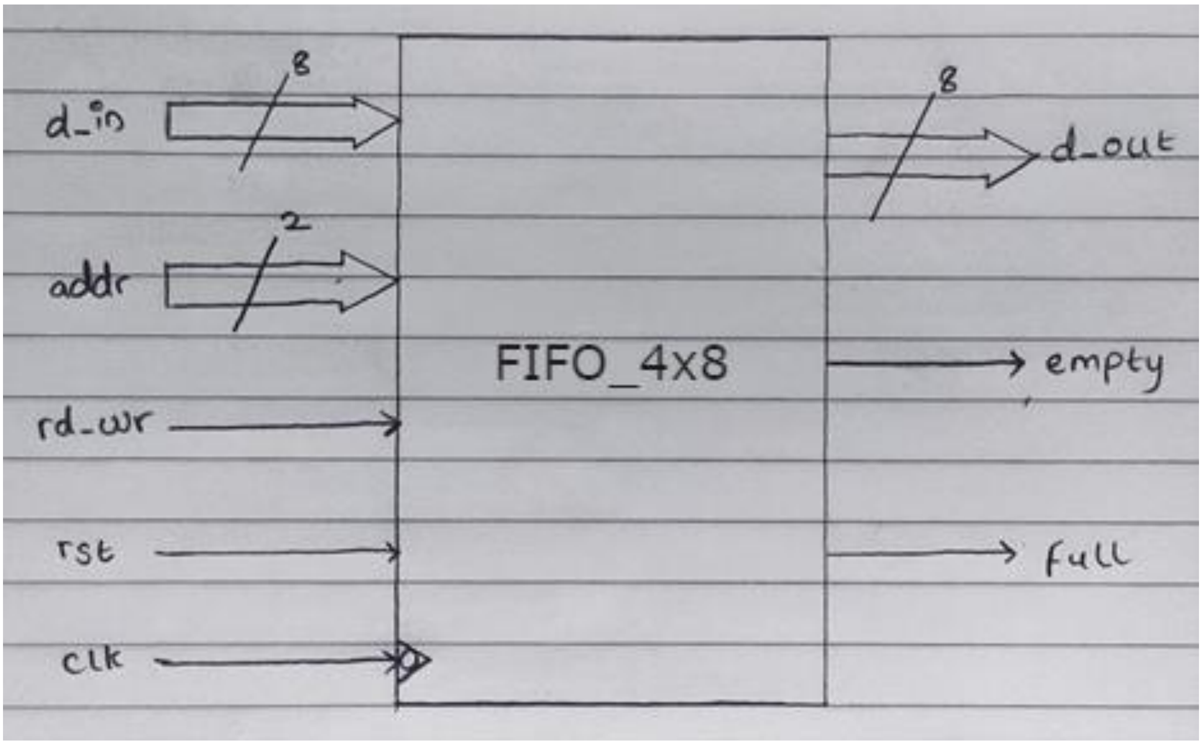
Class	:	
Batch	:	
ABC ID	:	
Roll. No	:	
Assignment No.	:	A.4
Assignment Name	:	FIFO (32-bit , organized as 4 X 8 , BYTE Addressable)
Date Of Performance	:	

