

Definition

- Combination of [[Logic Gates]] and [[Sequential Circuits and Registers]] with a specific layout on a chip

Circuit Description Options

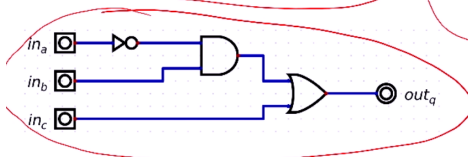
- boolean functions using [[Boolean Algebra]]
- truth tables
- circuit netlist
 - connected logic gates
- hardware description language
 - code that describes physical hardware
 - default
 - e.g. SystemVerilog
- examples

Example 1

in_a	in_b	in_c	out_q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$out_q = (\sim in_a \& in_b) \mid in_c$

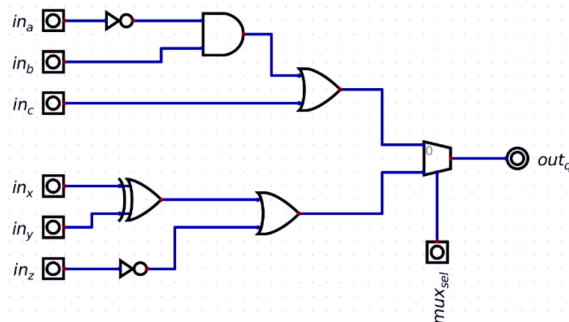
```
module simple_circuit (  
  input in_a,  
  input in_b,  
  input in_c,  
  output out_q  
);  
  assign out_q = ((~ in_a & in_b) | in_c);  
endmodule
```



Example 2

```
module simple_circuit_with_mux (  
  input logic in_a,  
  input logic in_b,  
  input logic in_c,  
  input logic in_x,  
  input logic in_y,  
  input logic in_z,  
  input logic mux_sel,  
  output logic out_q  
);
```

```
  always_comb begin  
    if (mux_sel == 0)  
      out_q = (~in_a & in_b) | in_c;  
    else  
      out_q = (in_x ^ in_y) | ~in_z;  
    end  
  end  
endmodule
```



Application Specific Integrated Circuit ASIC

- chip that physically realizes a given circuit
 - Basic steps to building your ASIC (very high level view):
 - Select your favorite semiconductor manufacturing plant (see https://en.wikipedia.org/wiki/List_of_semiconductor_fabrication_plants)
 - Receive the standard cell library from the plant (“the list of logic gates that the plant can build”)
 - Map our circuit to the available cells (called “synthesis”)
 - Place and route the cells
 - Let the plant physically build your circuit

Field Programmable Gate Arrays FPGA

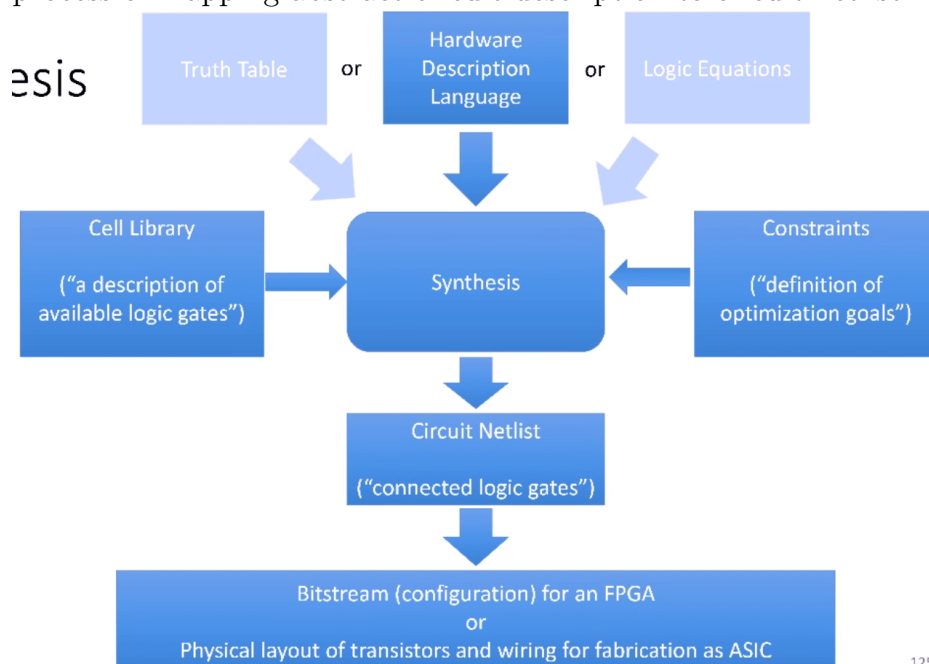
- existing hardware configured to correspond to a given circuit
- tradeoff between hardware and software
 - more efficient but more expensive than software
 - less efficient but less expensive than hardware

Basic concept (high level view):

- FPGA vendors build huge arrays of LUTs (Look-Up-Tables) and switches (highly regular repeated physical structure)
- You can map your design to this hardware (the gates are mapped to LUTs and the wiring is mapped to the switches connecting the LUTs)
- An FPGA bitfile stores how a given FPGA needs to be configured to realize your circuit (format is vendor-specific)
- Load the bitfile into the FPGA and the FPGA realizes your circuit

Logic Synthesis

- process of mapping abstract circuit description to circuit netlist



Addition of [[Binary Numbers]]

- adder (sum of two bits)
 - three inputs
 - * two digits to add
 - * carry over
 - two outputs
 - * digit of sum
 - * carry over
- n-bit addition
 - cascading these adders

