Ain Shams University
Faculty of Engineering
Electronics and Communication Department
ECE 342: Digital Circuits
3rd Year Communications

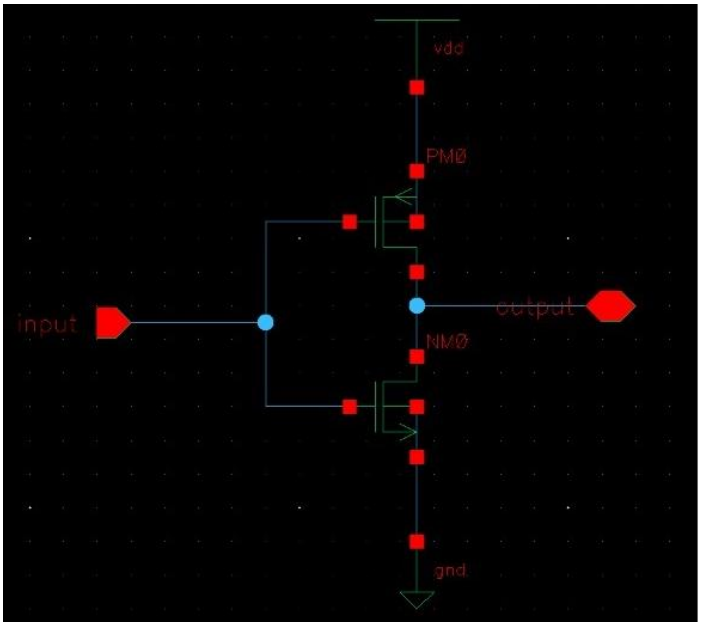
Arithmetic and Logical Unit Design Project

NAME	*CODE*
Moataz Mokhtar Said	*1701460*
Farid Mohammed Farid	*1700985*
Ahmed Ayman Ahmed	*1700037*
Mohamed Ashraf Farouk	*1701154*
Yumna Ahmed Mohammed	*1701692*
Youssef Ayman El-Rouby	*1701709*
Nour Mahmoud Galal	*1701587*
Abdelrahman Mohamed Anwar	*1700757*
Mahmoud Mohamed Hanafy	*1701360*
Salma Tarek El-Alfy	*1700620*

Contents

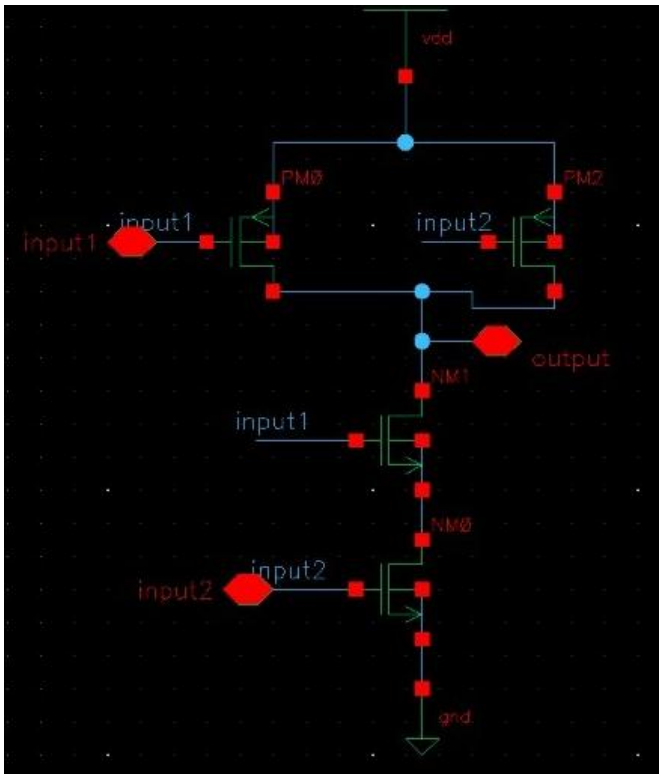
Circuit Schematics:

CMOS Inverter:

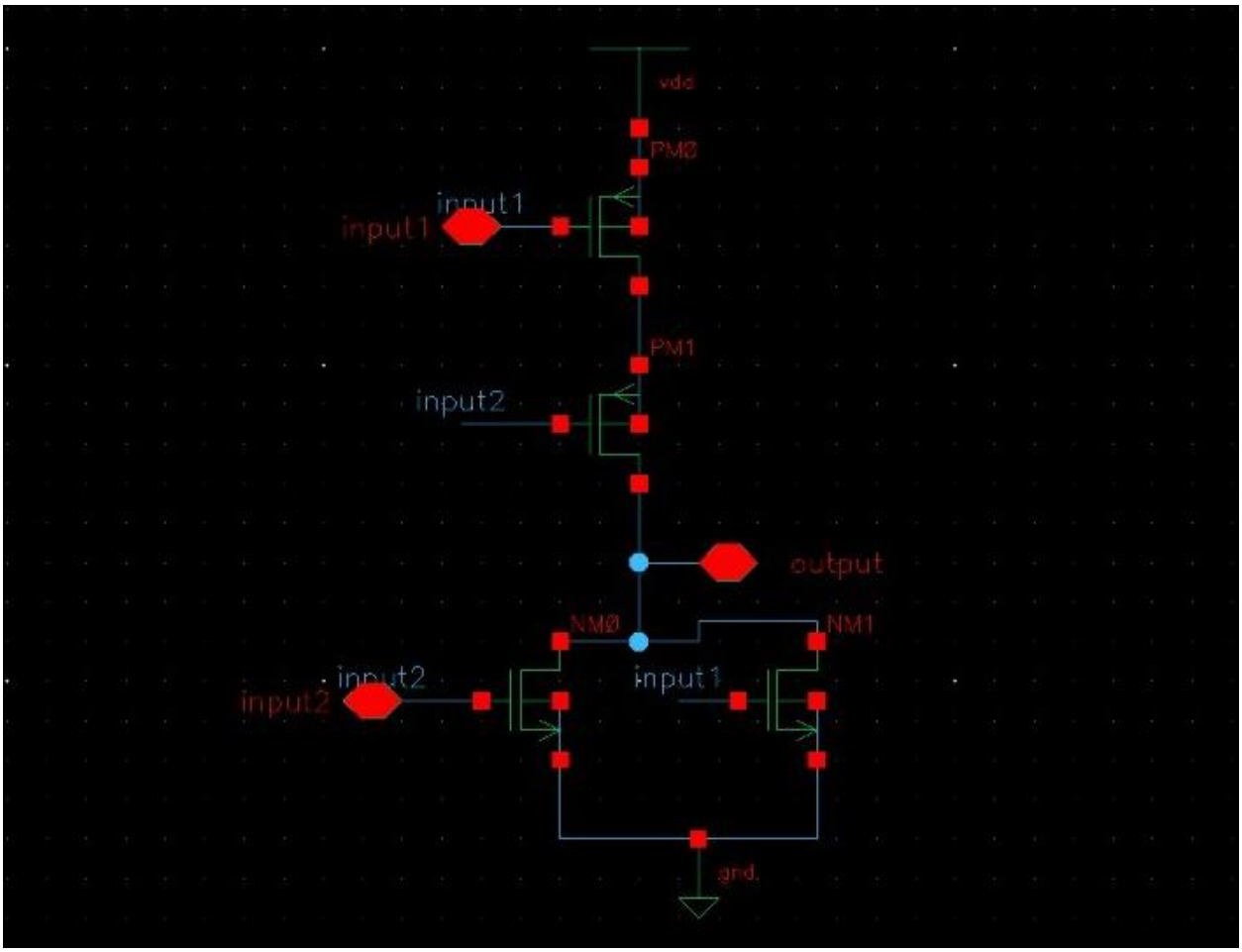


Reference sizing relative to CMOS inverter 2:1.

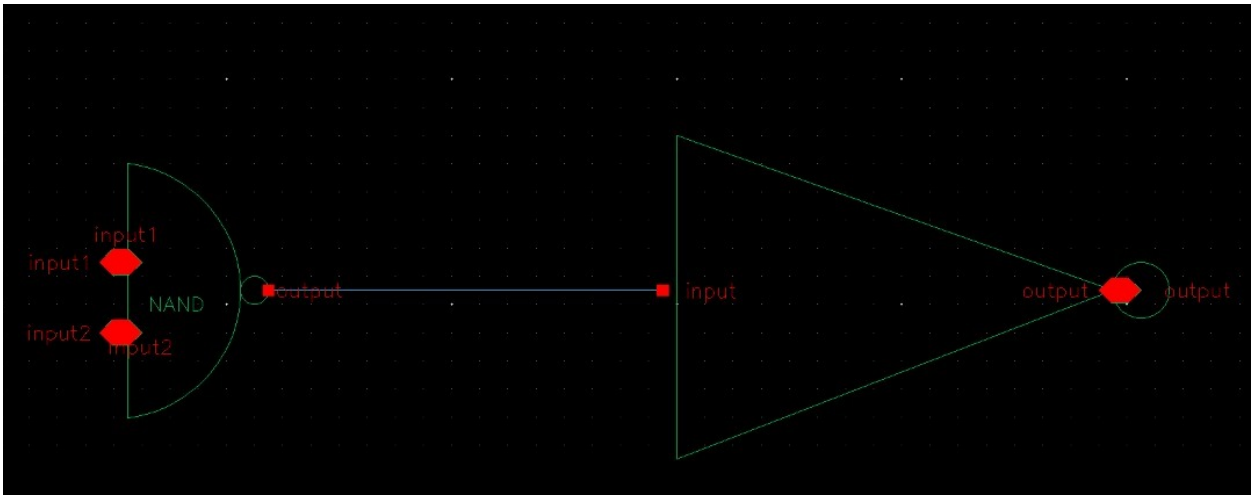
NAND Gate:



