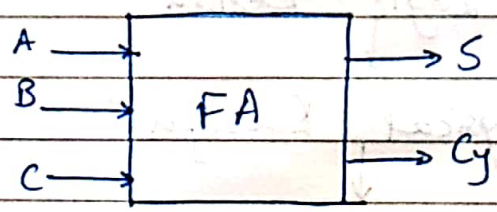
Lecture-2: Design Representation

* Three design representations
  i) Behavioral → Programs, Specifications, Truth table
  ii) Structural → Gates, adders, registers
  iii) Physical → Transistors / layouts, cells, chips / boards

* Behavioral example: full adder




```
module carry(S, Cy, A, B, C);
    input A, B, C;
    output S, Cy;
    assign S = A ^ B ^ C;
    assign Cy = (A & B) | (B & C) | (A & C);
endmodule
```

* Structural example: 4 bit binary adder

```
module add4 (s, cy4, cy_in, x, y);
    input [3:0] x, y;
    input cy_in;
    output cy4;
    wire [2:0] cy_out;
    add B0 (cy_out[0], s[0], x[0], y[0], cy_in);
    add B1 (                                    );
    add B2 (                                    );
    add B3 (                                    );
endmodule;
```

☆ Concept of Module

- It is the basic unit of h/w in verilog.
- A module cannot contain definition of another module.
- A module can however be instantiated within another module.

* Syntax:

```
module module_name (list of ports);
    input / output declarations
    local net declarations
    parallel statements
endmodule
```

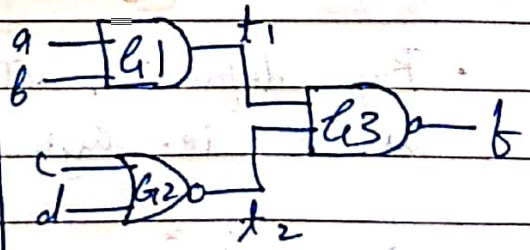
☆ Examples: AND Gate

```
module simple_AND (f, x, y);
    input x, y;
    output f;
    assign f = x & y;
endmodule
```

H-W Write similar codes for all gates.

☆ Example:

```
module two_level (a, b, c, d, f);
    input a, b, c, d;
    output f;
    wire t1, t2;
    assign t1 = a & b;
    assign t2 = ~ (c | d);
    assign f = ~ (t1 & t2);
endmodule
```



✱ Assign Statement

- assign statement represents "continuous assignment" i.e. LHS gets updated whenever expression on RHS changes
- assign variable = expression;
- LHS must be a "net" type variable
- RHS can be a "register" type or "net" type
- A module can contain any no. of assign statements.

✱ Data Types in Verilog

A variable in verilog belongs to one of the two data types:

a) NET

- must be continuously driven
- cannot be used to store a value
- used to model connections between continuous assignments & instantiations

b) REGISTER

- retains the last value assigned to it
- often used to represent storage elements
- default value is 'X'.

- By default, net are 1 bit values & with value as 'z' ie. high impedance state.

- Data types supported in NET:
  - wire, wor, wand, tri, supply0, supply1, etc.

- Verilog supports 0, 1, X & Z

✡ Example : Wired AND

```
module use_wand (a, b, c, d, f);
  input a, b, c, d;
  output f;
  wand f;
  assign f = a & b;
  assign f = c | d;
endmodule
```