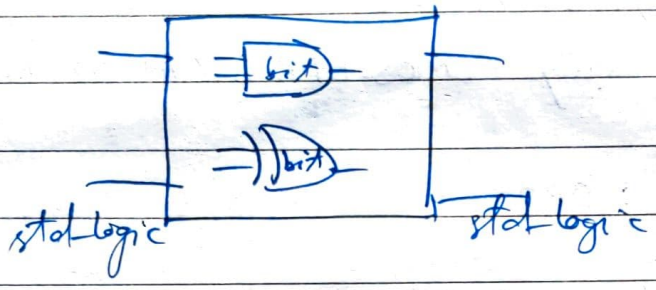
## ✱ Conversion Function (Pg. 176)

* Need: To resolve data type mismatch problem i.e.
  port type of entity ≠ port type of component

Ex:




Ex:

```
component COUNTER
    port (ctr : inout bit_vector (3 downto 0);
          clk, rst : in bit;
          par : out bit);
end component
```

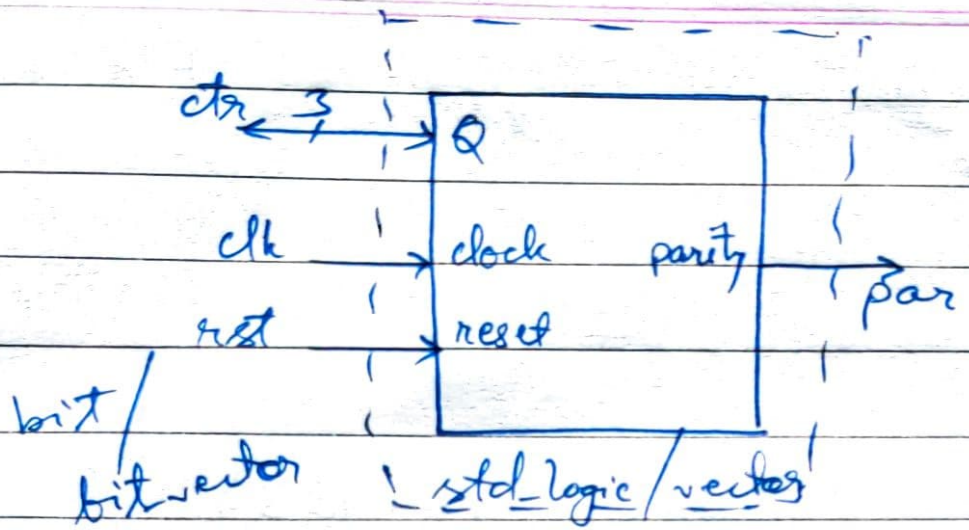
&

```
entity COUNTER is
    port (Q : inout std_logic_vector (3 downto 0);
          clock, reset : in std_logic;
          parity : out std_logic);
end ~~entity~~ COUNTER;
```

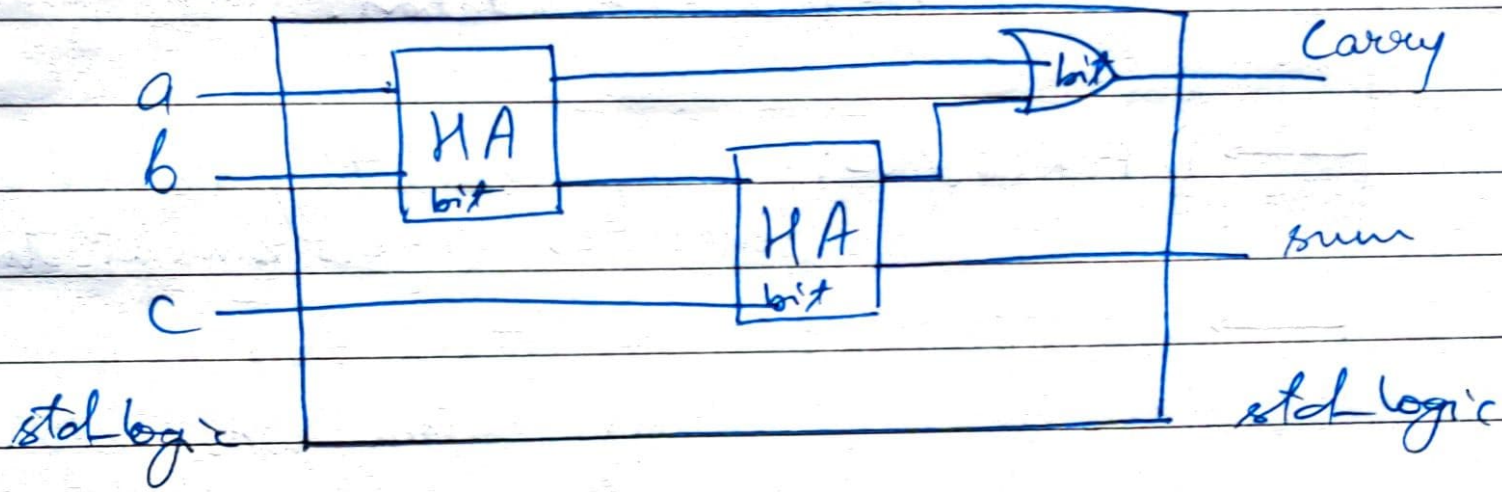
* Solution: Do the following ↓ in configuration specification

* Ex:-

```
for all: COUNTER use entity work.COUNTER
    port map (TO_BIT_VECTOR(Q) => TO_STD_LOGIC_VECTOR(ctr),
              clock => TO_STD_LOGIC(clk),
              reset ~~rst~~ => TO_STD_LOGIC(~~reset~~ rst),
              TO_BIT(parity) => par);
```




★ Write a conversion function for FA using HA



## ☆ Direct Instantiation (Pg. 178)

* Alternative to Configuration Specification & Configuration Declaration
* Example : Full Adder using Half Adder

```
entity full_adder is
    port(a, b, cin
         cout, sum
end          ;
or ——————————————
signal s1, c1, c2 : bit;
begin
    HA1 : entity work.HA(STR) port map (A, B, S1, C1);
    HA2 : entity work.HA(STR) port map (S1, Cin, sum, C2);
    O1 : configuration work.OR2CON port map(c1, c2, cout);
end
```

Note: No compo. declaration necessary / possible.