BLOCK DIAGRAM



FUNCTION TABLE

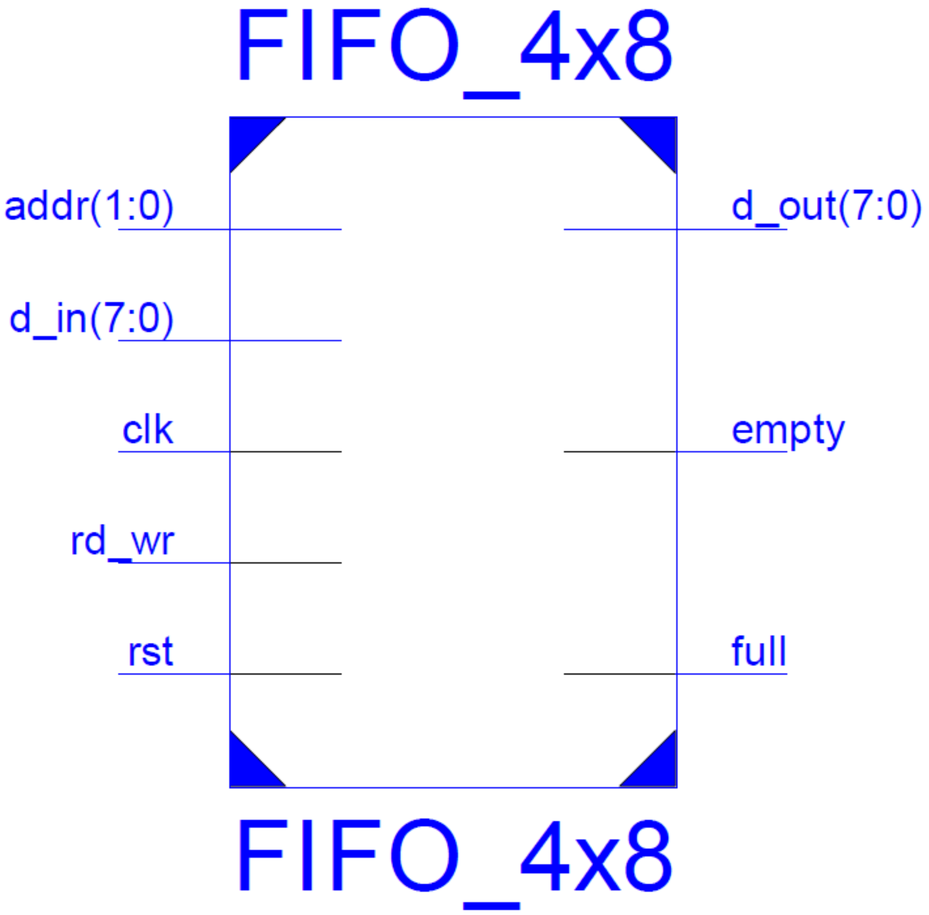
rst	clk	addr		rd_wr	d_out	empty	full
		A_1	A_0				
1	x	x	x	x	$(00)_{16}$	1	0
0	↓	0	0	0	mem_0	0	1
0	↓	0	1	0	mem_1	0	1
