UNIVERSITÀ
DEGLI STUDI DI TRIESTE

dipartimento
di ingegneria
e architettura

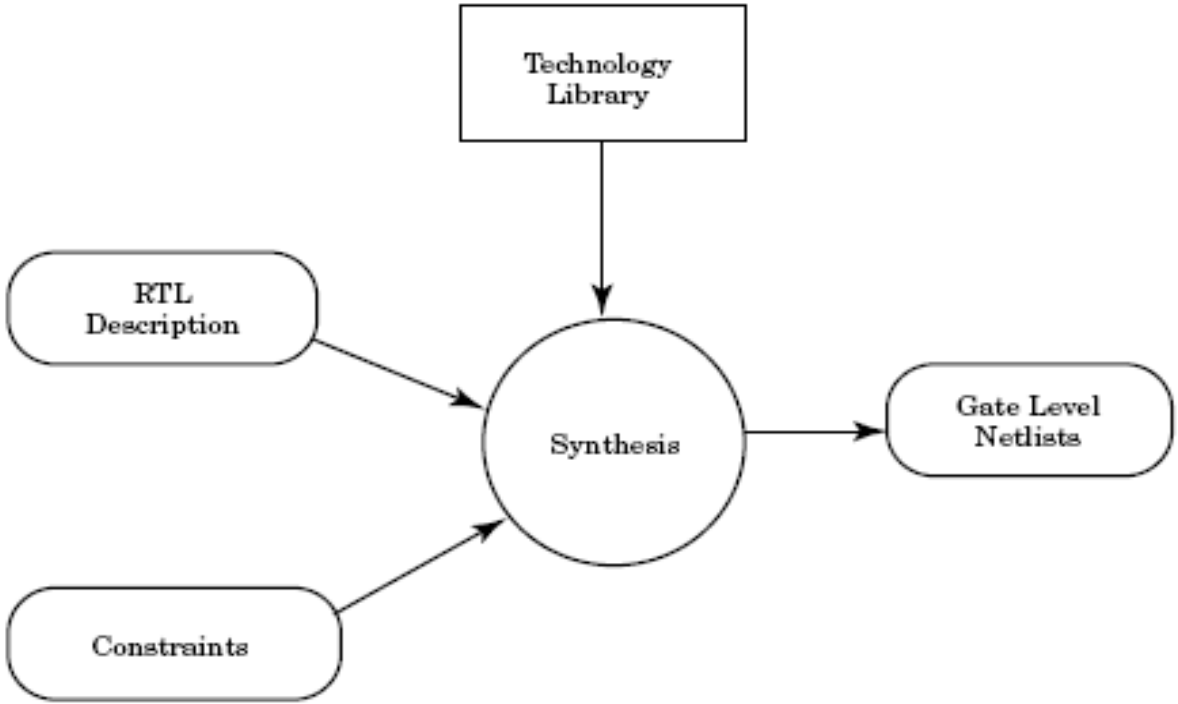
14 - VHDL Synthesis

A.Carini – Progettazione di sistemi elettronici

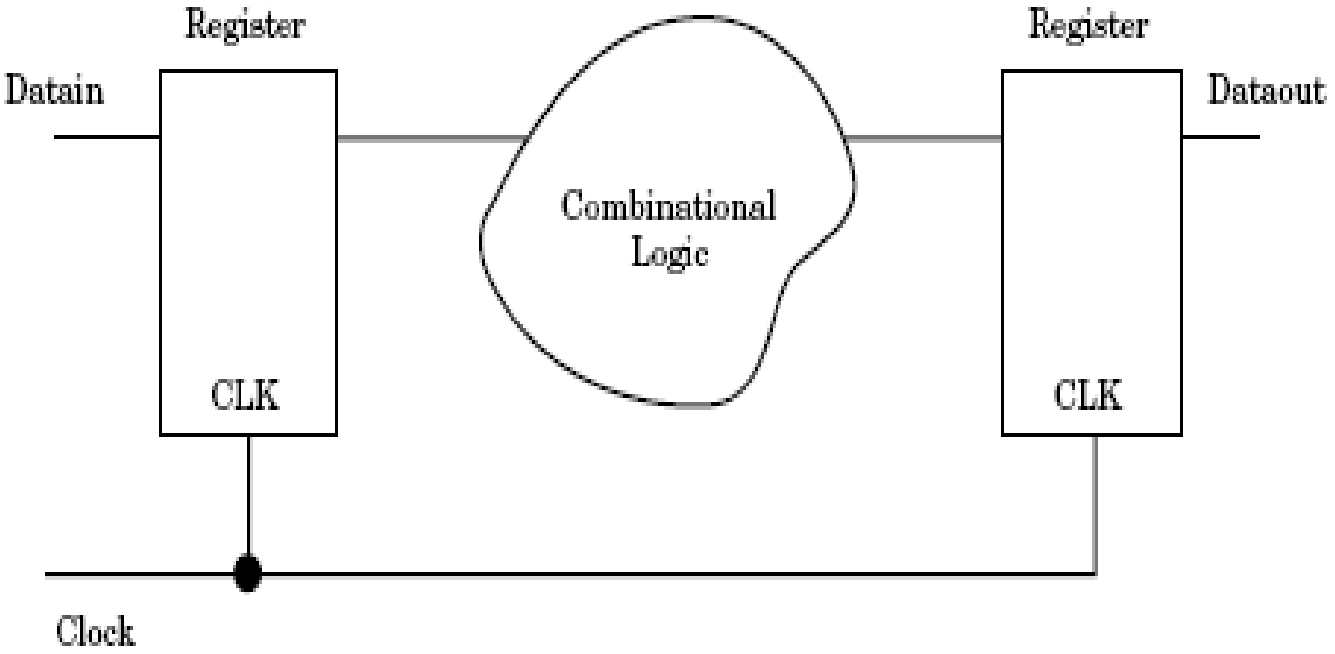
Synthesis

- It is an automatic method for passing from description at a high abstraction level to a description at a lower abstraction level.
- Current synthesis tools translates an RTL (register transfer level) description into a *gate level netlist*, i.e. into an interconnection of macro-cells at gate level. The models for these cells are included in *technology libraries*. Synthesis tools have a technology library for each supported technology.

