VHDL Simple Constructs

Last updated 1/7/25

VHDL Simple Constructs

This is not a programming language
It is a hardware description language

VHDL Simple Constructs

- VHDL supports many hardware description constructs
  - 2 simple constructs:
    - with-select
    - when-else
  - Both create multiplexor-based logic structures
    - Combinatorial logic

VHDL Simple Constructs

- with-select
  - Choose a value when a certain situation exists

```
with decision_signal select result_signal <=        -- exhaustive list
      result_value when decision_value,
      result_value when decision_value,
      result_value when decision_value,
      result_value when decision_value;
```

Limitation: Only one result signal

VHDL Simple Constructs

- with-select
  - inA – 2 bit std_logic_vector

Exhaustive List

```
with inA select outA <=   "0100" when "01",
                          "0010" when "10",
                          "0001" when "11",
                          "1010" when "00",
                          "0000" when others;
```

Partial List

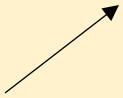
```
with inA select outA <=   "0100" when "01",
                          "0010" when "10",
                          "0000" when others;
```

Partially Common Result

```
with inA select outA <=   "0100" when "01",
                          "0010" when "10",
                          "0001" when ("11" or "00"),
                          "0000" when others;
```


Complex Selection

```
with (inA and inB) select outA <= "0100" when "01",
                                  "0010" when "10",
                                  "0001" when "00",
                                  "0000" when others;
```



VHDL Selection – with-select

- with-select
  - inA – 2 bit std_logic_vector

Exhaustive List

```
with inA select outA <=   sig1 when "01",
                          sig2 when "10",
                          sig3 when "11",
                          sig4 when "00",
                          sigx when others;
```

Partial List

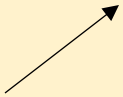
```
with inA select outA <=   sig1 when "01",
                          sig2 when "10",
                          sigx when others;
```

Partially Common Result

```
with inA select outA <=   sig1 when "01",
                          sig2 when "10",
                          sig3 when ("11" or "00"),
                          sigx when others;
```


Complex Selection

```
with (inA and inB) select outA <=sig1 when "01",
                                 sig2 when "10",
                                 sig3 when "00",
                                 sigx when others;
```



VHDL Simple Constructs

- with-select - example

```
----------------------------------------
--
-- with_select_example.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
----------------------------------------
--
-- with-select example
--
----------------------------------------
--
-- Inputs: inC
-- Outputs: outV
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity with_select_example is
   port (
         i_inC:   in std_logic_vector(3 downto 0);
         o_outV: out std_logic_vector(3 downto 0)
   );
end entity;
```

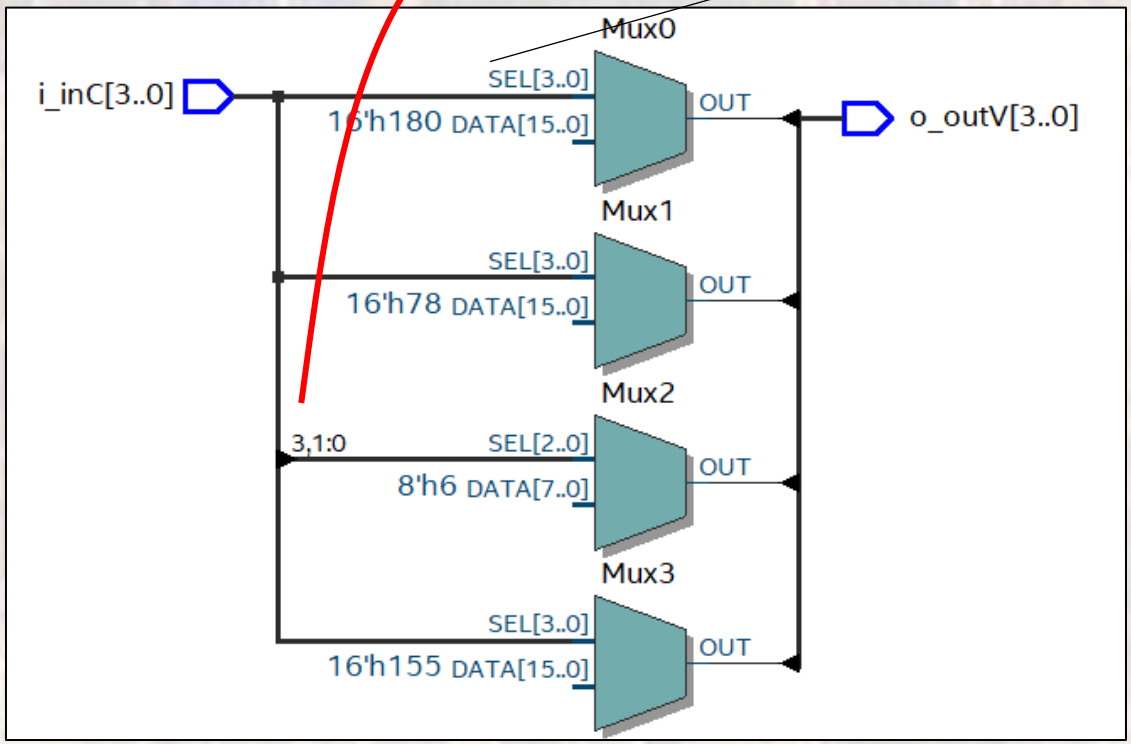
```
architecture behavioral of with_select_example is

begin
   with i_inC select o_outV <=
      "0001" when "0000",
      "0010" when "0001",
      "0011" when "0010",
      "0100" when "0011",
      "0101" when "0100",
      "0110" when "0101",
      "0111" when "0110",
      "1000" when "0111",
      "1001" when "1000",
      "0000" when others;

end behavioral;
```