0	↓	1	0	0	mem_2	0	1
0	↓	1	1	0	mem_3	0	1
0	↓	0	0	1	mem_0	0	0
0	↓	0	1	1	mem_1	0	0
0	↓	1	0	1	mem_2	0	0
0	↓	1	1	1	mem_3	0	1

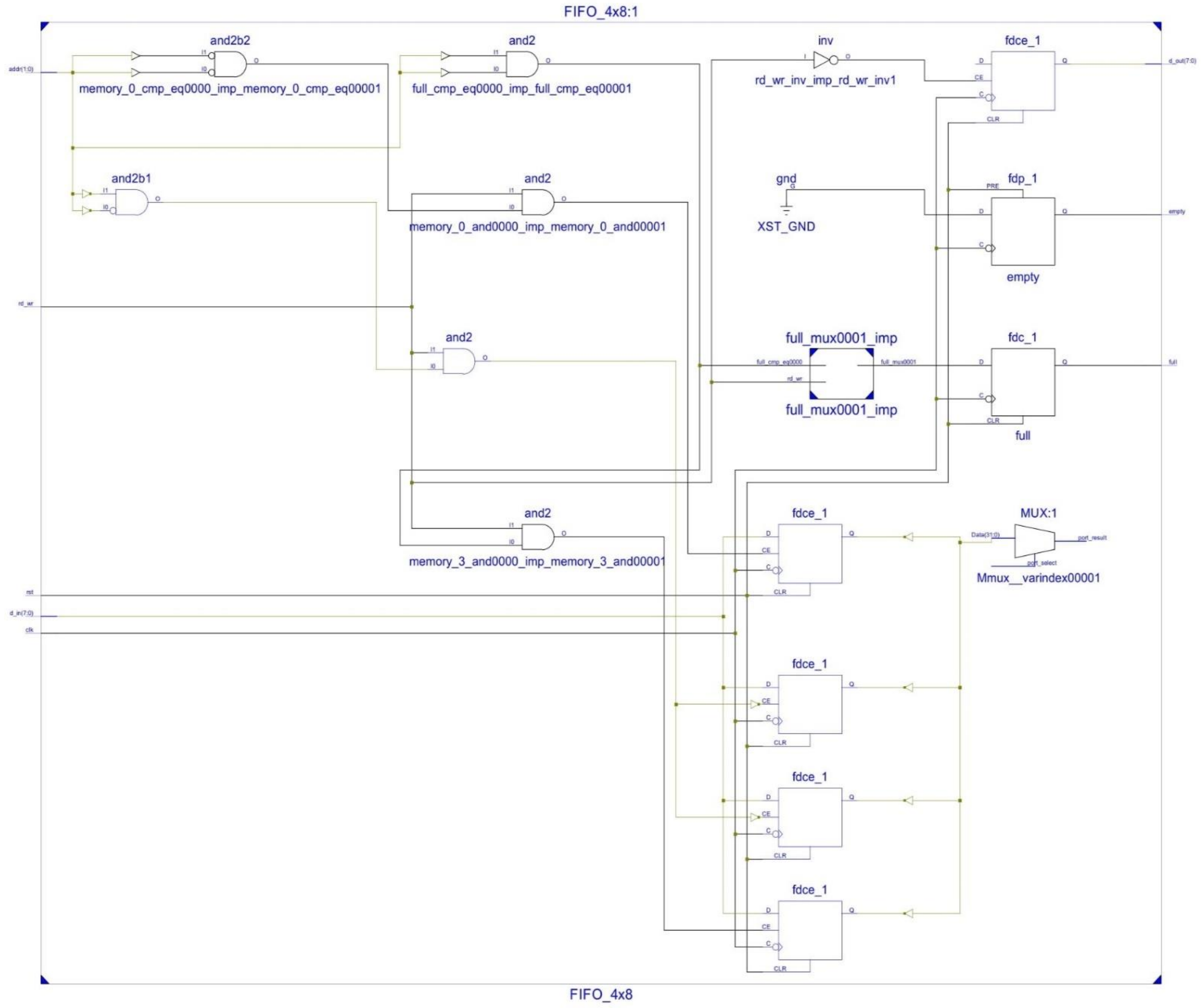
MAIN VHDL MODEL (MVM)