Synthesis


Technology Library
RTL Description
Synthesis
Gate Level Netlists
Constraints

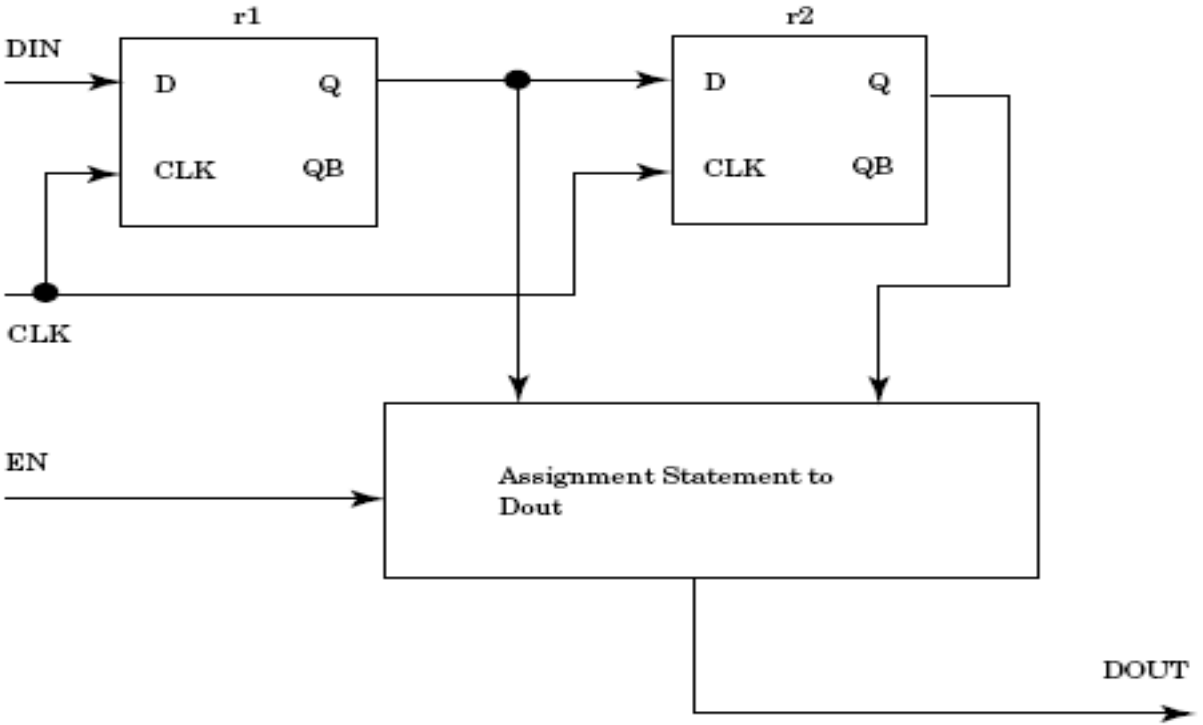
RTL description



Example of RTL description

```
ENTITY datadelay IS
 PORT( clk, din, en : IN BIT;
       dout : OUT BIT);
END datadelay;

ARCHITECTURE synthesis OF datadelay IS
  COMPONENT dff
    PORT(clk, din : IN BIT;
         q,qb : OUT BIT);
  END COMPONENT;
  SIGNAL q1, q2, qb1, qb2 : BIT;
BEGIN

  r1 : dff PORT MAP(clk, din, q1, qb1);
  r2 : dff PORT MAP(clk, q1, q2, qb2);

  dout <= q1 WHEN en = '1' ELSE
          q2;

END synthesis;
```

