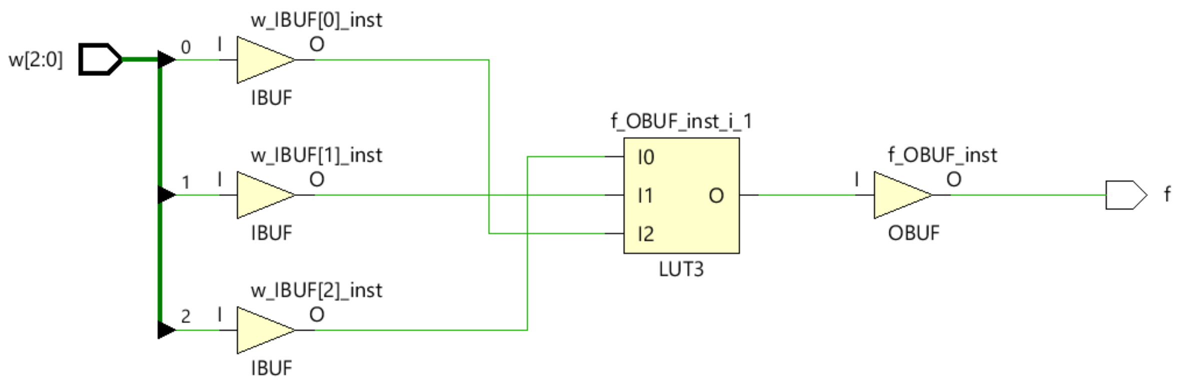
Exercise 1



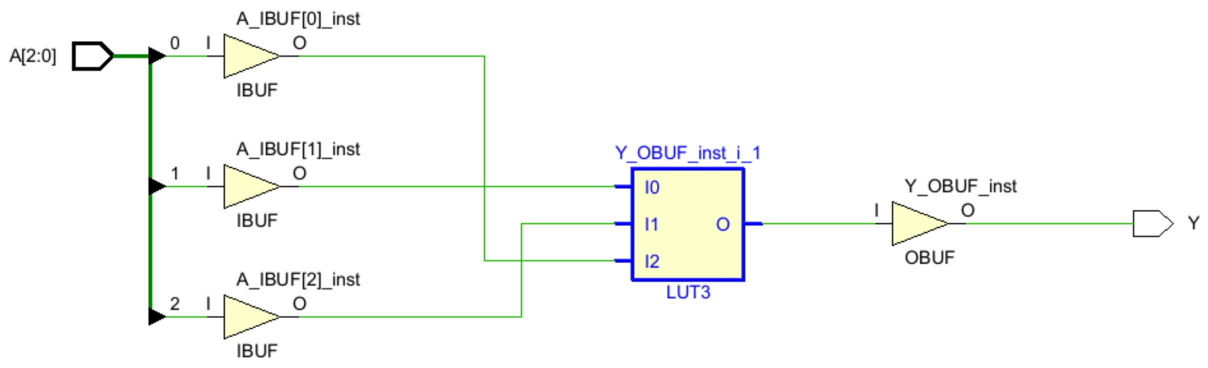
The LUT equation is: O = I0 & I2 + !I0 & I1 + !I1 & !I2

Equation Breakdown:

	Partial Results			Final Result
Partial Equ.	I0 & I2	!I0 & I1	!I1 & !I2	I0 & I2 + !I0 & I1 + !I1 & !I2
000	0	0	1	1
001	0	0	0	0
010	0	1	0	1
011	0	1	0	1
100	0	0	1	1
101	1	0	0	1
110	0	0	0	0
111	1	0	0	1

This LUT equation uses three separate AND statements (all ORed together) to build a result. Each partial result has 2 cases where they are set and between each partial result, no ‘1’s overlap.

Exercise 2



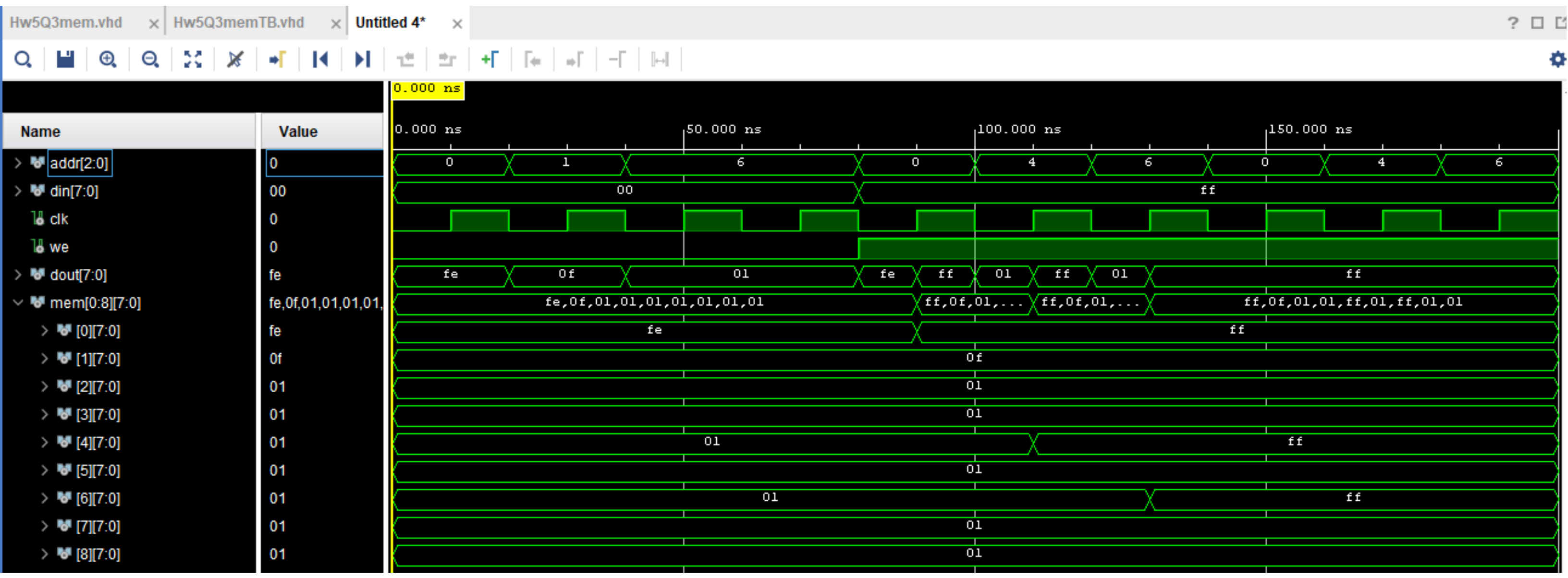
The LUT equation is: O = I0 & !I1 & !I2 + !I0 & I1 + I1 & I2