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

entity FIFO_4x8 is
   Port ( rst : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          addr : in  STD_LOGIC_VECTOR (1 downto 0) := "00";
          d_in : in  STD_LOGIC_VECTOR (7 downto 0);
          rd_wr : in  STD_LOGIC;
          d_out : out  STD_LOGIC_VECTOR (7 downto 0) := "00000000";
          empty : out  STD_LOGIC := '1';
          full : out  STD_LOGIC := '0');
end FIFO_4x8;

architecture FIFO_4x8_arch of FIFO_4x8 is

TYPE mem IS ARRAY(3 DOWNTO 0) OF STD_LOGIC_VECTOR (7 DOWNTO 0);
SIGNAL memory : mem := (others=>(others=>'0'));

begin

   PROCESS(rst, clk, addr, d_in, rd_wr)

            begin

               if rst = '1' then
                        d_out <= "00000000";
                                      empty <= '1';
                                      full <= '0';
                                      memory <= (others=>(others=>'0'));

     elsif falling_edge(clk) then

        case rd_wr is

                                        when '0' =>

                                                 d_out <= memory(conv_integer(addr));
                                                        empty <= '0';
                                          full <= '1';

                                        when others =>

                                                 memory(conv_integer(addr)) <= d_in;
                                                        empty <= '0';
                                                        if addr = "11" then
```

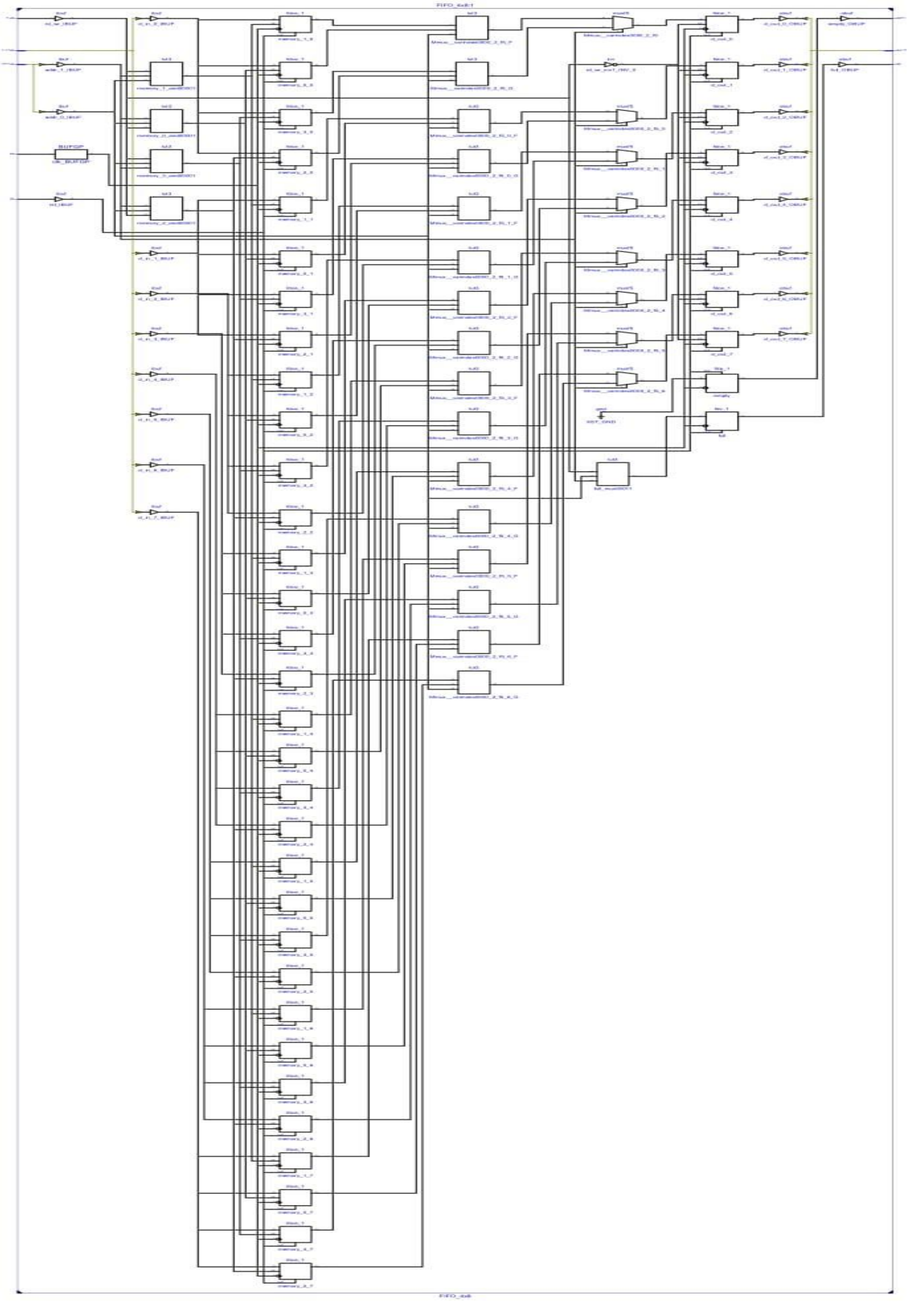
```
                                                            full <= '1';
                                                                       else
                                                            full <= '0';
                                                                       end if;
          end case;
        end if;
      end process;

end FIFO_4x8_arch;
```

RTL SCHEMATIC:




FIFO_4x8:1
addr(1:0)
and2b2
memory_0_cmp_eq0000_imp_memory_0_cmp_eq00001
and2
full_cmp_eq0000_imp_full_cmp_eq00001
inv
rd_wr_inv_imp_rd_wr_inv1
fdce_1
d_out(7:0)
and2b1
and2
memory_0_and0000_imp_memory_0_and00001
gnd
XST_GND
fdp_1
empty
empty
rd_wr
and2
full_mux0001_imp
full_cmp_eq0000
rd_wr
full_mux0001
full_mux0001_imp
fdc_1
full
full
and2
memory_3_and0000_imp_memory_3_and00001
fdce_1
MUX:1
Data(31:0)
port_result
port_select
Mmux__varindex00001
rst
d_in(7:0)
clk
fdce_1
fdce_1
fdce_1
FIFO_4x8

TECHNOLOGY SCHEMATIC



SYNTHESIS REPORT

a) Device Utilization Summary:

```
=========================================================================
*                        Final Report                        *
=========================================================================
Final Results
RTL Top Level Output File Name     : FIFO_4x8.ngr
Top Level Output File Name         : FIFO_4x8
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 23

Cell Usage :
# BELS                             : 32
#      GND                         : 1
#      INV                         : 2
#      LUT3                        : 21
#      MUXF5                       : 8
# FlipFlops/Latches                : 42
#      FDC_1                       : 1
#      FDCE_1                      : 40
#      FDP_1                       : 1
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 22
#      IBUF                        : 12
#      OBUF                        : 10
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 3s250epq208-5