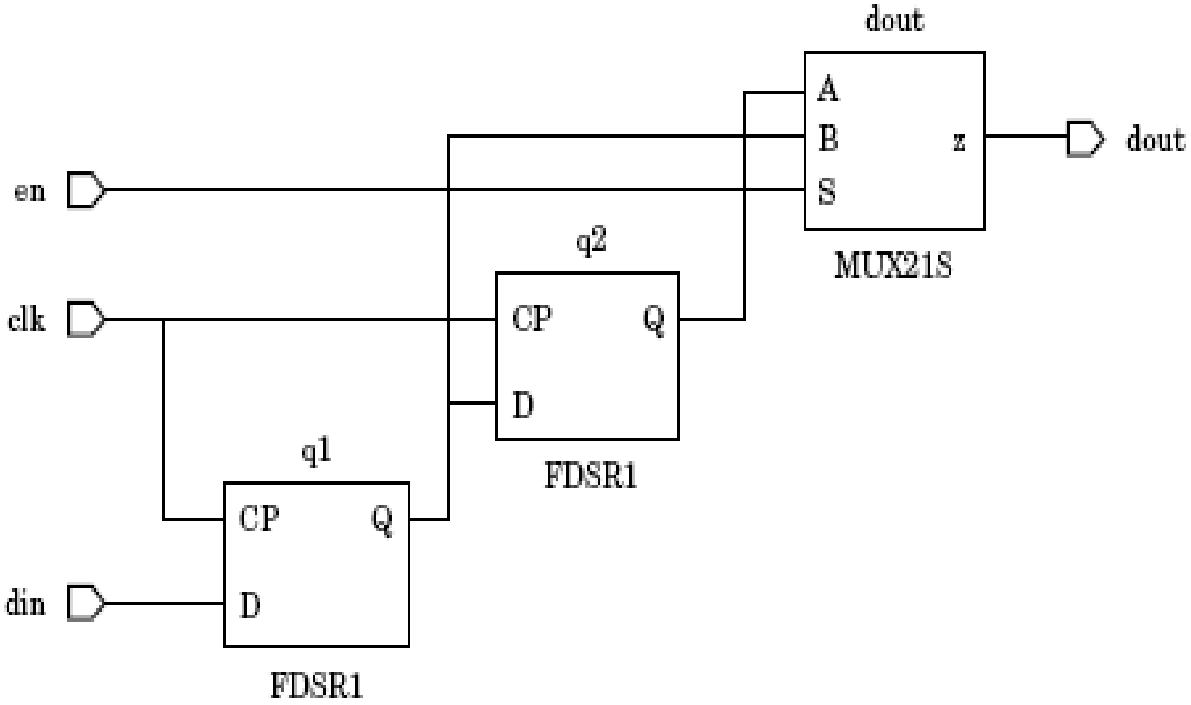
Example of RTL description



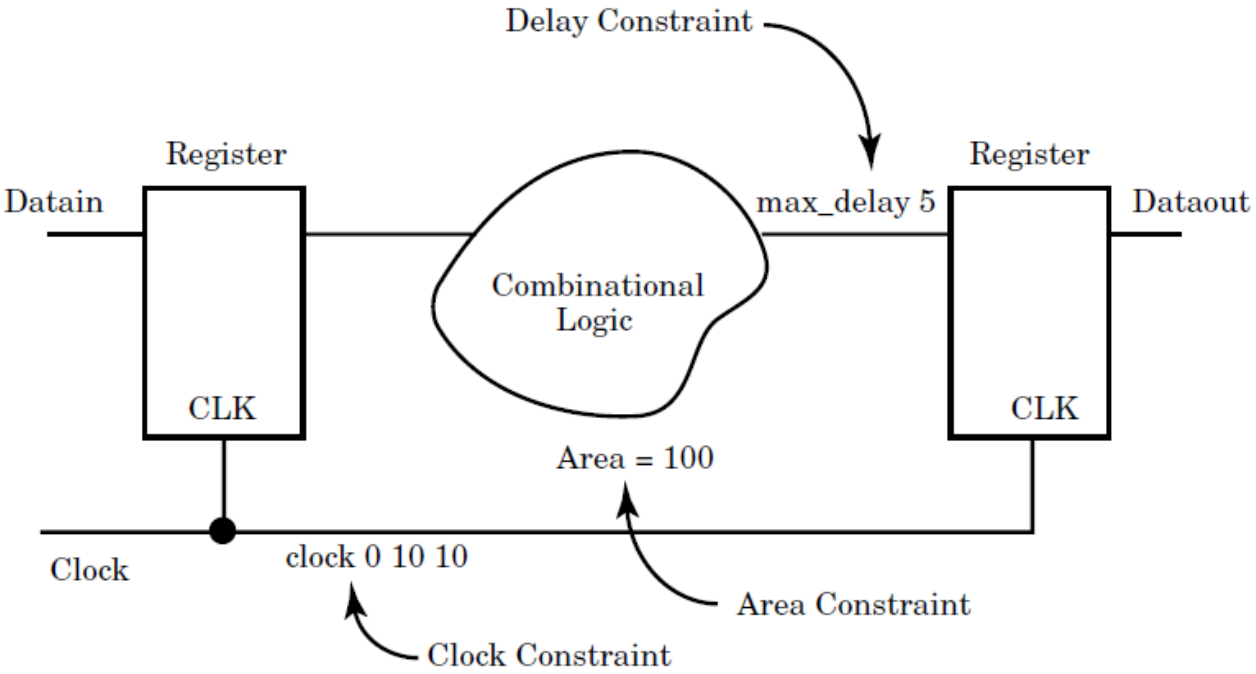
Example of RTL description

```
ENTITY datadelay IS
  PORT( clk, din, en : IN BIT;
        dout : OUT BIT);
END datadelay;

ARCHITECTURE inference OF datadelay IS
 SIGNAL q1, q2 : BIT;
BEGIN
 reg_proc: PROCESS
 BEGIN

  WAIT UNTIL clk'EVENT and clk = '1';

    q1 <= din;
    q2 <= q1;

END PROCESS;

dout <= q1 WHEN en = '1' ELSE
          q2;
END inference;
```