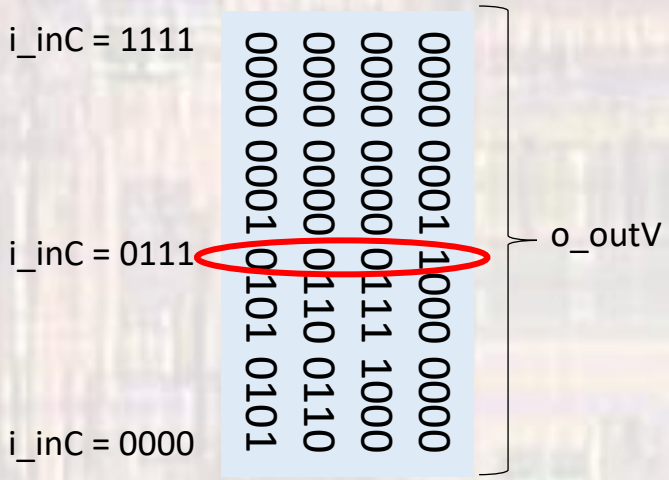
VHDL Simple Constructs

- with-select - example

Mux0 0000 0001 1000 0000
0000 0000 0111 1000
0000 0000 0110 0110
Mux3 0000 0001 0101 0101

complex wiring






```
"1000" when "0111",
"1001" when "1000"
```

Note: mux numbering does not correlate to bit numbering

VHDL Simple Constructs

- when-else
  - Choose a value when a certain situation exists

```
result_signal   <=   result_value when decision_signal = decision_value else
                     result_value when decision_signal = decision_value else
                     result_value when decision_signal = decision_value else
                     result_value;
```

Limitation: Only one result signal

VHDL Simple Constructs

- when-else

Exhaustive List

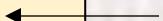
```
outA <= "1000" when inA = "00" else
        "0100" when inA = "01" else
        "0010" when inA = "10" else
        "0010" when inA = "11" else
        "0001";
```

Partial List

```
outA <= "1000" when inA = "00" else
        "0100" when inA = "01" else
        "0001";
```

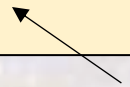
Partially Common Result

```
outA <= "1000" when inA = "00" else
        "0100" when inA = "01" else
        "0010" when inA = ("10"  or "11") else
        "0001";
```



Complex Selection

```
outA <= "1000" when (inA or inB) = "00" else
        "0100" when (inA or inB) = "01" else
        "0010" when (inA or inB) = "10" else
        "0001";
```



VHDL Selection – when-else

- when-else

Exhaustive List

```
outA <= sig1 when inA = "00" else
        sig2 when inA = "01" else
        sig3 when inA = "10" else
        sig4 when inA = "11" else
        sigx;
```

Partial List

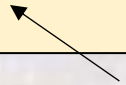
```
outA <= sig1 when inA = "00" else
        sig2 when inA = "01" else
        sigx;
```

Partially Common Result

```
outA <= sig1 when inA = "00" else
        sig2 when inA = "01" else
        sig3 when inA = ("10"  or "11") else
        sigx;
```


Complex Selection

```
outA <= sig1 when (inA or inB) = "00" else
        sig2 when (inA or inB) = "01" else
        sig3 when (inA or inB) = "10" else
        sigx;
```



VHDL Simple Constructs

- when-else - example

```
--------------------------------------
--
-- with_select_example.vhdl
--
-- created 7/5/2018
-- tj
--
-- rev 0
--------------------------------------
--
-- with-select example
--
--------------------------------------
--
-- Inputs: inC
-- Outputs: outV
--
--------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity with_select_example is
   port (
         i_inC:   in std_logic_vector(3 downto 0);
         o_outV: out std_logic_vector(3 downto 0)
   );
end entity;
```

```
architecture behavioral of with_select_example is

begin
   with i_inC select o_outV <=
      "0001" when "0000",
      "0010" when "0001",
      "0011" when "0010",
      "0100" when "0011",
      "0101" when "0100",
      "0110" when "0101",
      "0111" when "0110",
      "1000" when "0111",
      "1001" when "1000",
      "0000" when others;

end behavioral;
```

VHDL Simple Constructs

- when-else - example