Equation Breakdown:

	Partial Results			**Final Result**
Partial Equ.	I0 & !I1 & !I2	!I2 + !I0	I1 & I2	O = I0 & !I1 & !I2 + !I0 & I1 + I1 & I2
000	0	0	0	0
001	0	0	0	0
010	0	1	0	1
011	0	0	0	0
100	1	0	0	1
101	0	0	1	1
110	0	0	0	0
111	0	0	1	1

This LUT equation uses three separate AND statements (all ORed together) to build a result. One partial equation is more complex than the other two.

Exercise 3



Name	Value
addr[2:0]	0
din[7:0]	00
clk	0
we	0
dout[7:0]	fe
mem[0:8][7:0]	fe,0f,01,01,01,01,
[0][7:0]	fe
[1][7:0]	0f
[2][7:0]	01
[3][7:0]	01
[4][7:0]	01
[5][7:0]	01
[6][7:0]	01
[7][7:0]	01
[8][7:0]	01

```
library ieee;
use ieee.std_logic_1164.all;
use work.Hw5Q3pkgSoln.all;

entity Hw5Q3memTB is
end Hw5Q3memTB;

architecture tb of Hw5Q3memTB is
  signal addr : std_logic_vector (a_width-1 downto 0) := (others => '0');
  signal din : std_logic_vector (d_width-1 downto 0) := (others => '0');
  signal clk, we : std_logic := '0';
  signal dout : std_logic_vector (d_width-1 downto 0) := (others => '0');
begin

  dut : entity work.Hw5Q3mem
    port map (addr => addr, din => din, clk => clk, we => we, dout => dout);

  clk <= not clk after clk_per/2;

  stimuli : process
  begin
    we <= '0';
    addr <= "000";
    wait for 10 ns;
    assert (x"FE" = dout) report "error in result. dout is " &
    to_hstring (dout) & " and should be = FE";
    wait for 10 ns;
```

```
-- code to read and check location 1
addr <= "001";
wait for 10 ns;
assert (x"FE" = dout) report "error in result. dout is " &
to_hstring (dout) & " and should be = FE";
wait for 10 ns;

-- code to read and check location 6
addr <= "110";
wait for 10 ns;
assert (x"FE" = dout) report "error in result. dout is " &
to_hstring (dout) & " and should be = FE";
wait for 10 ns;

wait until clk = '0';
we <= '1';
addr <= "000";
din <= x"FF";

wait until clk = '0';
addr <= "100";

wait until clk = '0';
addr <= "110";

-- code to read and check location 0
wait until clk = '0';
addr <= "000";
wait for 10 ns;
assert (x"FF" = dout) report "error in result. dout is " &
to_hstring (dout) & " and should be = FF";
wait for 10 ns;

-- read and check location 4
addr <= "100";
wait for 10 ns;
assert (x"FF" = dout) report "error in result. dout is " &
to_hstring (dout) & " and should be = FF";
wait for 10 ns;

-- read and check location 6
addr <= "110";
wait for 10 ns;
assert (x"FF" = dout) report "error in result. dout is " &
to_hstring (dout) & " and should be = FF";
wait for 10 ns;
```

```
      wait;
   end process;
end tb;



library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.Hw5Q3pkgSoln.all;

entity Hw5Q3mem is
   port (
   addr: in std_logic_vector (a_width -1 downto 0);
   din: in std_logic_vector (d_width -1 downto 0);
   clk, we: in std_logic;
   dout: out std_logic_vector (d_width -1 downto 0)
   );
end entity Hw5Q3mem;

architecture obehavior of Hw5Q3mem is

   signal mem : mem_type(0 to (a_width**2)-1)(d_width-1 downto 0) := (x"FE", x"0F",others => x"01");

begin

   process (clk, we) is begin
      if (rising_edge(clk) and we = '1') then
         mem(to_integer(unsigned(addr))) <= din;
      end if;
   end process;

   dout <= mem(to_integer(unsigned(addr)));

end obehavior;


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

package Hw5Q3pkgSoln is
constant a_width: positive := 3;
constant d_width: positive := 8;
-- add a constant d_width and set it to 8

type mem_type is array (natural range <>) of std_logic_vector;
constant clk_per: time := 20 ns;
end package Hw5Q3pkgSoln;
```