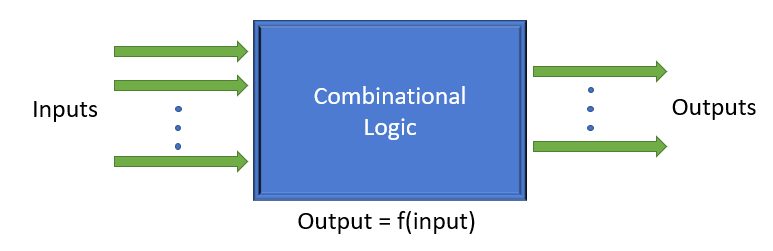
Digital Design and Verification

The working of digital computers, smart phones, etc all have some underlying basic logic manipulating just 0s and 1s to come up with interesting outcomes. With design specifications we convert our idea into silicon.

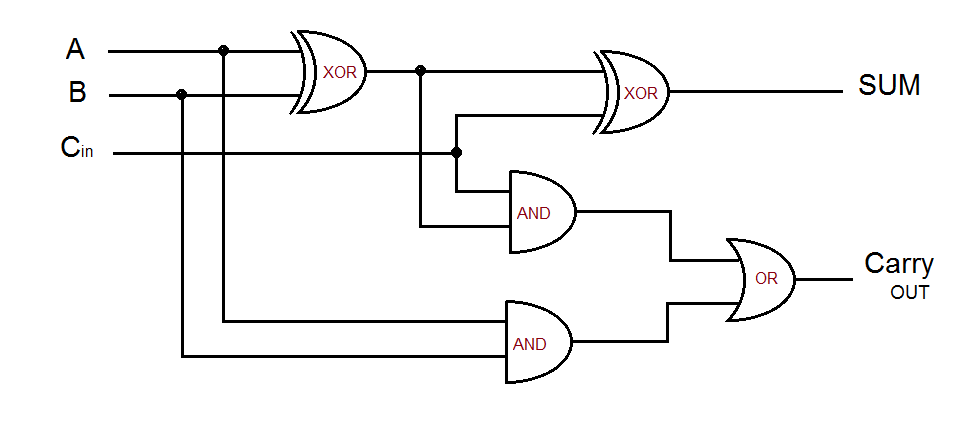
The design process of achieving it starts with our specifications on hand, then to model them using Hardware Description Languages (HDL) like verilog which describes the functionality as code in RTL Level. The RTL is then transformed into a GDSII file (or OASIS file) automatically using EDA (Electronic Design Automation) tools. The verification of the logic is a must and can’t be done manually as the design complexity has been exploding for the past several decades. Test benches and test cases are developed to verify the correctness of the logic as per the specification.

The circuit may be combinational i.e., which gives some output based on current input values or sequential where we include memory elements to make output as a function of past output values.

A Generic combinational block is represented as

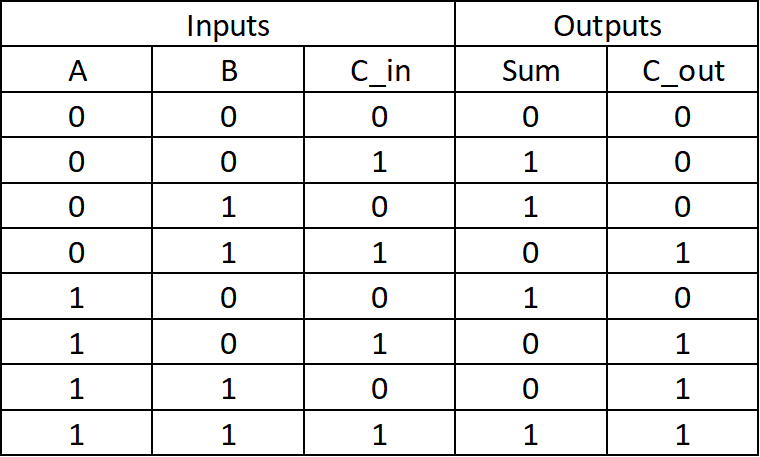


As an example, adder circuit

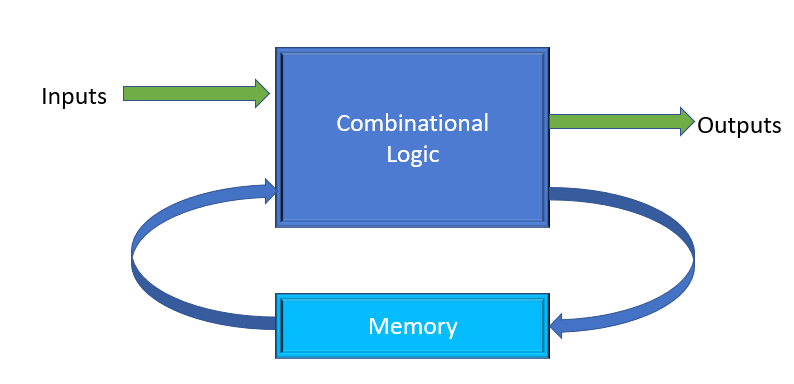


(Credit: <https://theorycircuit.com/full-adder-circuit-diagram/>)

