Lesson 10

Combinational Logic

Mohd Saufy Rohmad

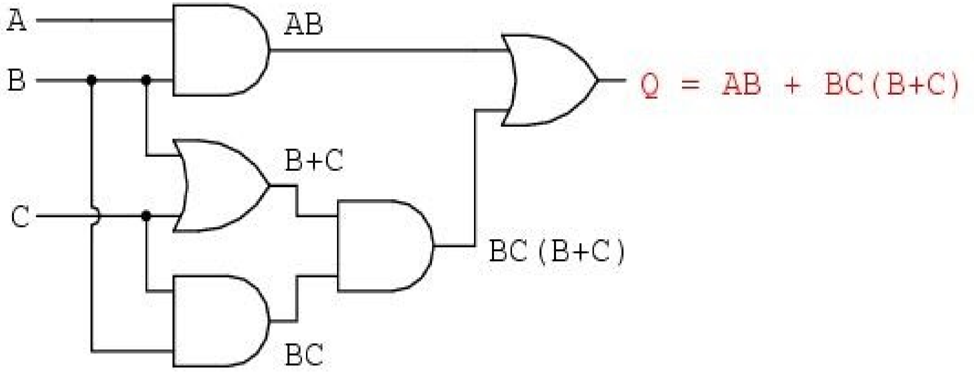
Technical Trainer and Consultant

Simple Combinational Logic

- Any section of logic that doesnt have any memory elements is considered as a combinational logic.
- So the simplest combinational logic represents a wire, for example,
- wire1 <= wire2;
- It means wire2 is connected to wire1.
- As mentioned in the previous lesson, we can create logic gates in the following way (lesson 5),
- Q <= AB + BC(B+C);
- We can visualize this logic here. Note that, A, B, C and Q are all translated as ports/wires.

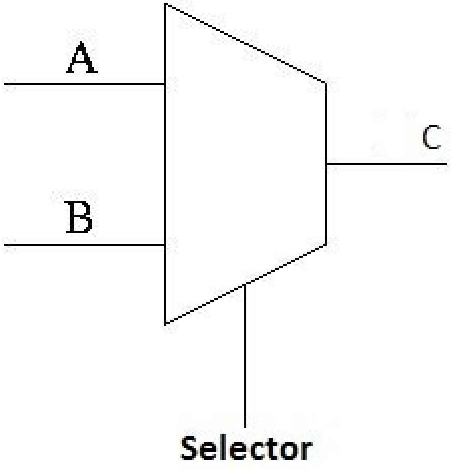
Simple Combinational Logic

- We can also write simple arithmetic units in this fashion (Hour 8),
- C <= resize(A+B,9);
- Here, A and B both have 8 wires to support 8-bit inputs. C has 9 wires for 9-bits.



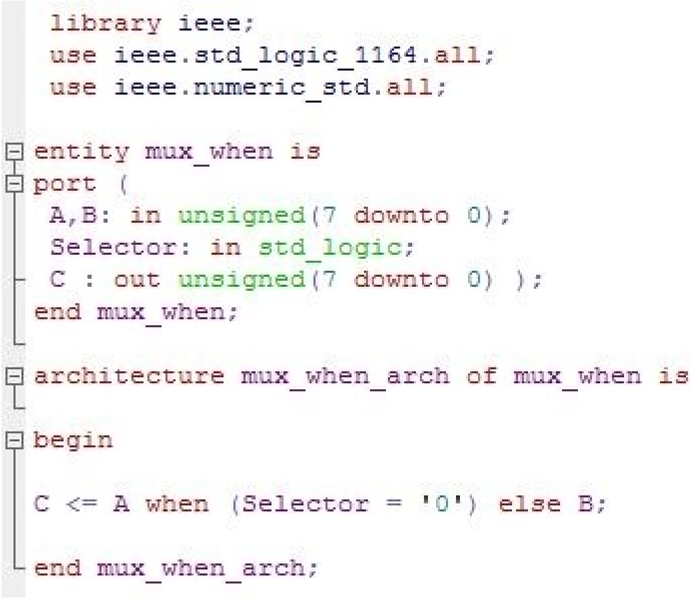
Conditional Statements

- We will encounter scenarios where we need a conditional statements.
- How can we use conditional statements as combinational logic? We will see that in this section.
- As an example, we will create the most basic 2:1 multiplexers with several conditional statements.
- Lets assume our multiplexer is selecting between inputs A and B based on the value of "Selector". If Selector is 0, the output will be A and if Selector is 1, the output will be B.



