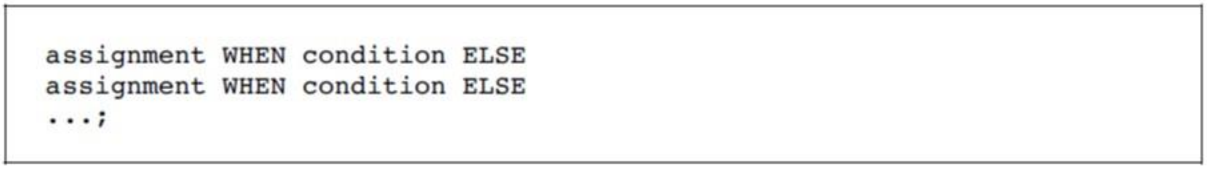
EE 213 Tutorial LAB 2

WHEN in VHDL

1.WHEN(Simple and Selected)

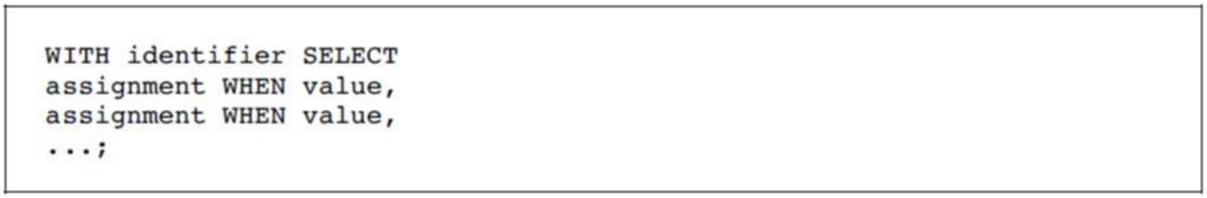
WHEN is one of the fundamental concurrent statements (along with operators and GENERATE). It appears in two forms: WHEN / ELSE (simpleWHEN) and WITH / SELECT / WHEN (selected WHEN). It ssyntax is shown below.

WHEN / ELSE:



```
assignment WHEN condition ELSE
assignment WHEN condition ELSE
...;
```

WITH / SELECT / WHEN:



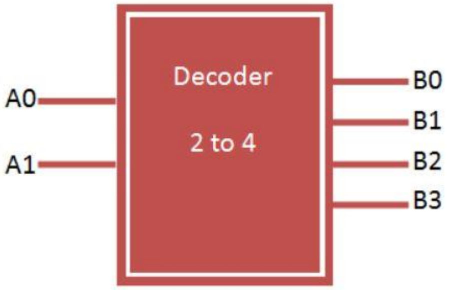
```
WITH identifier SELECT
assignment WHEN value,
assignment WHEN value,
...;
```

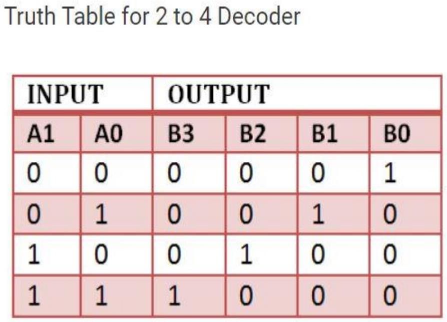
Whenever WITH / SELECT / WHEN is used, all permutations must be tested, so the keyword OTHERS is often useful. Another important keyword is UNAFFECTED, which should be used when no action is to take place.

Example:

```
------ With WHEN/ELSE ---------------------------------

outp<= "000" WHEN (inp='0' OR reset='1') ELSE

        "001" WHEN ctl='1' ELSE

        "010";

---- With WITH/SELECT/WHEN -----------------------

WITH control SELECT

        output<= "000" WHEN ‘0’,

                  "111" WHEN ‘1’,

                  UNAFFECTED WHEN OTHERS;
```

Example : Decoder




Truth Table for 2 to 4 Decoder

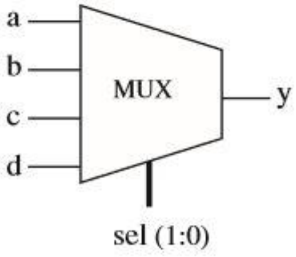
INPUT		OUTPUT			
A1	A0	B3	B2	B1	B0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

This example shows the implementation of a 2 to 4 decoder.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity decoder is
port(
a : in STD_LOGIC_VECTOR(1 downto 0);
y : out STD_LOGIC_VECTOR(3 downto 0)
);
end decoder;
architecture bhv of decoder is
begin
    y <= “0001” WHEN a="00" ELSE
         “0010” WHEN a="01" ELSE
         “0100” WHEN a="10" ELSE
         “1000”;
end bhv;
```

Example : Multiplexer

This example shows the implementation of a multiplexer. Two solutions are presented: one using WHEN/ELSE (simple WHEN) and the other with WITH/SELECT/WHEN (selected WHEN).



------- Solution 1: with WHEN/ELSE -------

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----------------------------------------
ENTITY mux IS
PORT ( a, b, c, d: IN STD_LOGIC;
            sel: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
            y : OUT STD_LOGIC);
END mux;
-----------------------------------------
ARCHITECTURE mux1 OF mux IS
BEGIN
      y <= a WHEN sel="00" ELSE
             b WHEN sel="01" ELSE
             c WHEN sel="10" ELSE
             d;
END mux1;
-----------------------------------------
```

--- Solution 2: with WITH/SELECT/WHEN ----

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----------------------------------------
ENTITY mux IS
PORT ( a, b, c, d: IN STD_LOGIC;
            sel: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
            y : OUT STD_LOGIC);
```

```
END mux;
------------------------------------------
ARCHITECTURE mux2 OF mux IS
BEGIN
    WITH sel SELECT
        y <= a WHEN "00", -- notice "," instead of ";"
             b WHEN "01",
             c WHEN "10",
             d WHEN OTHERS; -- cannot be "d WHEN "11" "
END mux2;
-------------------------------------------
```