 Number of Slices:                      26  out of   2448     1%
 Number of Slice Flip Flops:            40  out of   4896     0%
 Number of 4 input LUTs:                23  out of   4896     0%
 Number of IOs:                         23
 Number of bonded IOBs:                 23  out of    158    14%
    IOB Flip Flops:                      2
 Number of GCLKs:                        1  out of     24     4%
```

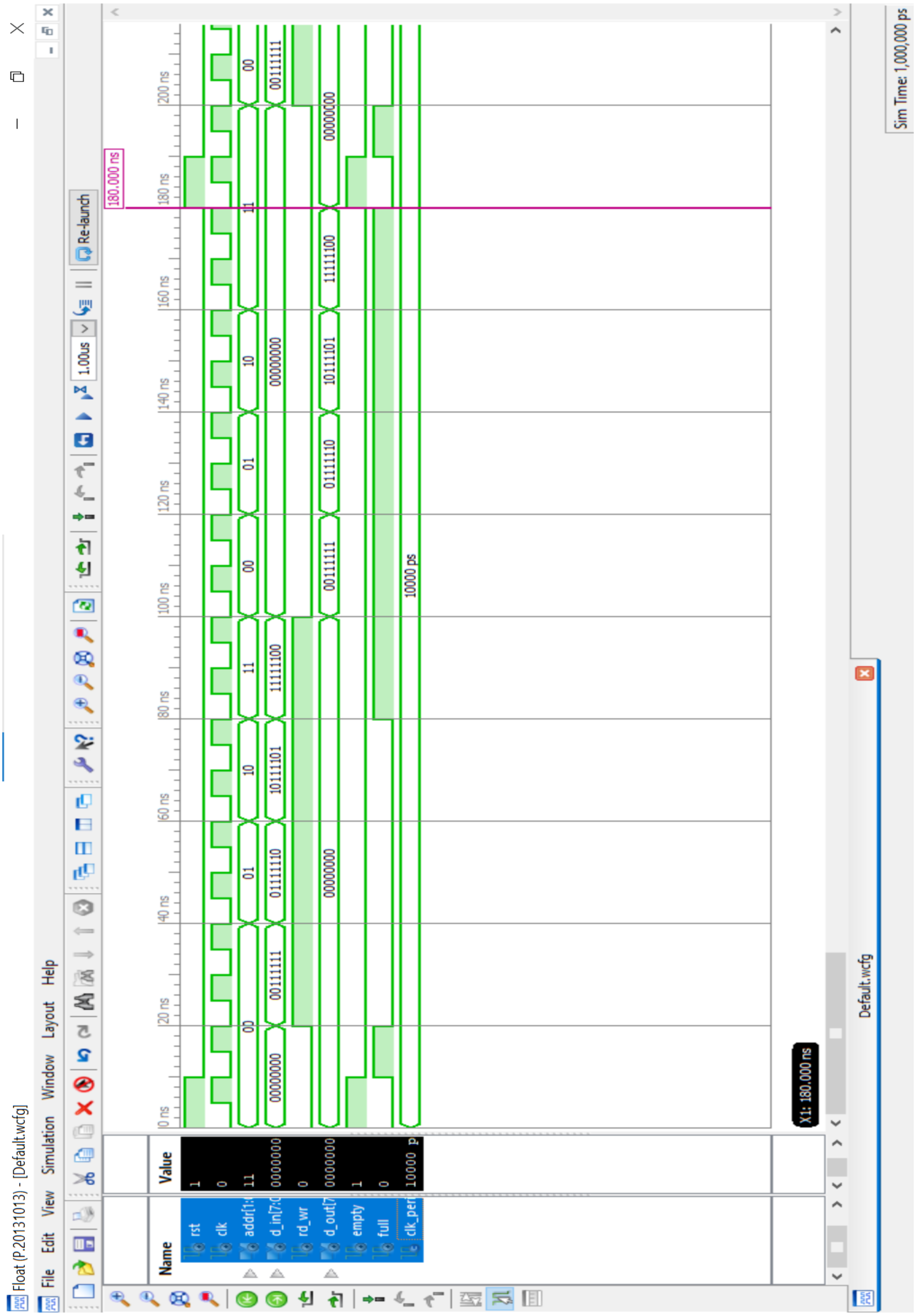
b) TIMING REPORT:

```
NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:
------------------
-----------------------------------+------------------------+-------+
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 42    |
-----------------------------------+------------------------+-------+

Asynchronous Control Signals Information:
----------------------------------------
-----------------------------------+------------------------+-------+
Control Signal                     | Buffer(FF name)        | Load  |
-----------------------------------+------------------------+-------+
rst_inv(rst_inv1_INV_0:O)          | NONE(d_out_0)          | 42    |
-----------------------------------+------------------------+-------+