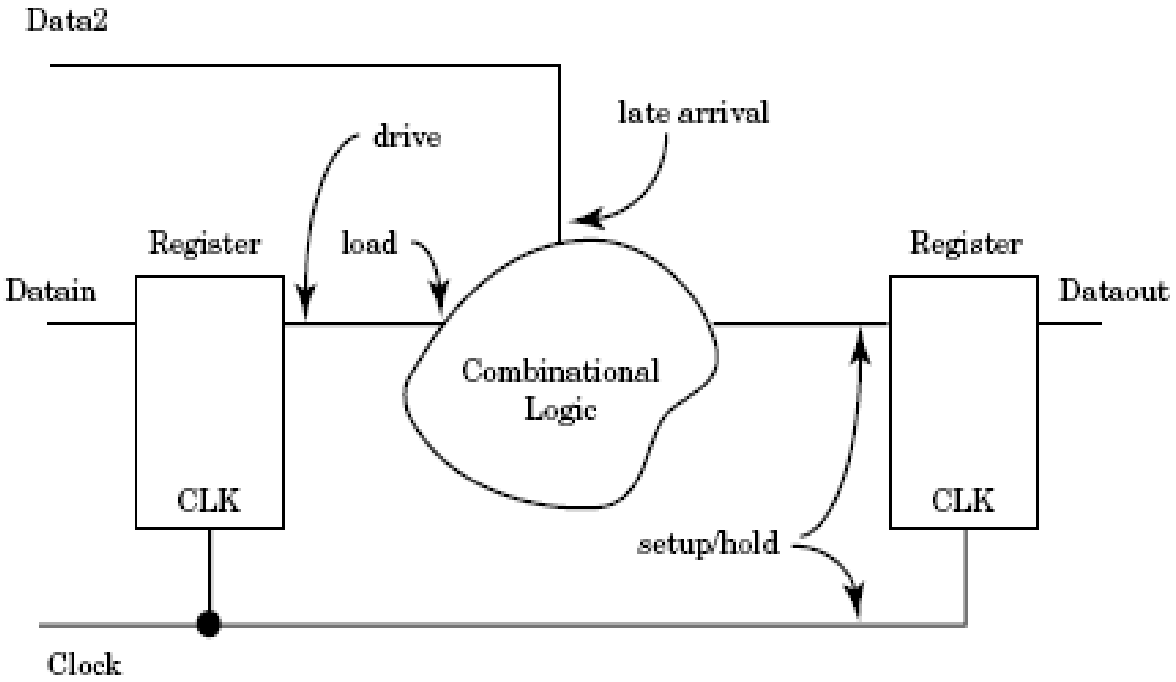
Gate level description



Constraints



Attributes


Data2
late arrival
drive
Register
load
Register
Datain
Dataout
Combinational
Logic
CLK
CLK
setup/hold
Clock

Technology libraries

- Contain the information necessary for creating an hardware *netlist* for the design, given the desired logic behavior of the system.
- Contain the information useful to the synthesis process for doing the right choices in building the design.
- Contain not only the logic function of the macro-cells, but also information on the area of the cell, on the input to output delays, on every constraint on fan-out, etc.