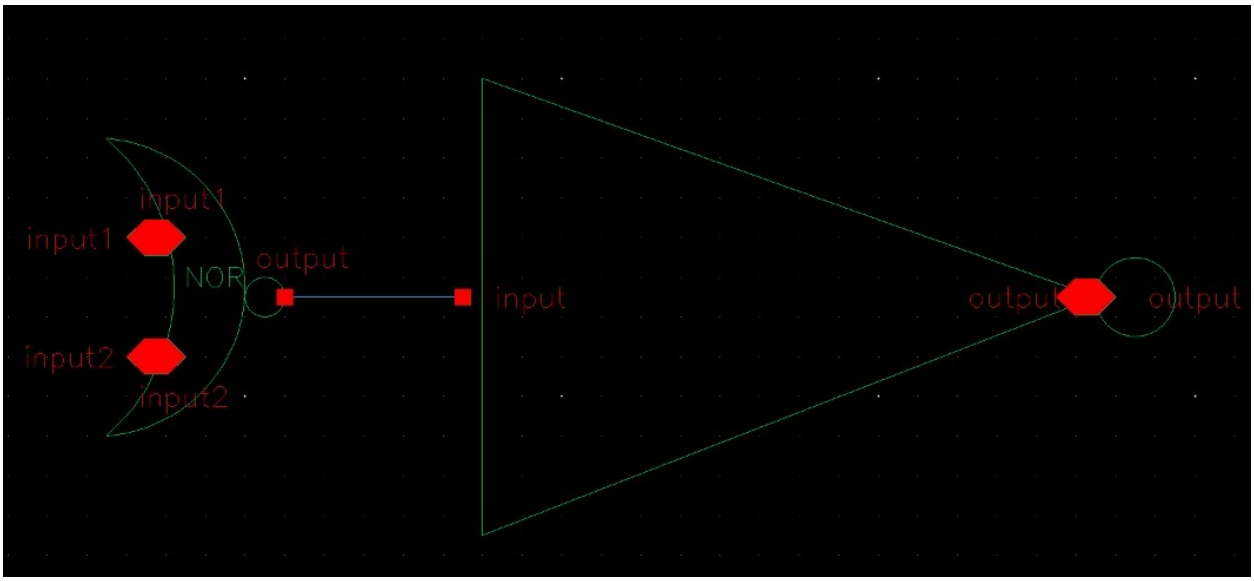
NOR Gate:



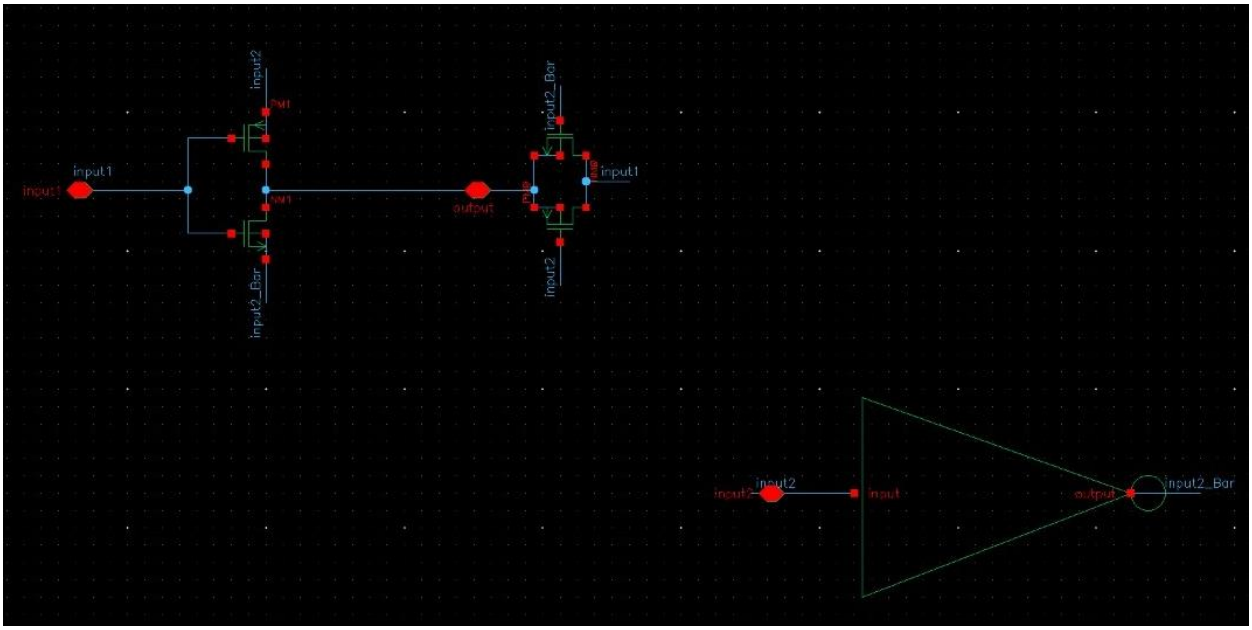
AND Gate:



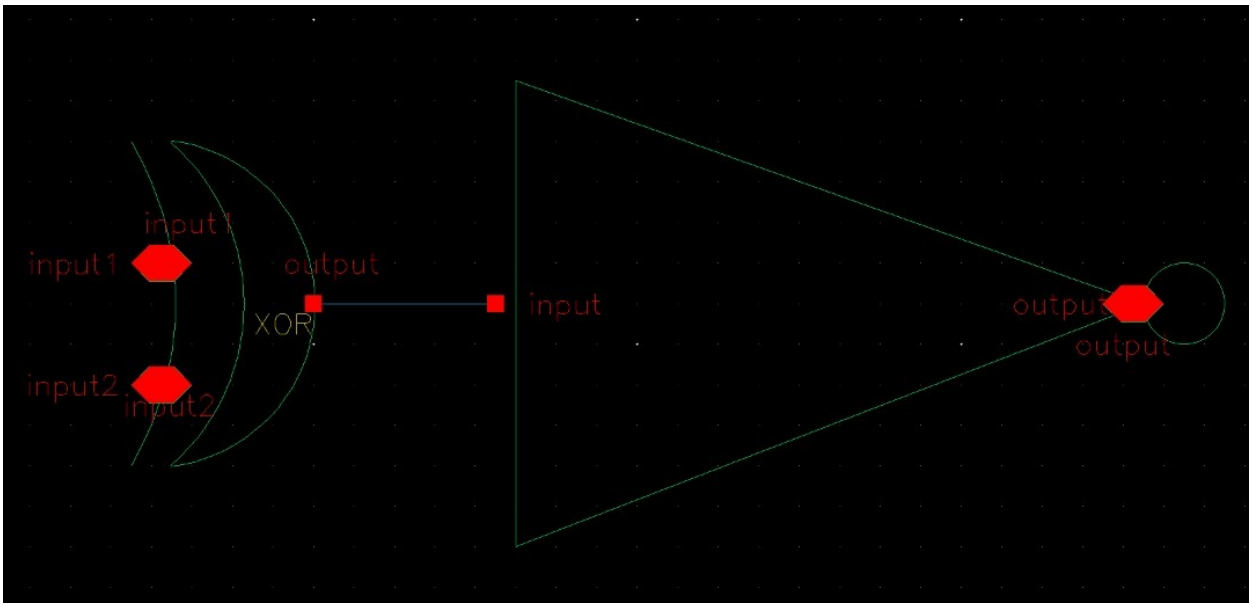
OR Gate:



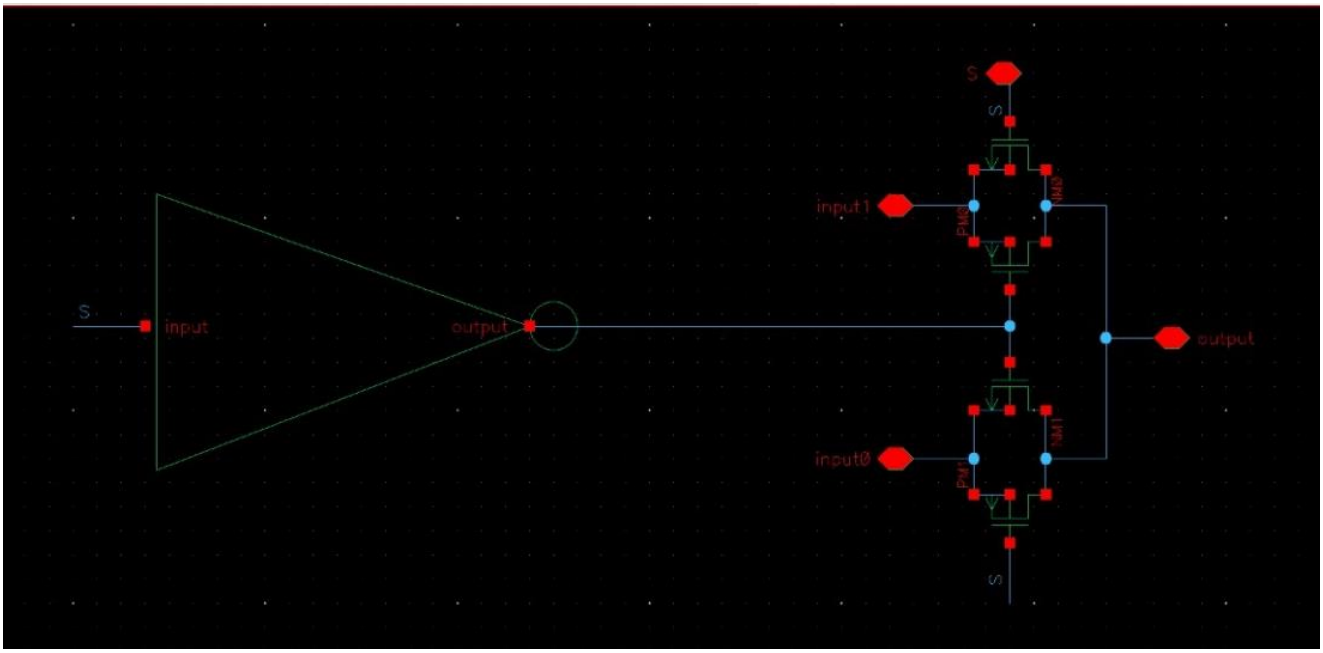
XOR Gate:



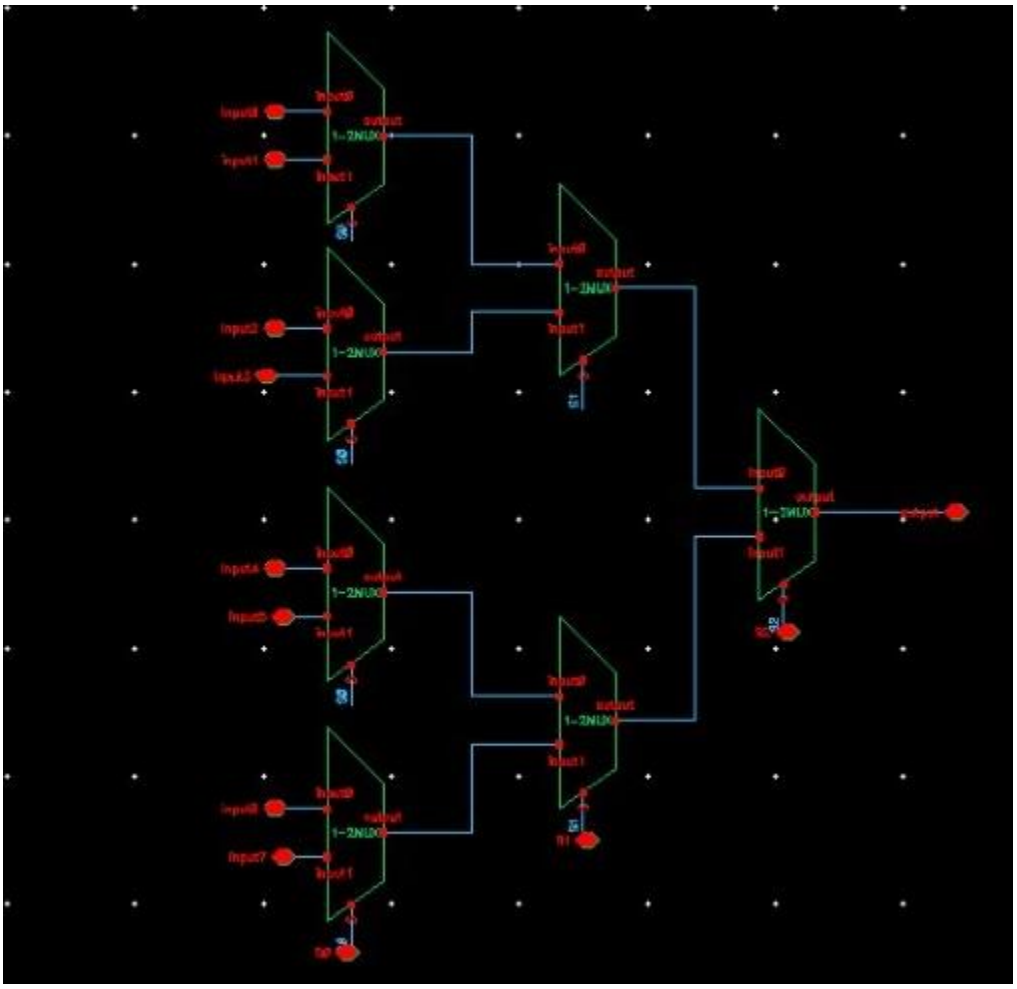
XNOR Gate:



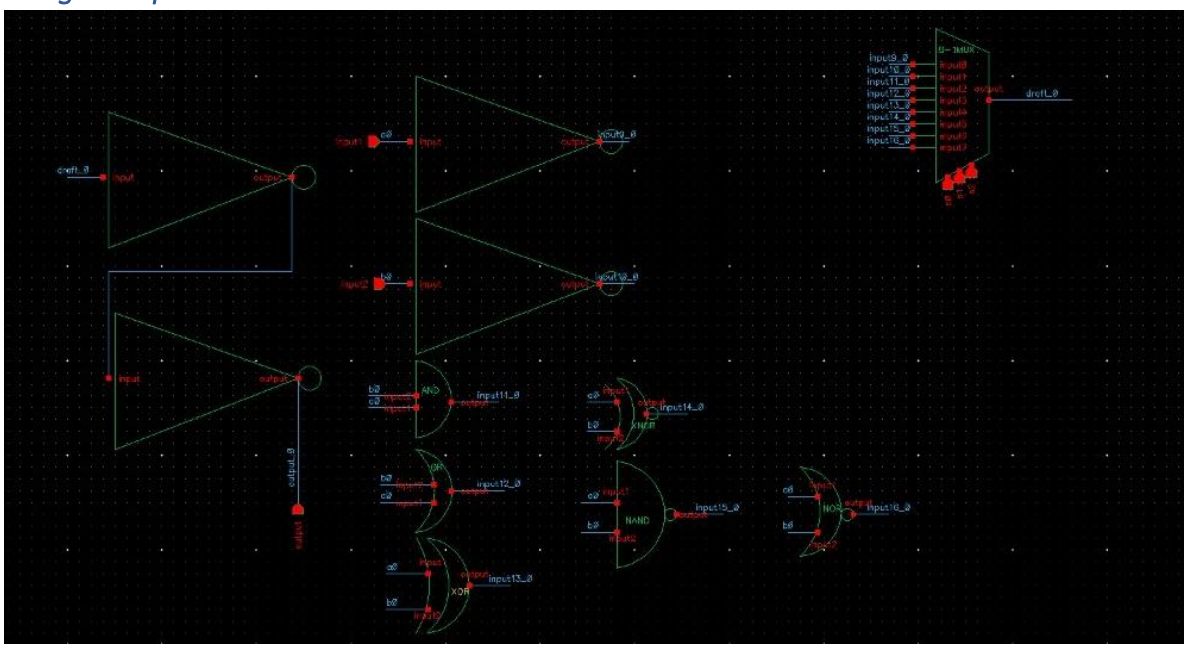
2-1 Multiplexer:



8-1 Multiplexer:

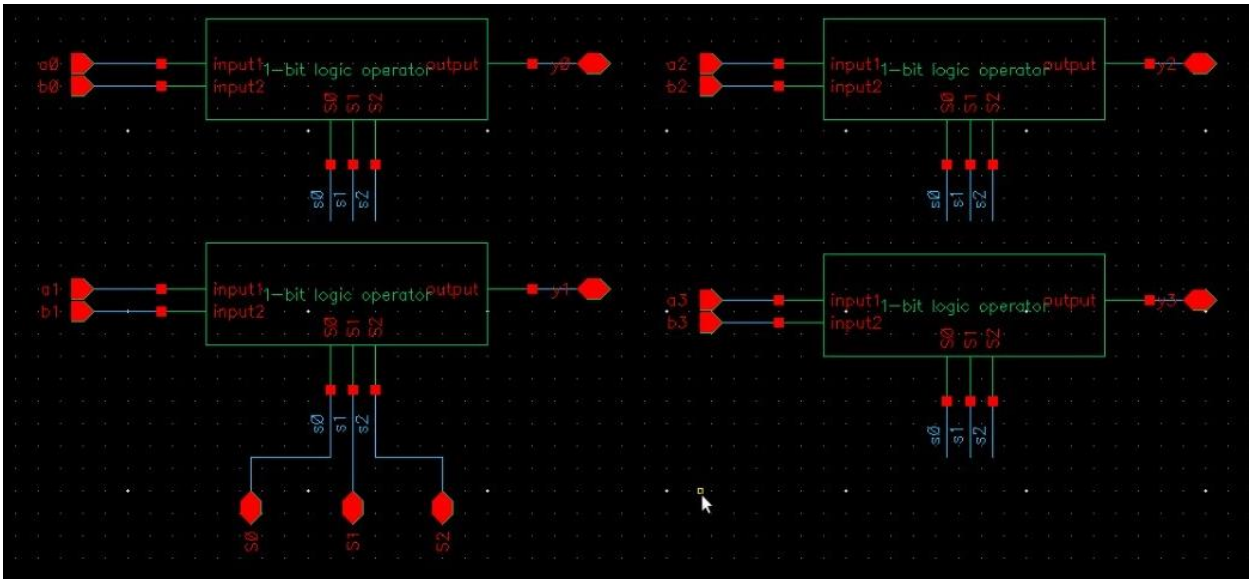


1-bit Logical Operator MUX:

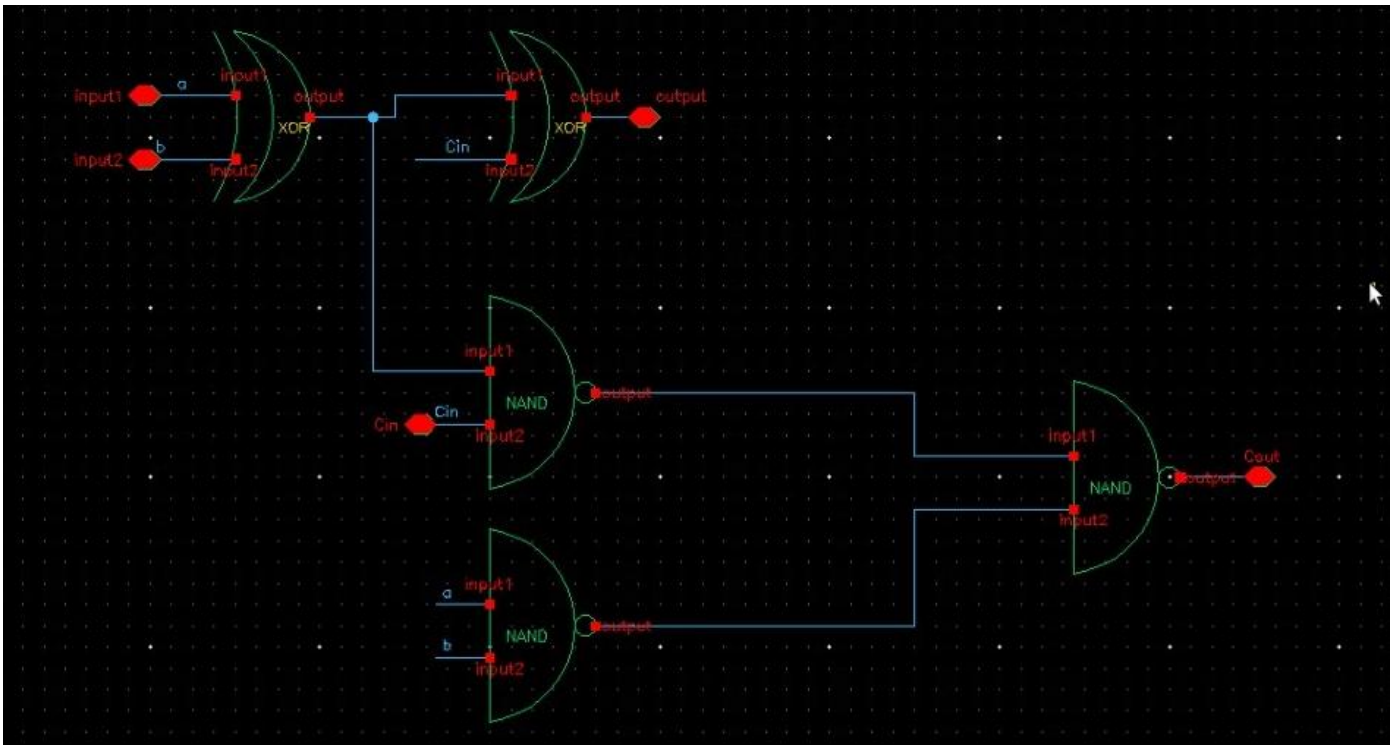


Note: We put 2 CMOS inverter after the output of the mux because our output wasn't rail to rail.

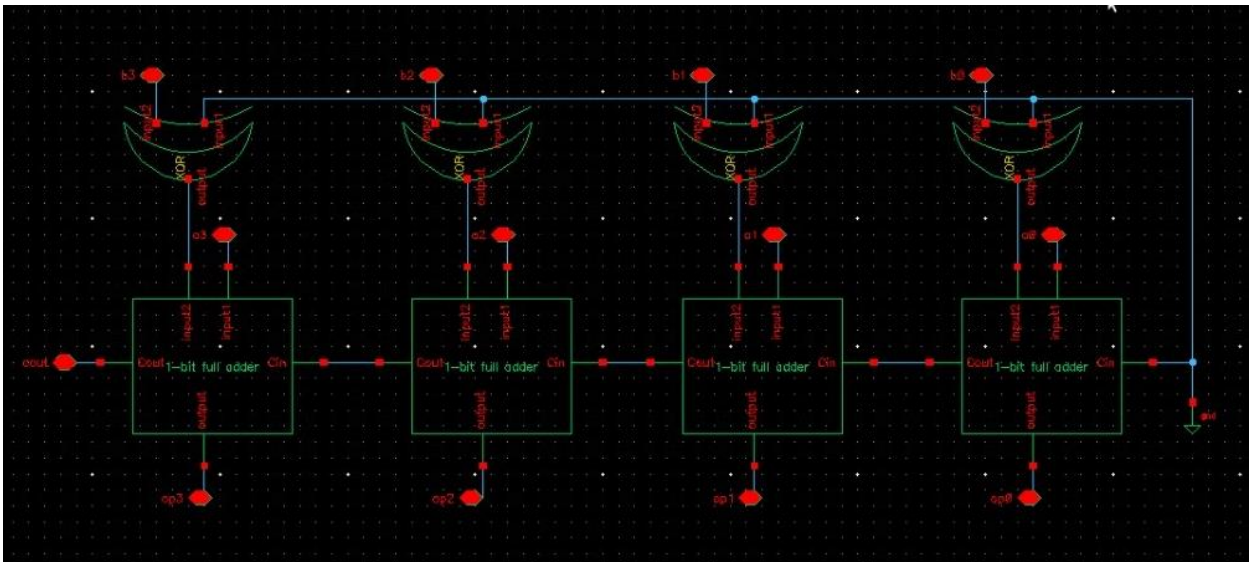
4-bit Logical Operator MUX:



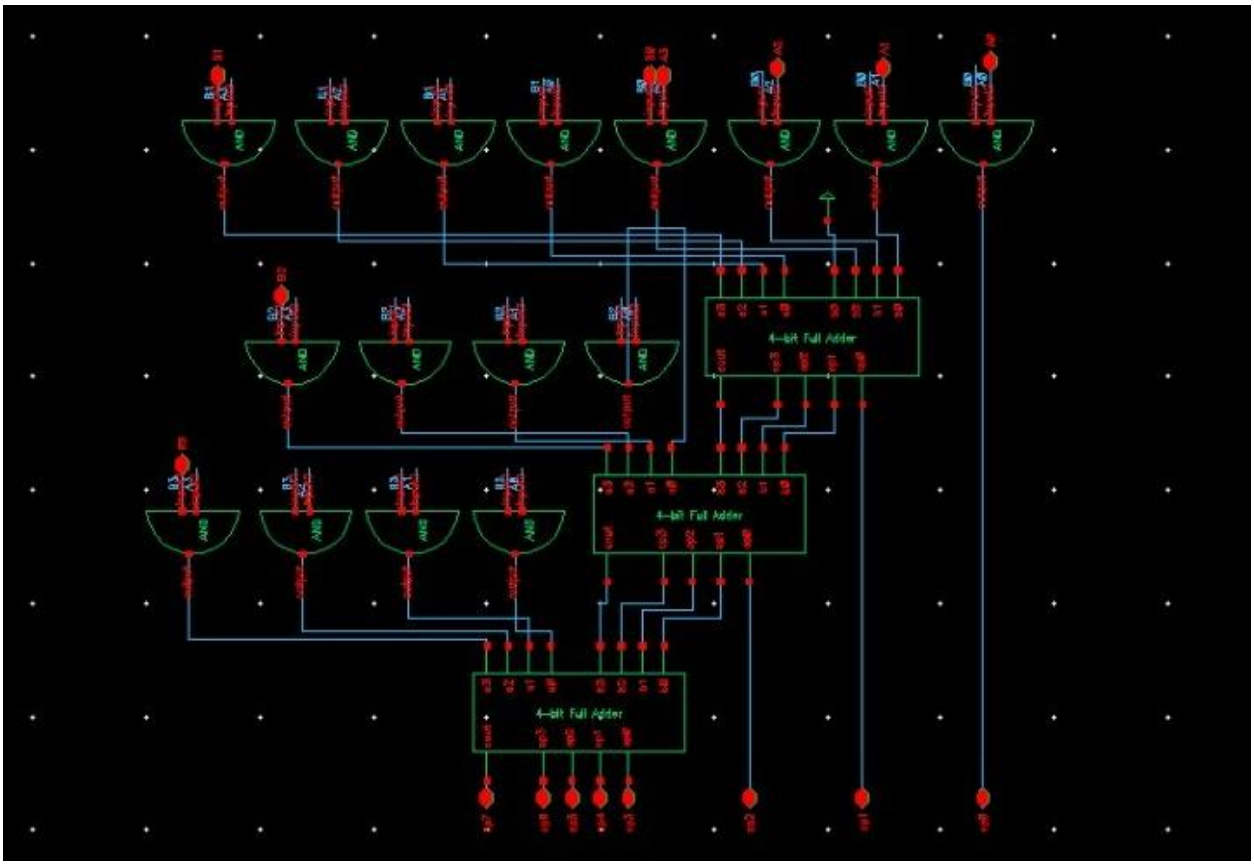
1-bit Adder:



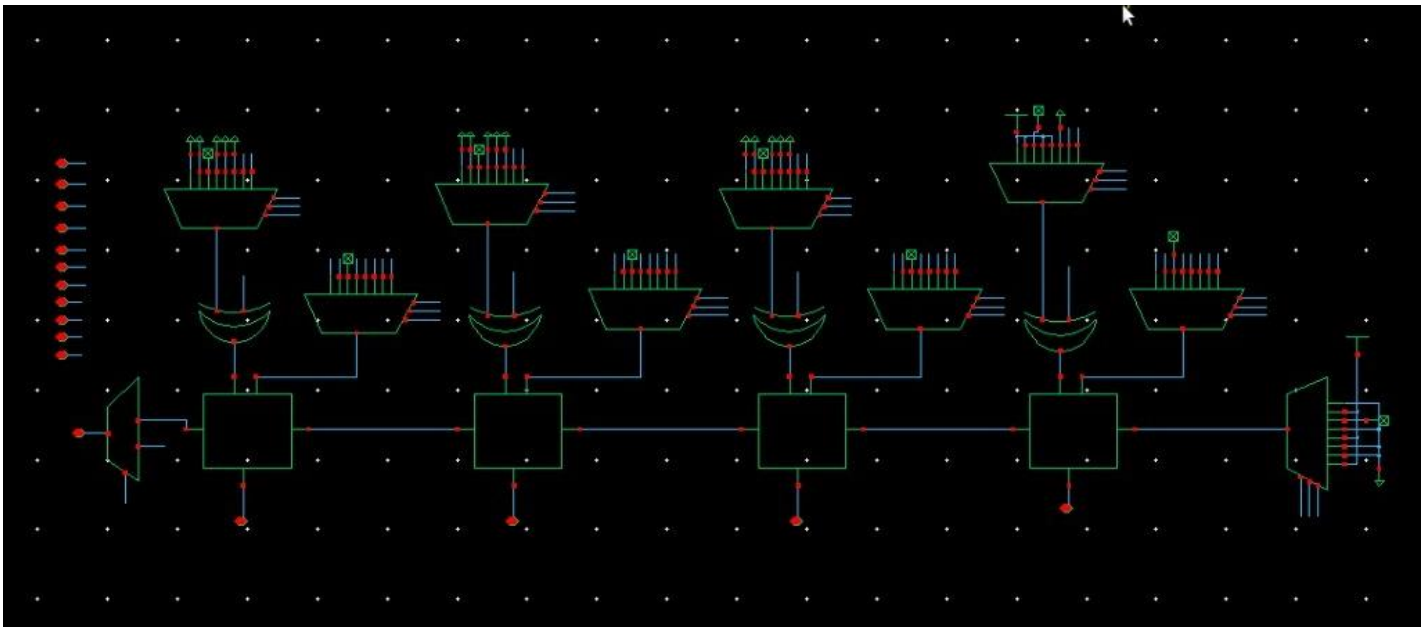
4-bit Adder:



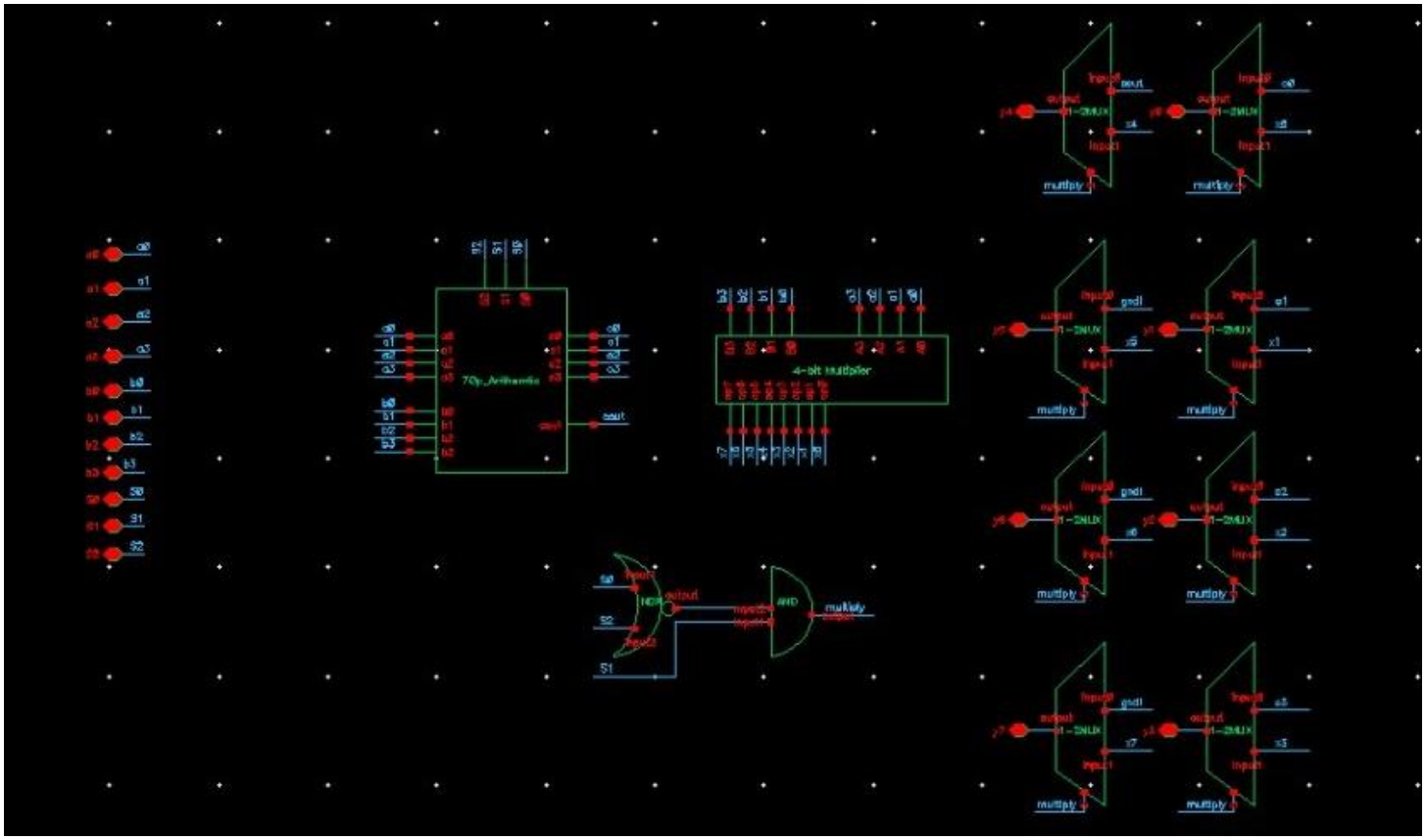
4-bit Multiplier:



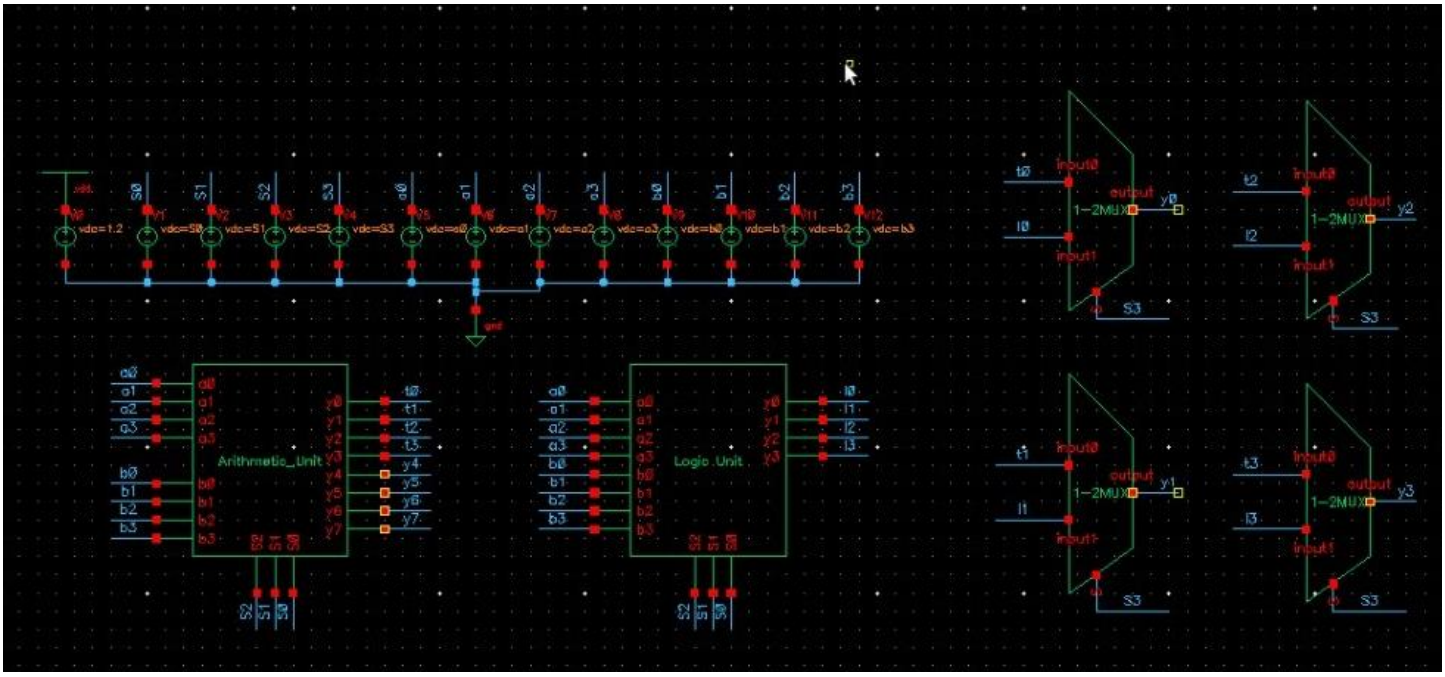
Arithmetic unit operations without multiplication:



Arithmetic unit:



Arithmetic and logical unit:



ALU Operations:

Arithmetic Operation
Input A = 9 (00001001) Input B=5 (00000101)
Output Y (8 bits) =(Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0)
DC-Analysis

0000
inc a

Test	Output	Nominal
Digital:Test1:1	VDC("/y0")	23.52u
Digital:Test1:1	VDC("/y1")	1.2
Digital:Test1:1	VDC("/y2")	29.12u
Digital:Test1:1	VDC("/y3")	1.2
Digital:Test1:1	VDC("/y4")	2.906u
Digital:Test1:1	VDC("/y5")	2.309u
Digital:Test1:1	VDC("/y6")	25.56p
Digital:Test1:1	VDC("/y7")	13.16p

0
1
0
1
0
0
0
0
Decimal = 10

0001
dec a

Test	Output	Nominal
Digital:Test1:1	VDC("/y0")	18.66u
Digital:Test1:1	VDC("/y1")	11.8u
Digital:Test1:1	VDC("/y2")	18.74u
Digital:Test1:1	VDC("/y3")	1.2
Digital:Test1:1	VDC("/y4")	2.351u
Digital:Test1:1	VDC("/y5")	2.309u
Digital:Test1:1	VDC("/y6")	25.56p
Digital:Test1:1	VDC("/y7")	13.16p