Timing Summary:
---------------
Speed Grade: -5

   Minimum period: 2.098ns (Maximum Frequency: 476.644MHz)
   Minimum input arrival time before clock: 3.955ns
   Maximum output required time after clock: 4.040ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)
```

TESTBENCH VHDL MODEL (TVM)

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;

ENTITY FIFO_4x8_tb IS
END FIFO_4x8_tb;

ARCHITECTURE behavior OF FIFO_4x8_tb IS

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

   COMPONENT FIFO_4x8
   PORT(
        rst : IN  std_logic;
        clk : IN  std_logic;
        addr : IN  std_logic_vector(1 downto 0);
        d_in : IN  std_logic_vector(7 downto 0);
        rd_wr : IN  std_logic;
        d_out : OUT  std_logic_vector(7 downto 0);
        empty : OUT  std_logic;
        full : OUT  std_logic
       );
   END COMPONENT;


  --Inputs
  signal rst : std_logic := '0';
  signal clk : std_logic := '1';
  signal addr : std_logic_vector(1 downto 0) := (others => '0');
  signal d_in : std_logic_vector(7 downto 0) := (others => '0');
  signal rd_wr : std_logic := '0';

          --Outputs
  signal d_out : std_logic_vector(7 downto 0);
  signal empty : std_logic;
  signal full : std_logic;

  -- Clock period definitions
  constant clk_period : time := 10 ns;

BEGIN

          -- Instantiate the Unit Under Test (UUT)
  uut: FIFO_4x8 PORT MAP (
        rst => rst,
        clk => clk,
        addr => addr,
        d_in => d_in,
        rd_wr => rd_wr,
        d_out => d_out,
        empty => empty,
        full => full
       );

  -- Clock process definitions
  clk_process :process
  begin
                clk <= not(clk);
                wait for clk_period/2;
  end process;


  -- Stimulus process
  stim_proc: process
  begin

                rd_wr <= '1';
                for address in 0 to 3 loop
                      addr <= std_logic_vector(to_unsigned(address, 2));
                      d_in <= std_logic_vector(to_unsigned(63*(address + 1), 8));
                      wait for clk_period*2;
                end loop;

                d_in <= std_logic_vector(to_unsigned(0, 8));

                rd_wr <= '0';
                for address in 0 to 3 loop
                      addr <= std_logic_vector(to_unsigned(address, 2));
                      wait for clk_period*2;
                end loop;

  end process;

stim_proc_reset :PROCESS
                  BEGIN
                   wait for clk_period*12;
        rst <= '1';
                   wait for clk_period*2;
                   rst <= '0';
                   wait ;

                      end process;

END;
```

ISIM WAVEFORMS



PIN-LOCKING REPORT

```
# PlanAhead Generated physical constraints

NET "d_in[7]" LOC = P165;       #sw4-0
NET "d_in[6]" LOC = P167;       #sw4-1
NET "d_in[5]" LOC = P163;       #sw4-2
NET "d_in[4]" LOC = P164;
NET "d_in[3]" LOC = P161;
NET "d_in[2]" LOC = P162;
NET "d_in[1]" LOC = P160;
NET "d_in[0]" LOC = P153;       #sw4-7
NET "d_out[7]" LOC = P179;      #sw3-0
NET "d_out[6]" LOC = P180;      #sw3-1
NET "d_out[5]" LOC = P177;
NET "d_out[4]" LOC = P178;
NET "d_out[3]" LOC = P152;
NET "d_out[2]" LOC = P168;
NET "d_out[1]" LOC = P171;
NET "d_out[0]" LOC = P172;      #sw3-7
NET "clk" LOC = P132;
NET "rst" LOC = P204;           #k0
NET "rd_wr" LOC = P184;                 #sw2-6
NET "empty" LOC = P199;#sw1-6
NET "full" LOC = P196;   #sw1-7
```

CONCLUSION

Thus, we have:

1) Modeled a 4x8 FIFO using Behavioral Modeling Style.
2) Observed following Schematics: **RTL** & **Technology Schematics** generated **Post-Synthesis**.
3) Interpreted **Device Utilization Summary** in terms of LUTs, SLICES, IOBs, Multiplexers &D FFs used out of the available device resources.
4) Interpreted the TIMING Report in terms of Maximum combinational delay as indicative of the Maximum Operating Frequency.
5) Written a TESTBENCH to verify the functionality of 4x8 FIFO & verified the functionality as per the FUNCTION-TABLE, by observing ISIM Waveforms.
6) Used PlanAhead Editor for pin-locking.
7) Prototyped the FPGA **XC3S250EPQ208-5** to realize 4x8 FIFO & verified its operation by giving suitable input combinations.