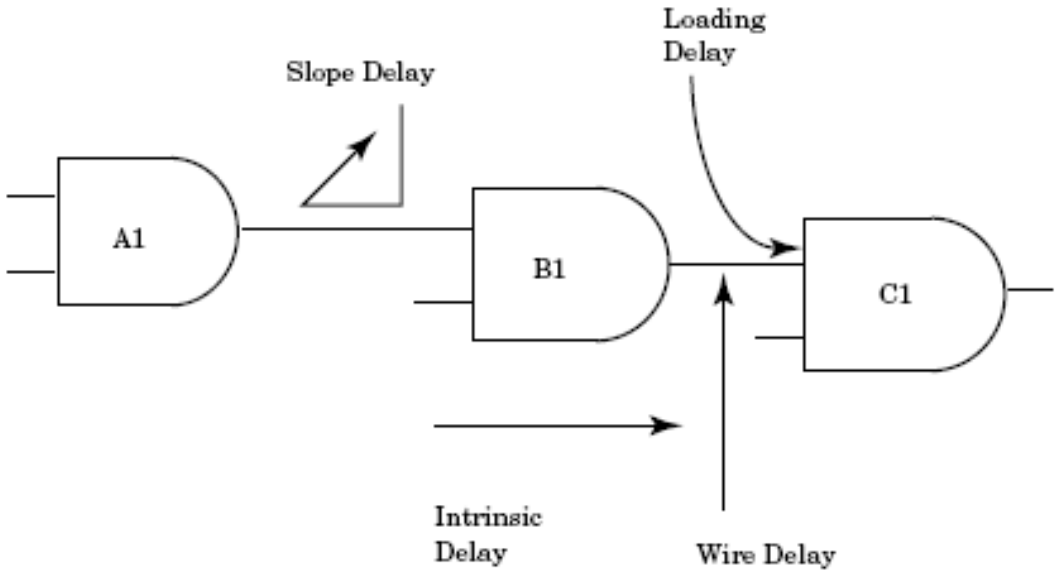
Technology libraries example

A 2 input AND gate in the Synopsys .lib format:

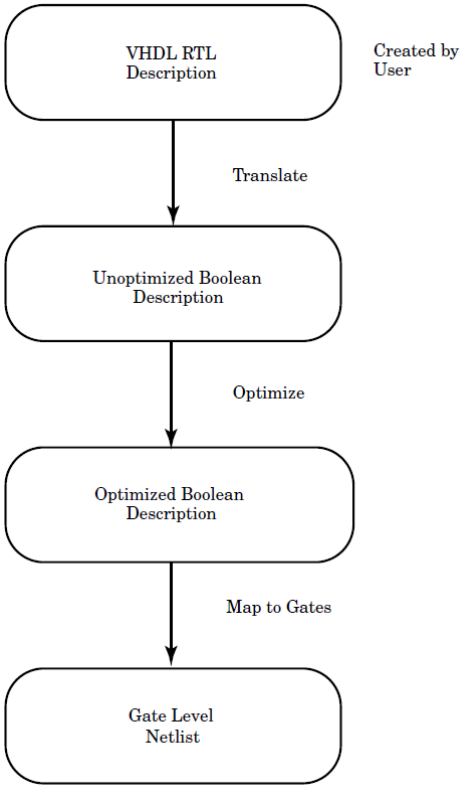
```
library (xyz) {
cell (and2) {
 area : 5;
 pin (a1, a2) {
    direction : input;
    capacitance : 1;
 }
 pin (o1) {
    direction : output;
    function : "a1 * a2";
    timing () {
       intrinsic_rise : 0.37;
       intrinsic_fall : 0.56;
       rise_resistance : 0.1234;
       fall_resistance : 0.4567;
       related_pin : "a1 a2";
    }
 }
}
}
```

Typical delay estimation

```
intrinsic_delay  + loading_delay + wire_delay + slope_delay
```



Typical delay estimation



Boolean optimization

- For example, by *flattening* and *factoring*.
- The unoptimized boolean description is translated in a low level description similar to that of a PLA (sum of products).
- Then, an optimization algorithm try to reduce the generated logic by sharing common terms (i.e., by introducing intermediate variables).