0
0
0
1
0
0
0
0
Decimal = 8

0010
a * b

Test	Output	Nominal
Digital:Test1:1	VDC("/y0")	1.2
Digital:Test1:1	VDC("/y1")	6.618u
Digital:Test1:1	VDC("/y2")	1.2
Digital:Test1:1	VDC("/y3")	1.2
Digital:Test1:1	VDC("/y4")	6.236u
Digital:Test1:1	VDC("/y5")	1.2
Digital:Test1:1	VDC("/y6")	5.646u
Digital:Test1:1	VDC("/y7")	2.905u

1
0
1
1
0
1
0
0
Decimal = 45

0011 *inc b*	Test	Output	Nominal
	Digital:Test1:1	VDC("/y0")	23.52u
	Digital:Test1:1	VDC("/y1")	1.2
	Digital:Test1:1	VDC("/y2")	1.199
	Digital:Test1:1	VDC("/y3")	650u
	Digital:Test1:1	VDC("/y4")	2.905u
	Digital:Test1:1	VDC("/y5")	2.309u
	Digital:Test1:1	VDC("/y6")	25.56p
	Digital:Test1:1	VDC("/y7")	13.16p

0
1
1
0
0
0
0
0
Decimal = 6

0100 *dec b*	Test	Output	Nominal
	Digital:Test1:1	VDC("/y0")	18.61u
	Digital:Test1:1	VDC("/y1")	11.87u
	Digital:Test1:1	VDC("/y2")	1.2
	Digital:Test1:1	VDC("/y3")	18.76u
	Digital:Test1:1	VDC("/y4")	2.4u
	Digital:Test1:1	VDC("/y5")	2.309u
	Digital:Test1:1	VDC("/y6")	25.56p
	Digital:Test1:1	VDC("/y7")	13.16p

0
0
1
0
0
0
0
0
Decimal = 4

0101 *transfer a*	Test	Output	Nominal
	Digital:Test1:1	VDC("/y0")	1.2
	Digital:Test1:1	VDC("/y1")	8.04u
	Digital:Test1:1	VDC("/y2")	338.1u
	Digital:Test1:1	VDC("/y3")	1.199
	Digital:Test1:1	VDC("/y4")	2.906u
	Digital:Test1:1	VDC("/y5")	2.309u
	Digital:Test1:1	VDC("/y6")	25.56p
	Digital:Test1:1	VDC("/y7")	13.16p

1
0
0
1
0
0
0
0
Decimal = 9

0110
$add\ a, b$

Test	Output	Nominal
Digital:Test1:1	VDC("/y0")	23.52u
Digital:Test1:1	VDC("/y1")	1.2
Digital:Test1:1	VDC("/y2")	1.199
Digital:Test1:1	VDC("/y3")	1.199
Digital:Test1:1	VDC("/y4")	2.906u
Digital:Test1:1	VDC("/y5")	2.309u
Digital:Test1:1	VDC("/y6")	25.56p
Digital:Test1:1	VDC("/y7")	13.16p

0
1
1
1
0
0
0
0
Decimal = 14

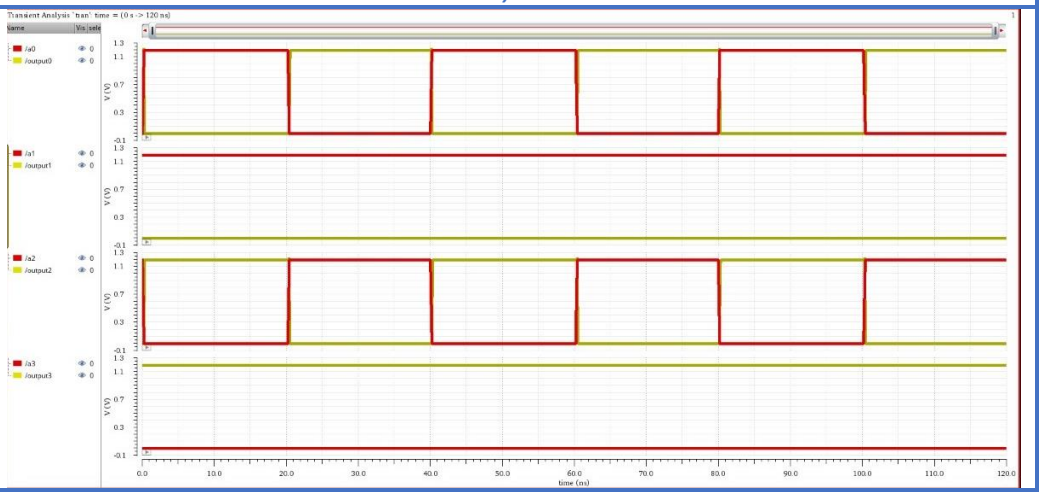
0111
$sub\ a, b$

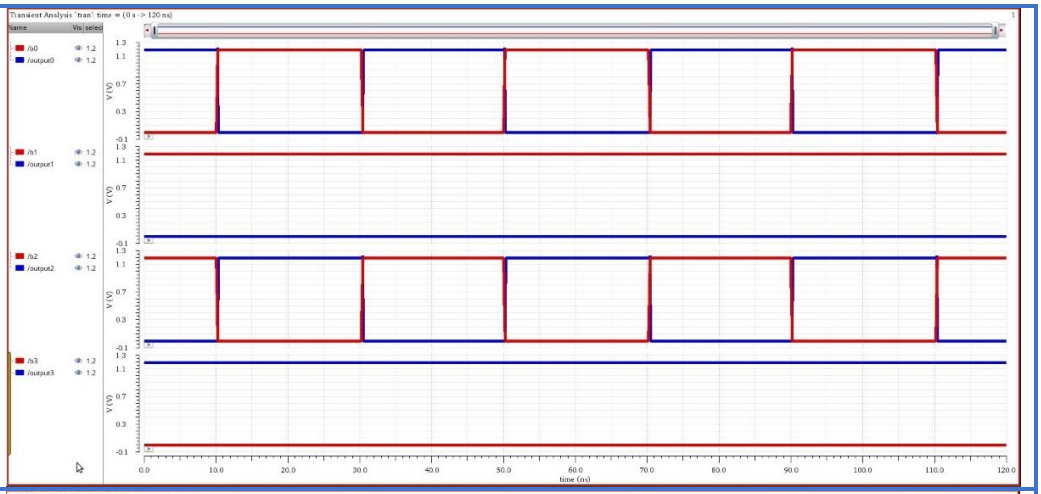
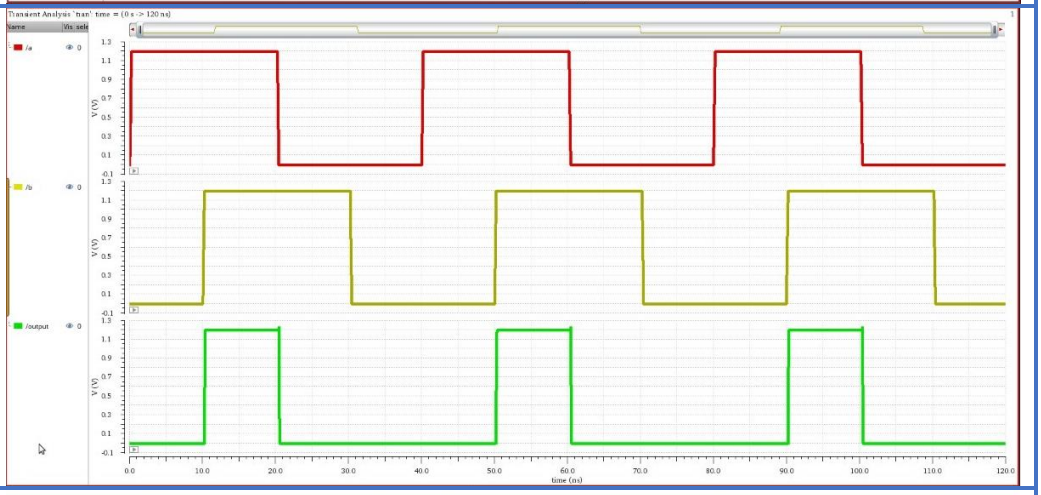
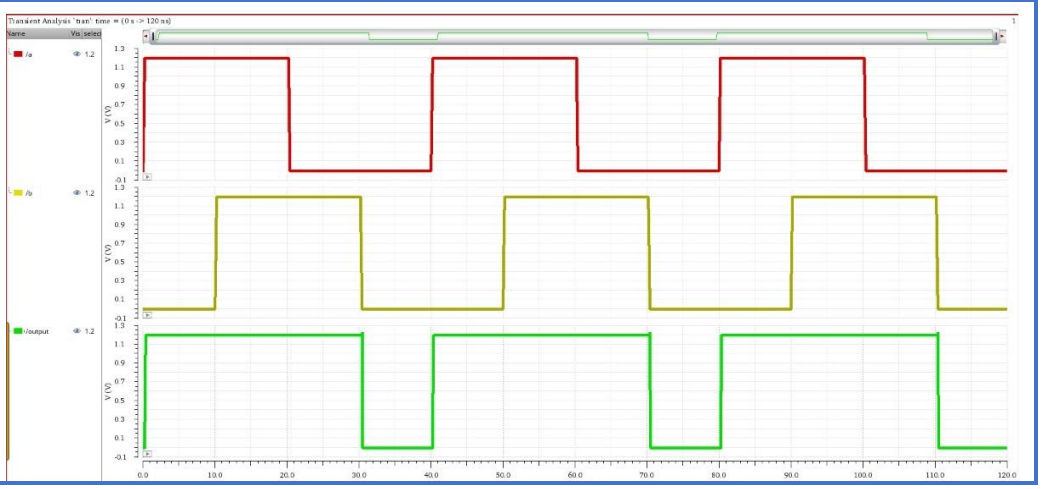
Test	Output	Nominal
Digital:Test1:1	VDC("/y0")	18.55u
Digital:Test1:1	VDC("/y1")	11.98u
Digital:Test1:1	VDC("/y2")	1.2
Digital:Test1:1	VDC("/y3")	23.34u
Digital:Test1:1	VDC("/y4")	2.471u
Digital:Test1:1	VDC("/y5")	2.309u
Digital:Test1:1	VDC("/y6")	25.56p
Digital:Test1:1	VDC("/y7")	13.16p

