# LAB2: Parallel VHDL

## **Introduction:**

In this lab, I have designed a single bit and four bit multiplexer by taking four inputs (a, b, c, d) and an output (x) with a select bit (s) through which we can select an output.

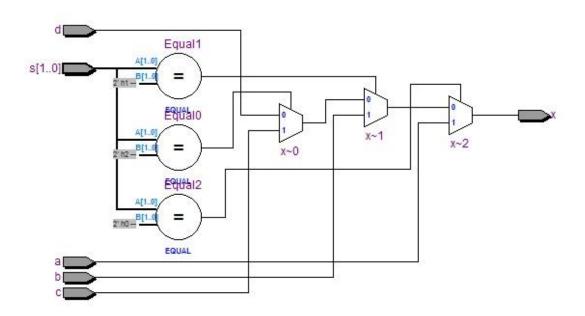
In a single bit multiplexer, we will take select (s) as two bit input and both input and output would be single bit.

In four bit multiplexer, we will take four bit input and output whereas, select bit will remain same.

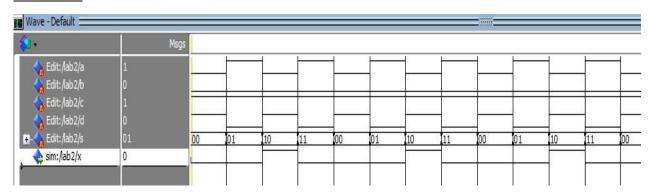
For an output waveform, I have taken input (a) as clock starting from 0, input (b) as a constant of logic 0, similarly input (c) as a constant with logic 1, input (d) as a clock starting from 1 and a select bit (s) as a two bit counter.

#### Single bit Multiplexer:

## **RTL VIEW:**



## Waveform:



# CODE:

```
LIBRARY IEEE;

USE IEEE.std_logic_1164.all;

Entity lab2 IS

PORT (a, b, c, d : IN std_logic;
s : IN std_logic_vector (1 downto 0);
x : OUT std_logic
);

END lab2;
```

ARCHITECTURE mux OF lab2 IS

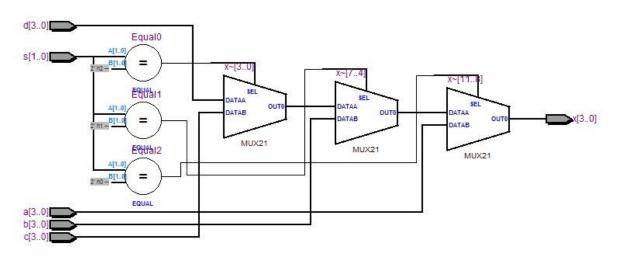
#### **BEGIN**

x <= a when s="00" ELSE
b when s="01" ELSE
c when s="10" ELSE
d when s="11" ELSE
'X';

END mux;

# Four bit Multiplexer:

## **RTL VIEW:**



## **WAVEFORM:**

<b>&amp;</b> i+	Msgs		18.		50		AV.		V.O	1.0		50		
₽-{} Edit:/lab2/a	0000	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	110
Edit:/lab2/b	0010	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110
	0100	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	0100
Edit:/lab2/d	1000	1000	1001	1010	1011	1100	1101	1110	1111	1000	1001	1010	1011	110
	00	00	01	10	11	00	01	10	11	00	01	10	11	00
+	0000	(0000	0011	0110	1011	0100	0111	1010	1111	1000	1011	1110	1011	1100

# CODE:

```
Library ieee;
Use ieee.std_logic_1164.All;
Entity lab2 is
port
(a,b,c,d: in std_logic_vector(3 downto 0);
s: in std_logic_vector(1 downto 0);
x: out std_logic_vector(3 downto 0)
);
End Entity;
Architecture mux4 of lab2 is
begin
x <= a when s= "00" else
b when s= "01" else
c when s= "10" else
d when s= "11" else
(others=>'X');
End mux4;
```

# **Conclusion:**

We can easily design a multiplexer in VHDL and verify its result through simulation.