Chapter. 4

☆ Behavioral Modeling ~~(Pg 8.7)~~ (Pg. 87)

☆ Entity Declaration

* syntax:

```
entity entity_name is
    generic (list of generics & their types);
    port (list of interfacing port names & their types);
end entity_name;
```

☆ Architecture Body (Pg. 89)

* Syntax:

```
architecture arch_name of entity_name is
    architecture_item_declaration;
begin
    concurrent statements, these can be:
        process statement
        Block statement
        concurrent procedure-call-statement
        generate statement
end architecture architecture_name;
```

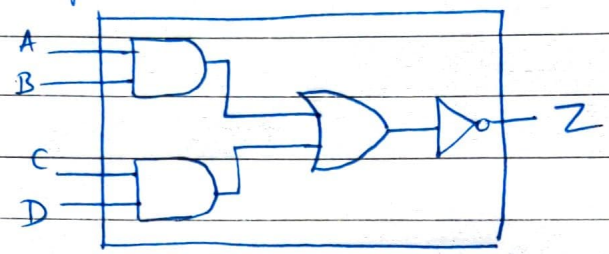
☆ Process Statement (Pg. 91)

* Syntax:

```
process_label: process (sensitivity_list)
    process-items_declaration;
    begin
        sequential statements;

    end process process_label;
```

✱ Example




```
library ieee;
use ieee.std_logic_1164.all;
entity logic_ckt is
port (A,B,C,D : in bit;
        Z : out bit);
end logic_ckt;
architecture seq. of logic_ckt is
begin
    process (A,B, C,D)
      variable Temp1, Temp2 : bit;
    begin
      Temp1 := A and B;
      Temp2 := C and D;
      Temp1 := Temp1 or Temp2;
      Z <= not Temp1;
    end process;
end seq;
```

✱ Example: NAND gate (Take this later)

```
architecture arch of NANDgate is
begin
    nandgate: process (a, b)
    begin
      if a = '1' and b = '1' then
        Z <= '0';
      else
        Z <= '1';
      end if;
```

```
    end process nand_gate;
end arch;
```

✡ Example : AND gate

```
process (a, b)
begin
    y <= a and b;
end process
```

OR gate

:

:

Lecture-4

✡ Recap

```
process (sensitivity list)
begin
    sequential statements
end process;
```

✡ Variable Assignment Statement (Pg. 92

* variables are declared & used inside process.
* Example: process (A)

```
    variable events_on_A : integer := -1;
begin
    events_on_A := events_on_A + 1;
end process;
```

✡ Signal Assignment Statement (Pg. 93)

* signals are used as concurrent as well sequential
* signal assignment can appear within or outside a process
* Example:

```
counter <= counter + "0010";  ---assign after delta ck
par <= par xor din after 12ns;
Z <= (A0 and A1) or (B0 or B1) after 6ns;
```