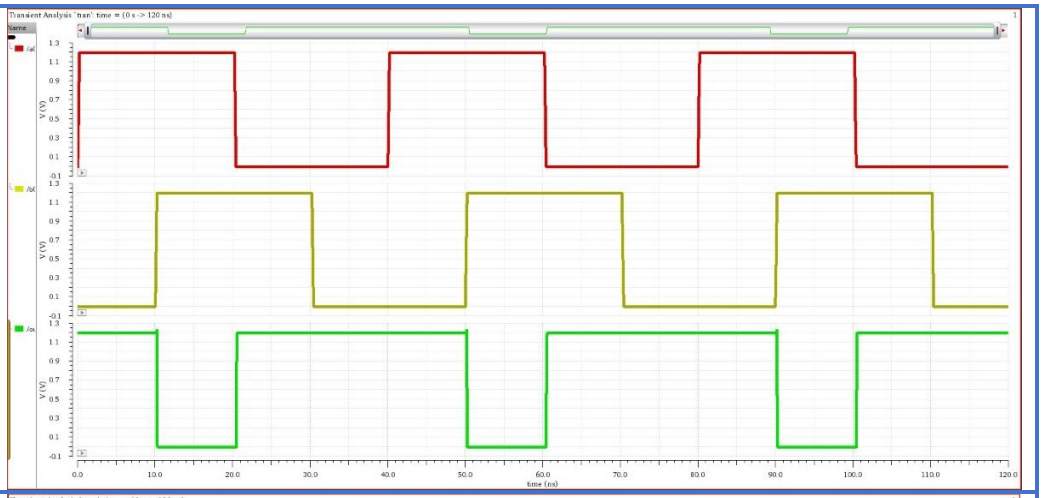
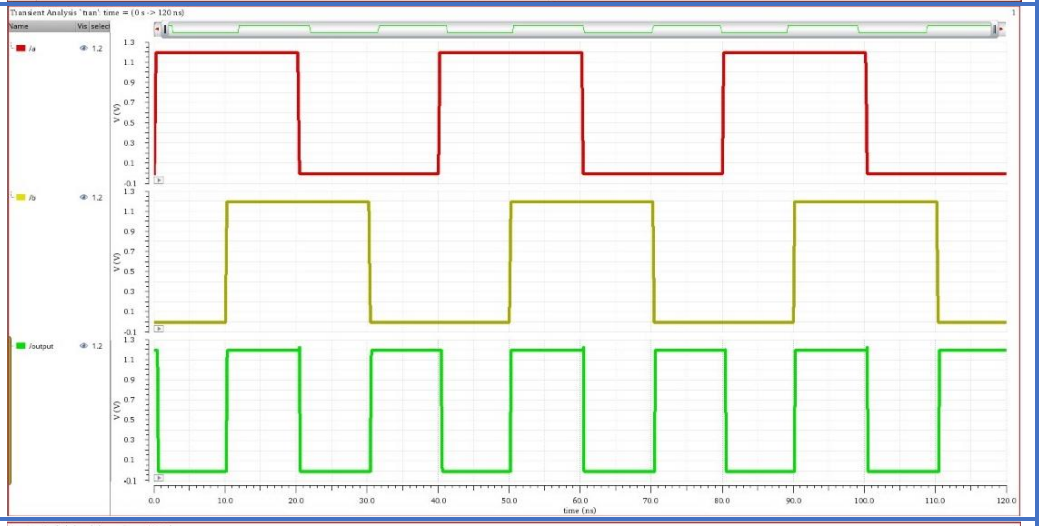
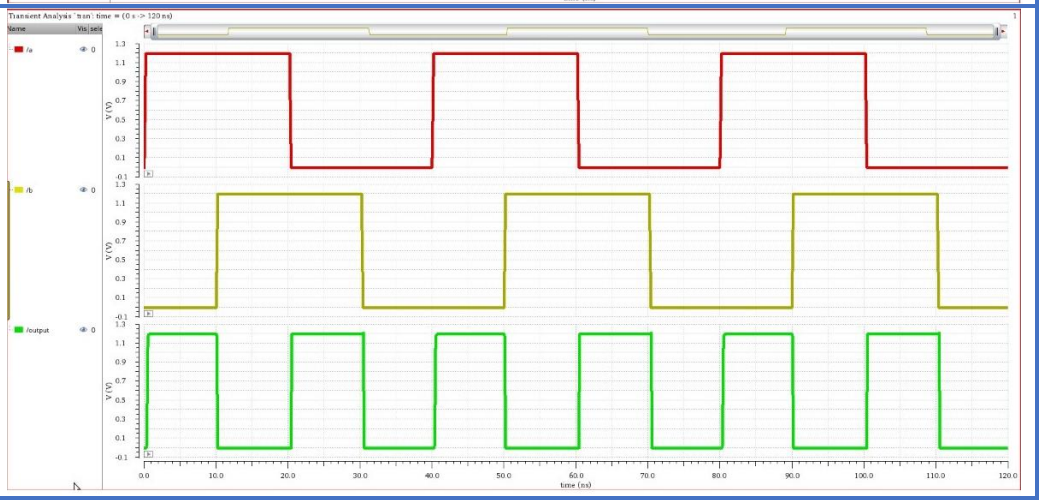
0
0
1
0
0
0
0
0
Decimal = 4

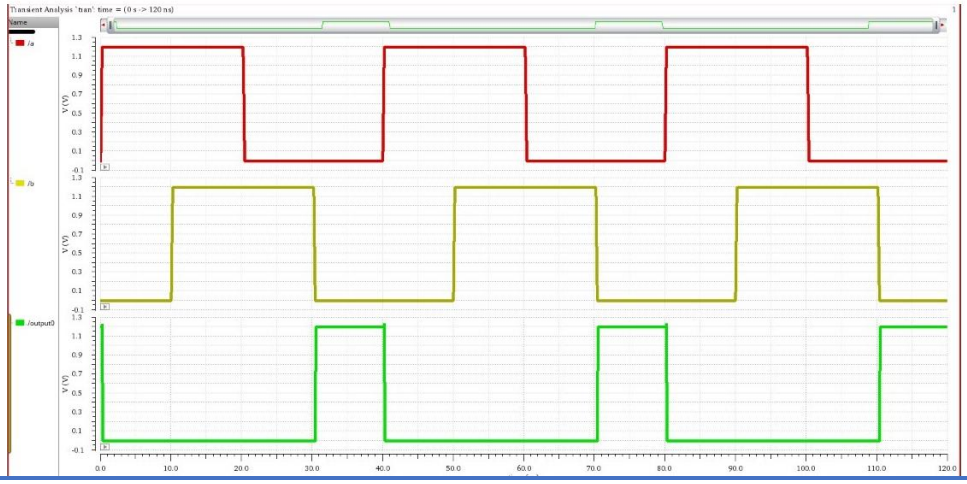
Logic Operations
Waveform testing
Transient Analysis

1000
$\bar{a}$



1001 $\bar{b}$	
1010 *AND*	
1011 *OR*	

1100 *XOR*	
1101 *XNOR*	
1110 *NAND*	

1111 *NOR*	

Delay and Power Consumption:

Delay:

At worst-case (multiplication) between inputs of A = (1111) and B=(1111).

At Rising Edges:

$$t_{plh} = 425.3x10^{-12}\ s$$
$$t_{plh} = 425.3\ ps$$

	1
Expression	(cross(value(VT("/y7") "Wp" 2e-06) 0.6 1 "rising" ...
Value	425.3E-12
Expression	(cross(value(VT("/y7") "Wp" 2e-06) 0.6 1 "falling"...
Value	102.6E-12

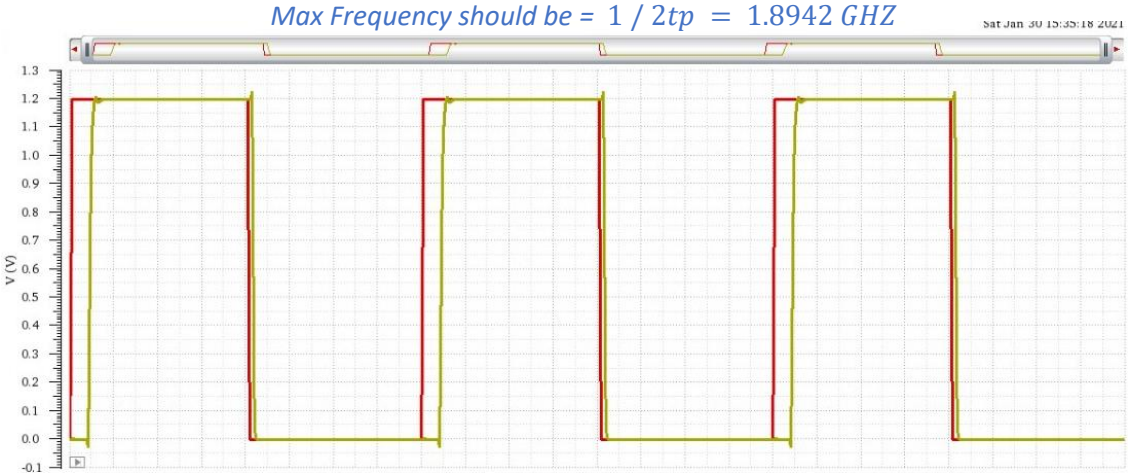
At Rising Edges:

$$t_{phl} = 102.6x10^{-12}s$$
$$t_{phl} = 102.6\ ps$$

Total propagation delay:

$$t_p = t_{phl} + t_{plh} = 2.639x10^{-12}$$
$$t_p = 2.638\ ps$$

Max Frequency should be = $1 / 2tp = 1.8942\ GHZ$



Power Consumption:

Total Current Drain from the supply Vdd =

$$100.3x10^{-6}\ A = 100.3\ \mu A$$

Average power = total current x voltage of the output

Outputs	***Average power***
$y0$	$58.9\ \mu watt$
$y1$	$12.4\ \mu watt$
$y2$	$50.6\ \mu watt$
$y3$	$58\ \mu watt$
$y4$	$62.8\ \mu watt$
$y5$	$79.9\ \mu watt$
$y6$	$53.6\ \mu watt$
$y7$	$73.9\ nwatt$

VHDL Code:

```
--Library decleration--
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;
USE ieee.numeric_std.all;
USE ieee.std_logic_unsigned.all;

--ENTITY decleration of MUX--
ENTITY mux_16_1 is
PORT (
ip0MUX: in std_logic_vector (7 downto 0);
ip1MUX: in std_logic_vector (7 downto 0);
ip2MUX: in std_logic_vector (7 downto 0);
ip3MUX: in std_logic_vector (7 downto 0);
ip4MUX: in std_logic_vector (7 downto 0);
ip5MUX: in std_logic_vector (7 downto 0);
ip6MUX: in std_logic_vector (7 downto 0);
ip7MUX: in std_logic_vector (7 downto 0);
ip8MUX: in std_logic_vector (7 downto 0);
ip9MUX: in std_logic_vector (7 downto 0);
ip10MUX: in std_logic_vector (7 downto 0);
ip11MUX: in std_logic_vector (7 downto 0);
ip12MUX: in std_logic_vector (7 downto 0);
ip13MUX: in std_logic_vector (7 downto 0);
ip14MUX: in std_logic_vector (7 downto 0);
ip15MUX: in std_logic_vector (7 downto 0);
sel_16_1: in bit_vector (3 downto 0);
out_mux_16_1: out std_logic_vector (7 downto 0)
);
end mux_16_1;
--ENTITY decleration of MUX--
--ARCHITECTURE of MUX decleration--
ARCHITECTURE mux_op OF mux_16_1 is
BEGIN
ALUselection: process(sel_16_1, ip0MUX, ip1MUX , ip2MUX ,ip3MUX,ip4MUX,
ip5MUX,ip6MUX , ip7MUX,
ip8MUX, ip9MUX, ip10MUX , ip11MUX ,ip12MUX,ip13MUX, ip14MUX,ip15MUX)
BEGIN
--selection lines of the MUX--
case sel_16_1 is
when "0000" => out_mux_16_1 <=  ip0MUX;
when "0001" => out_mux_16_1 <=  ip1MUX;
when "0010" => out_mux_16_1 <=  ip2MUX;
when "0011" => out_mux_16_1 <=  ip3MUX;
when "0100" => out_mux_16_1 <=  ip4MUX;
when "0101" => out_mux_16_1 <=  ip5MUX;
when "0110" => out_mux_16_1 <=  ip6MUX;
when "0111" => out_mux_16_1 <=  ip7MUX;
when "1000" => out_mux_16_1 <=  ip8MUX;
when "1001" => out_mux_16_1 <=  ip9MUX;
when "1010" => out_mux_16_1 <=  ip10MUX;
when "1011" => out_mux_16_1 <=  ip11MUX;
when "1100" => out_mux_16_1 <=  ip12MUX;
when "1101" => out_mux_16_1 <=  ip13MUX;
when "1110" => out_mux_16_1 <=  ip14MUX;
```

```
when "1111" => out_mux_16_1 <=  ip15MUX;
end case;
end process ALUselection;
end mux_op;
--library decelrations--
LIBRARY ieee;
USE ieee.STD_LOGIC_1164.ALL;
USE ieee.numeric_std.all;
USE ieee.std_logic_unsigned.all;
--ENTITY ALU decleration--
ENTITY A_L_U is
PORT (
a :IN std_logic_vector(3 downto 0); --input a of 3 bits--
b :IN std_logic_vector(3 downto 0); --input b of 3 bits--
sel :IN bit_vector(3 downto 0); -- selection line-
y :OUT std_logic_vector(7 downto 0)
);
--ENTITY ALU decleration--
end A_L_U;
--Architecture decleration of A_L_U --
architecture operations of A_L_U is
--Signal deceleration--
signal
inc_a,dec_a,inc_b,dec_b,transfer_a,add_a_b,sub_a_b,compa,compb,and_op,or_op,x
or_op,xnor_op,nand_op,nor_op: std_logic_vector (7 downto 0);
signal multiply_ab: std_logic_vector (7 downto 0);
--Component decelration--
--16:1 Mux implementation--
component mux_16_1 is
port(
ip0MUX: in std_logic_vector (7 downto 0);
ip1MUX: in std_logic_vector (7 downto 0);
ip2MUX: in std_logic_vector (7 downto 0);
ip3MUX: in std_logic_vector (7 downto 0);
ip4MUX: in std_logic_vector (7 downto 0);
ip5MUX: in std_logic_vector (7 downto 0);
ip6MUX: in std_logic_vector (7 downto 0);
ip7MUX: in std_logic_vector (7 downto 0);
ip8MUX: in std_logic_vector (7 downto 0);
ip9MUX: in std_logic_vector (7 downto 0);
ip10MUX: in std_logic_vector (7 downto 0);
ip11MUX: in std_logic_vector (7 downto 0);
ip12MUX: in std_logic_vector (7 downto 0);
ip13MUX: in std_logic_vector (7 downto 0);
ip14MUX: in std_logic_vector (7 downto 0);
ip15MUX: in std_logic_vector (7 downto 0);
sel_16_1: in bit_vector (3 downto 0);
out_mux_16_1: out std_logic_vector (7 downto 0)
);
end component;
--Architecture BEGIN Operations --
begin
inc_a<= ("0000"& a(3 downto 0))+"1"; --incerement a--
dec_a<= ("0000"& a(3 downto 0))-"1";--decrement a--
multiply_ab<=std_logic_vector(to_unsigned((to_integer(unsigned("0000"&
a))*to_integer(unsigned("0000"& b))),8));--multiplication--
inc_b<= ("0000"& b(3 downto 0))+"1";--incerement b--
```

```
dec_b<= ("0000"& b(3 downto 0))-"1";--decrement b--
transfer_a<=("0000"& a(3 downto 0));--transfer a--
add_a_b<=std_logic_vector(to_unsigned((to_integer(unsigned("0000"&
a))+to_integer(unsigned("0000"& b))),8));--add a and b--
sub_a_b<=std_logic_vector(to_unsigned((to_integer(unsigned("0000"& a))-
to_integer(unsigned("0000"& b))),8));--sub a and b--
compa <= ("0000"& not(a(3 downto 0)));--complimenet a--
compb <= ("0000"& not(b(3 downto 0)));--complimenet b--
and_op <= ("0000"&(a(3 downto 0) and b(3 downto 0)));--a and b--
or_op <= ("0000"&(a(3 downto 0) or b(3 downto 0)));--a or b--
xor_op <= ("0000"&(a(3 downto 0) xor b(3 downto 0)));--a xor b--
xnor_op <= ("0000"&(a(3 downto 0) xnor b(3 downto 0)));--a xnor b--
nand_op <= ("0000"&(a(3 downto 0) nand b(3 downto 0)));--a nand b--
nor_op <= ("0000"&(a(3 downto 0) nor b(3 downto 0)));--a nor b--
mux_1: mux_16_1 port map (ip0MUX => inc_a ,ip1MUX => dec_a,ip2MUX =>
multiply_ab,ip3MUX => inc_b,ip4MUX => dec_b,ip5MUX => transfer_a ,ip6MUX =>
add_a_b,ip7MUX => sub_a_b,
 ip8MUX => compa, ip9MUX => compb, ip10MUX => and_op, ip11MUX => or_op,
ip12MUX => xor_op, ip13MUX => xnor_op, ip14MUX => nand_op, ip15MUX => nor_op,
  sel_16_1=>sel, out_mux_16_1 => y);
end operations ;
```

VHDL Test Benching:

```
--Testbench For A_L_U--
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;
entity tb_ALU is
end tb_AlU;

architecture behaviour of tb_ALU is
component A_L_U
port( a :in std_logic_vector(3 downto 0);
b :in std_logic_vector(3 downto 0);
sel :in bit_vector(3 downto 0);
y :OUT std_logic_vector(7 downto 0)
);
end component;
signal a_test : std_logic_vector(3 downto 0) ;
signal b_test : std_logic_vector(3 downto 0) ;
signal sel_test : bit_vector(3 downto 0) ;
signal y_test : std_logic_vector(7 downto 0);   --output--
constant t : time := 50 ns;
begin
test: A_L_U port map (a => a_test, b => b_test, sel => sel_test, y =>
y_test);
TESTING: PROCESS
begin
a_test <= "1110";
b_test <= "0011";
sel_test <= "0000"; --Increment a--
wait for t;
```

```
sel_test <= "0001"; --Decrement a--
wait for t;
sel_test <= "0010";--Multiplycation--
wait for t;
sel_test <= "0011";--Increment b--
wait for t;
sel_test <= "0100";--Decrement b--
wait for t;
sel_test <= "0101";--Transfer a--
wait for t;
sel_test <= "0110";--ADD--
wait for t;
sel_test <= "0111";--Subtract--
wait for t;
sel_test <= "1000";--Complement a--
wait for t;
sel_test <= "1001";--Complement b--
wait for t;
sel_test <= "1010";--And Gate--
wait for t;
sel_test <= "1011";--OR Gate--
wait for t;
sel_test <= "1100";--XOR Gate--
wait for t;
sel_test <= "1101";--XNOR Gate--
wait for t;
sel_test <= "1110";--Nand Gate--
wait for t;
sel_test <= "1111";--Nor Gate--
wait for t;
wait ; --wait forever--
end process ;
end behaviour;
```