CO102
Computer Hardware Experiment

Lecture 04: VHDL – Combinational Statement

Liang Yanyan

澳門科技大學
Macau of University of Science and Technology

Combinational Circuits

- Circuits whose outputs depend only on current input values.
  - No storage of past input values.
  - No state.
- Can be analyzed using laws of logic
  - Boolean algebra
  - E.g. a = b and c + d
- Signals change values immediately according to the assignments.

Concurrent code

- VHDL code is inherently concurrent (parallel).
  - Except that statements inside a PROCESS is sequential (will be addressed later).

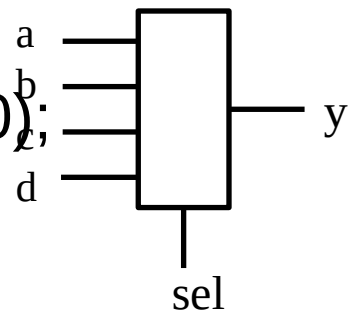
```
entity circuit is
        port (x, y : in std_logic;
                f1 : out std_logic;
                 f2 : out std_logic);
end circuit;
architecture boolean_eqn of circuit is
begin
        f1 <= x or y;
        f2 <= x and y;
end boolean_eqn;
```

when/else statement

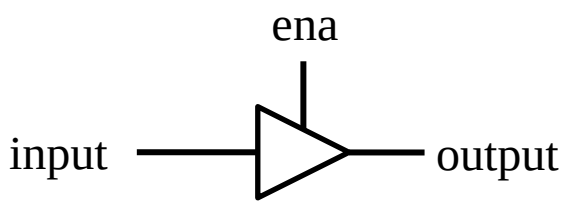
```
entity circuit is
        port (in1 : in std_logic_vector(1 downto 0);
                out1 : out std_logic_vector(2 downto 0));
end circuit;
architecture curcuit_arch of circuit is
begin
        out1 <=  “000” when (in1 = “00”) else
                 “010” when (in1 = “01”) else
                 “011” when (in1 = “10”) else
                 “100”;
end curcuit_arch;
```

With/select/when statement

```
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 mux_arch of mux is
begin
        with sel select
        y <= a when “00”,          -- notice “,” instead of “;”
                b when “01”,
                c when “10”,
                d when others;      -- cannot use “d when “11””
end mux_arch;
```



Tri-state buffer




```
entity tri_state is
        port (ena : in std_logic;
                input: in std_logic_vector (7 downto 0);
                output: out std_logic_vector (7 down to 0));
end tri_state;
architecture tri_state_arch of tri_state is
begin
        output <= input when (ena = ‘0’) else
                  (others => ‘Z’);
end tri_state_arch;
```

'Z‘ : High Impedance, put ‘Z’ in all 8 bits of output

Common mistakes

```
entity circuit is
        port (x, y : in std_logic;
                f : out std_logic);
end circuit;
architecture circuit_arch of circuit is
begin
        f <= x - y;
        f <= x + y;
end circuit_arch;
```

Common mistakes

```
entity circuit is
        port (x, y : in std_logic;
                f, g : out std_logic);
end circuit;
architecture circuit_arch of circuit is
begin
        f <= x - y;
        f <= g + y;
end circuit_arch;
```