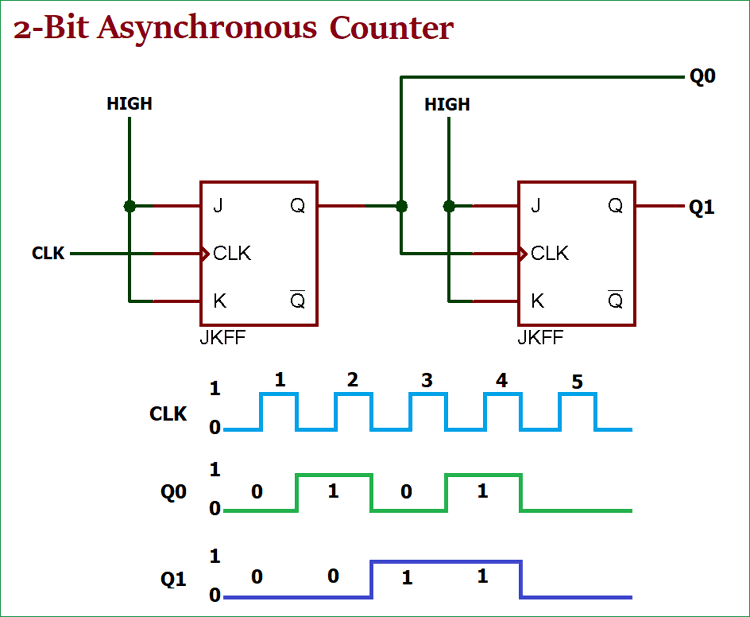
This circuit takes two input bits along with the carry bit from the previous stage of the adder( if it’s the first stage maybe we can take C\_in as 0). The circuit performs the addition of two bits along with carry and results in sum and carry out bits as per the truth table given below.



A Generic sequential circuit is represented as



A two bit asynchronous counter is as shown



(Credit: https://circuitdigest.com/tutorial/asynchronous-counter)

This block counts 0, 1, 2, 3 repeatedly. The JK Flip Flop with both input as HIGH results in toggling of the output as the circuit gets triggered.

Verilog is one of the many available HDLs used for modelling such circuits. Verilog makes logic blocks that implement one functionality as a *module*. The inputs and output from the logic design are named as *ports*. Syntax for creating a module is as shown

module <module-name> (<module-port-list>);

//functionality description

endmodule

Design simulators are tools used to verify the intended functionality of the logic block. There are several design simulators both commercial and opensource available for simulating the Verilog designs. For example, Verilator, Icarus Verilog are the open source tools used to simulate the logic.

Typically the design was verified using test benches written again in verilog. The test benches call instatiate the design module and provide necessary test inputs exercising the module. The output signals are observed to see the values given by the design. The output is then compared to the reference model which gives the expected output. The comparison explains whether the design is performing as expected.

A sample verilog code that implements full adder is as follows

module full\_adder(

input a, b, c\_in,

output sum, c\_out

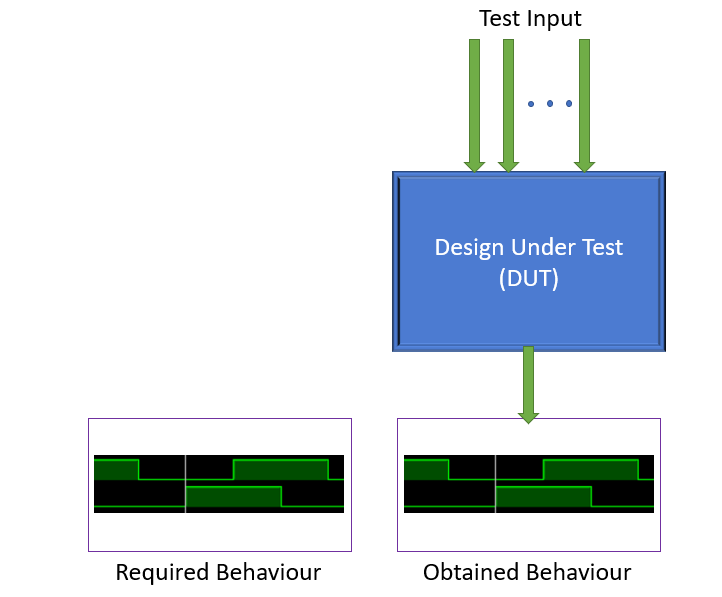
);

assign sum = a ^ b ^ c\_in;

assign c\_out = (a & b) + (b & c\_in) + (c\_in & a);

endmodule

Verification of circuit design takes place by comparing the desired result with passing the test input to the device under test



Full adders are simple and just require hardly a few minutes to test its functionality. Think of some complex circuits like code converters or finite state machines that do some complex task, which require many test cases to be tested upon. Manually verifying the logic by giving a set of inputs and comparing with expected one is tedious in this case because there are many different cases to test on with a lot of input bits. Use of tools which can automatically give input and check for proper output and return a summary of tests done can help in this regard. Now the verifier writes in a file the possible inputs and expected output. The tool will take up the design in HDL and provide input as mentioned and compare the output with the expected one and return a summary of the testing. Vyoma’s UpTickPro is a tool that does this.