When / Select

- It is possible the design the multiplexer with when-select keyword. The code is given below,



```
 library ieee;
 use ieee.std_logic_1164.all;
 use ieee.numeric_std.all;

entity mux_when is
port (
 A,B: in unsigned(7 downto 0);
 Selector: in std_logic;
 C : out unsigned(7 downto 0) );
end mux_when;

architecture mux_when_arch of mux_when is

begin

C <= A when (Selector = '0') else B;

end mux_when_arch;
```

When / Select

- When-select is very useful for writing short codes. Another example of the when-select is,

```
C <= "1000" when selector = "00" else
     "0100" when selector = "01" else
     "0010" when selector = "10" else
     "0001" when selector = "11";
```

With/Select/When

- It is possible the design the multiplexer with with-select keyword.
- It is very similar to When-Select. The code is given below,

```
Library ieee;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity mux_with is
port(
A,B: in unsigned(7 downto 0);
Selector: in std_logic;
C: out unsigned(7 downto 0));
end mux_with;

architecture mux_with_arch of mux_with is
begin
with selector select
    C <= A when '0',
         B when others;
end mux_with_arch;
```

- With-select-when can also be very useful, for example,

```
with selector select C <=

      "1000" when "00",

      "0100" when "01",

      "0010" when "10",

      "0001" when "11";
```

Process (if else statement)

- We will introduce a very important construct for the VHDL language called process. Any if-else or case statement should be inside the process construct.

- Any process should have the following structure,

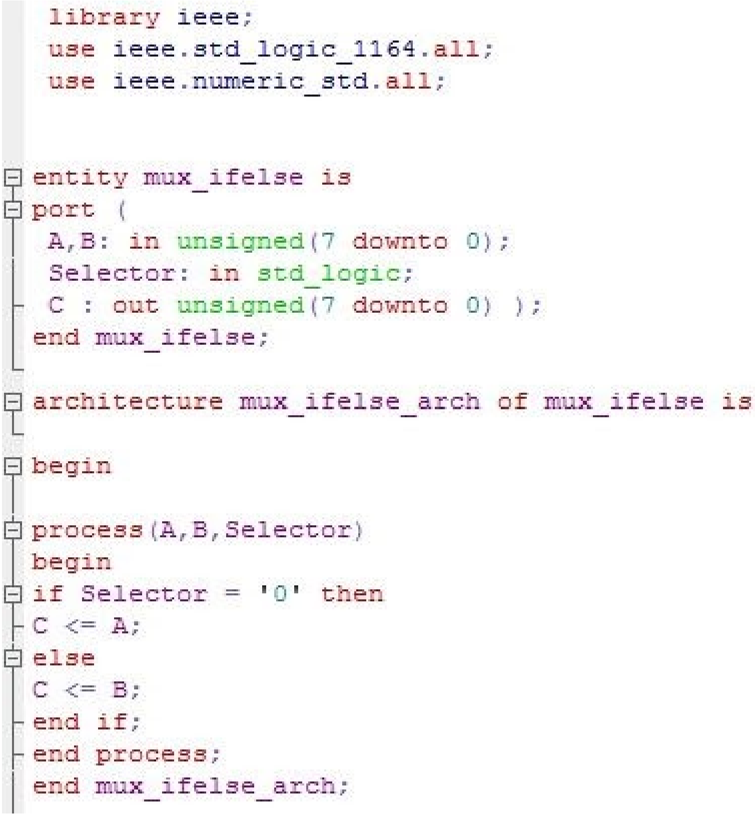
```
process(sensitivity_list)
begin

VHDL_code (if-else and case statements goes here)

end process;
```

A 2:1 multiplexer can also be written with if-else inside a process as,

Process (case statement)



```
 library ieee;
 use ieee.std_logic_1164.all;
 use ieee.numeric_std.all;


entity mux_ifelse is
port (
 A,B: in unsigned(7 downto 0);
 Selector: in std_logic;
 C : out unsigned(7 downto 0) );
end mux_ifelse;

architecture mux_ifelse_arch of mux_ifelse is

begin

process(A,B,Selector)
begin
if Selector = '0' then
C <= A;
else
C <= B;
end if;
end process;
end mux_ifelse_arch;
```

Thank You