Flattening

```
Original equations
a = b and c;
b = x or (y and z);
c = q or w;
```

```
a = (x and q) or (q and y and z) or (w and x) or (w and y
      and z);
```

- Typically a fast implementation.
- Nevertheless, it could also be slower than an implementation with more logic levels, because of the high load on the input drivers.
- It requires a large area occupation.

Factoring

```
x = a and b or a and d;
y = z or b or d;
```

```
x = a and q;
y = z or q;
q = b or d;
```

- Surely the most efficient solution from the area viewpoint.
- It may not be the optimal solution from the delay viewpoint.
- Generally, a compromise is done:
  - Flattening on critical paths.
  - Factoring on all other paths.

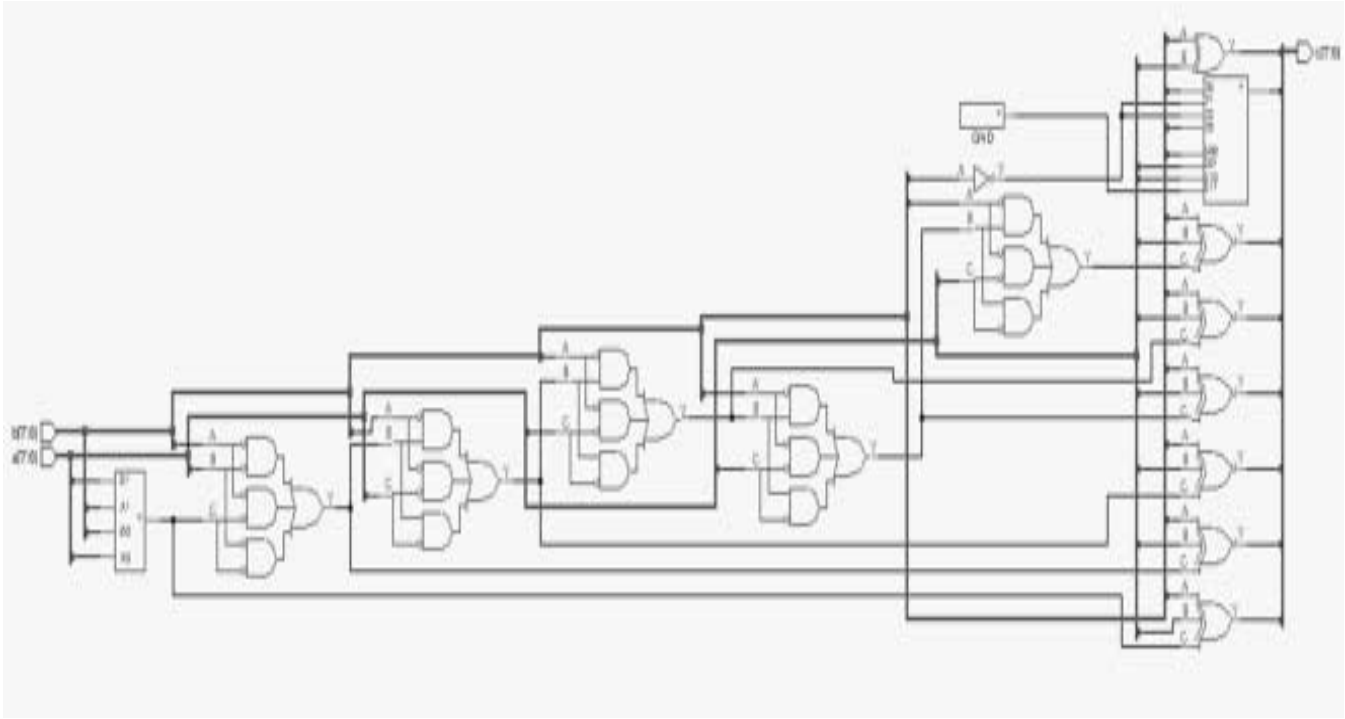
Mapping to gates

- The mapping process takes the optimized logic description, the *technology library,* the user constraints and it generates an optimized network, built entirely by cells of the technology library.
- During the mapping process the cells implement the optimized logic function.
- The cells themselves are optimized for meeting the speed and area constraints.
- As final step, the synthesis tool verifies that no technology rule has been violated (e.g., the maximum fan-out of every cell, etc.)

