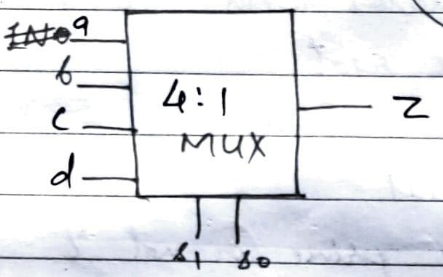
✯ Conditional Signal Assignment (Pg. 134)

→ Equivalent to "if else".

* Example:



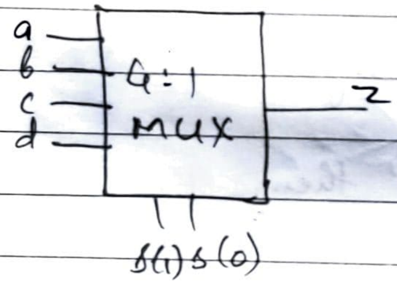
```
z <= a when s0 = '0' and s1 = '0' else
     b when s0 = '1' and s1 = '0' else
     c when s0 = '0' and s1 = '1' else
     d ;
```

~~* Write~~

✯ Selected Signal Assignment (Pg. 135)

→ Equivalent to "case".

* Example:



```
with s select
     z <= a when "00",
          b when "01",
          c when "10",
          d when others;
```

✯ Encoder using "when else" & "with select"

✯ Decoder using "when else" & "with select".

✯ Unaffected:

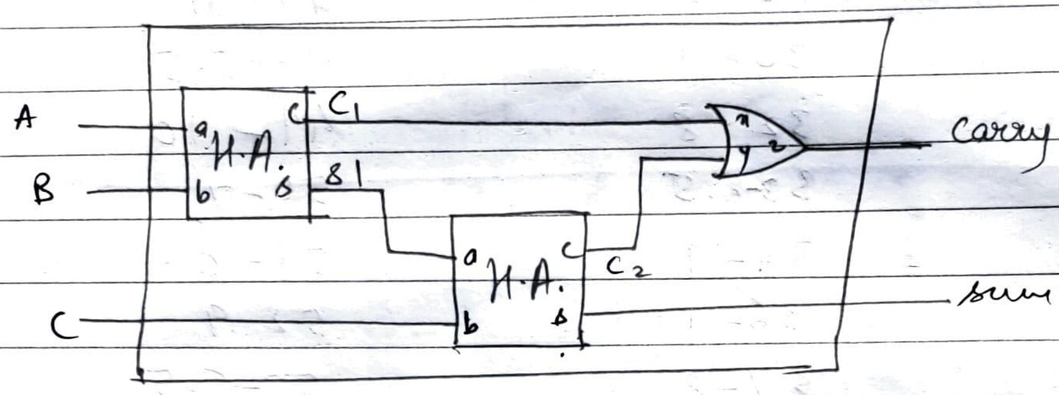
```
flag <= '1' when strobe = '0' else
        unaffected;
```

→ Equivalent to "null" statement.

* Structural Modeling

* coding is done blockwise

* Eg: Full adder using Half adder



* Write code for OR_Gate.vhd

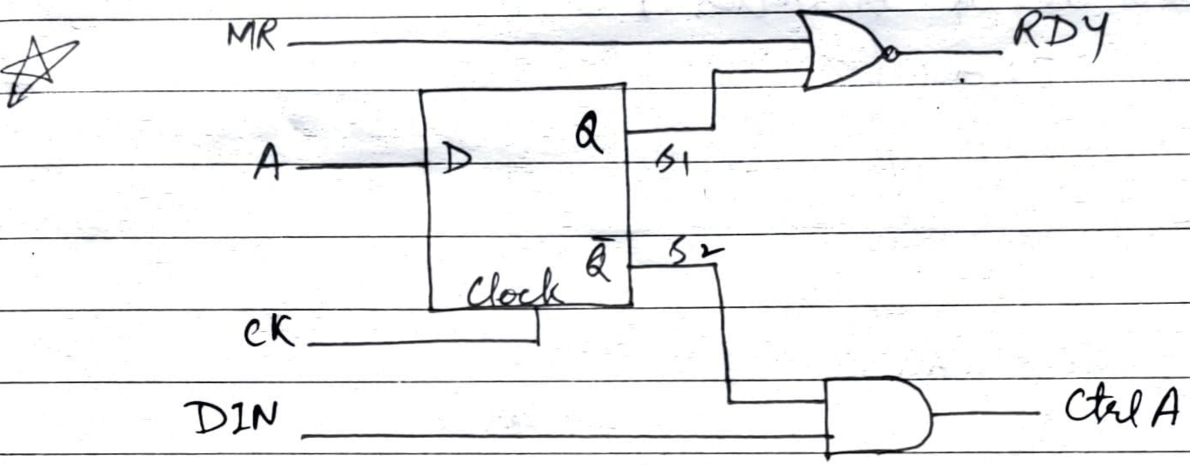
* Write code for half_adder.vhd

fulladder.vhd

```
library ieee;
use  ieee.std_logic_1164.all;
entity full_adder is
   port (A, B, C : in bit;
         sum, carry : out bit);
end full_adder;
architecture structural of fulladder is
   signal c1, s1, c2 : bit;
~~begin~~ component OR_Gate
        port (x, y : in bit;
                 z : out bit);
     end component
     component half_adder
        port (a, b : in bit;
                 s, c : out bit);
     end component
```

```
begin
    H1: halfadder (a => A, ~~B=>B~~ b => B, C1 => C, s1 => S);
    H2: half_adder (a => s1, b => C, c => c2, s => sum);
    O1: OR Gate (x => C1, y => C2, z => carry);
end structural;
```

☆



☆ 4 bit binary adder using fulladder.vhd

☆ Carry look ahead adder

☆ 4 bit counter using flipflop

☆ 4 bit shift register using flipflop.