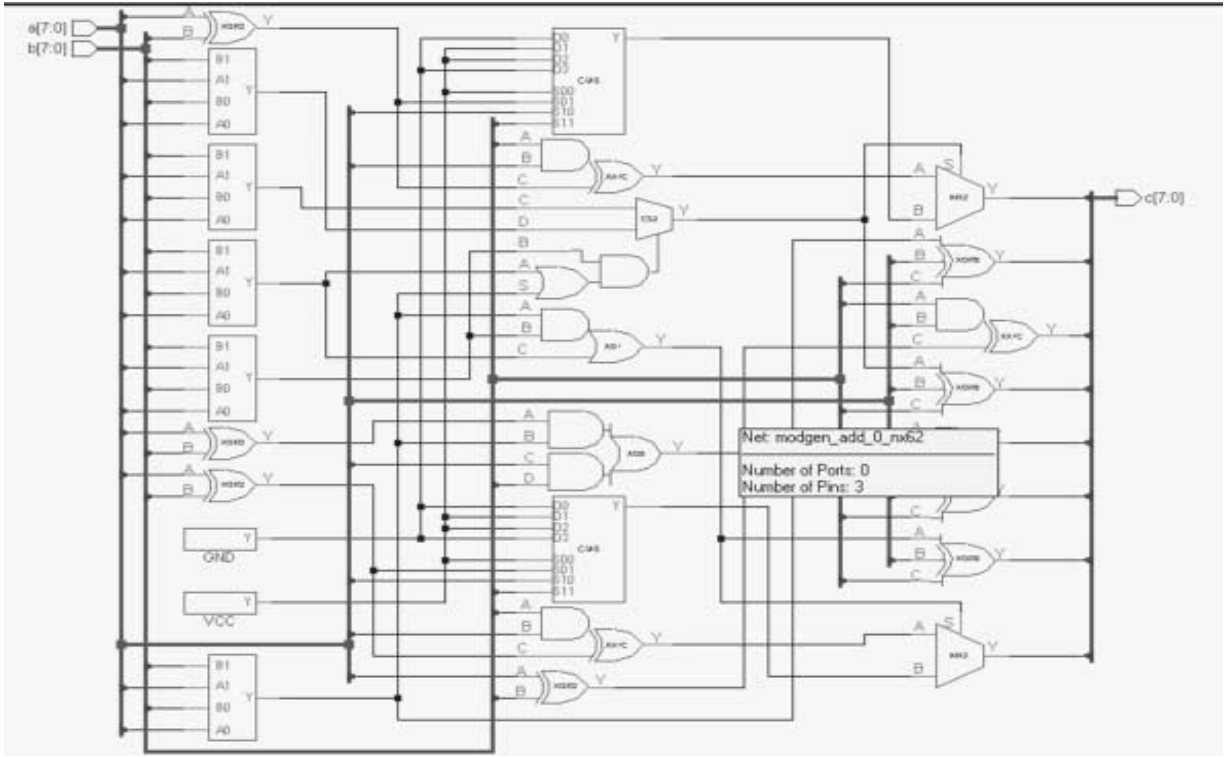
Mapping to gates

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.std_logic_unsigned.ALL;
ENTITY adder IS
 PORT( a,b : IN std_logic_vector(7 DOWNTO 0);
       c : OUT std_logic_vector(7 DOWNTO 0)
       );
END adder;

ARCHITECTURE test OF adder IS
BEGIN
 c <= a + b;
END test;
```

Ripple carry adder



Look ahead adder



See:

- Douglas L. Perry, «VHDL programming by example» McGraw Hill,
  - Chapter 9