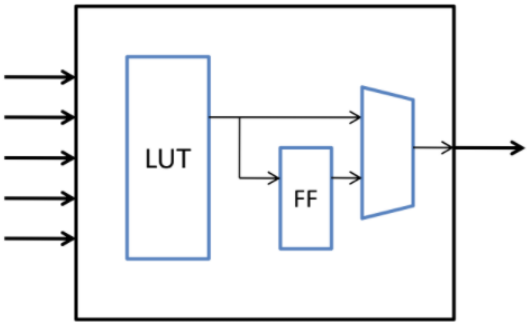
Field Programmable Gate Array

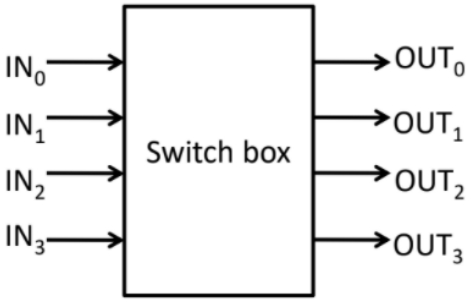
Aim: To make an FPGA using logic tiles and switch boxes to implement multiple designs using the same circuit.

Theory:

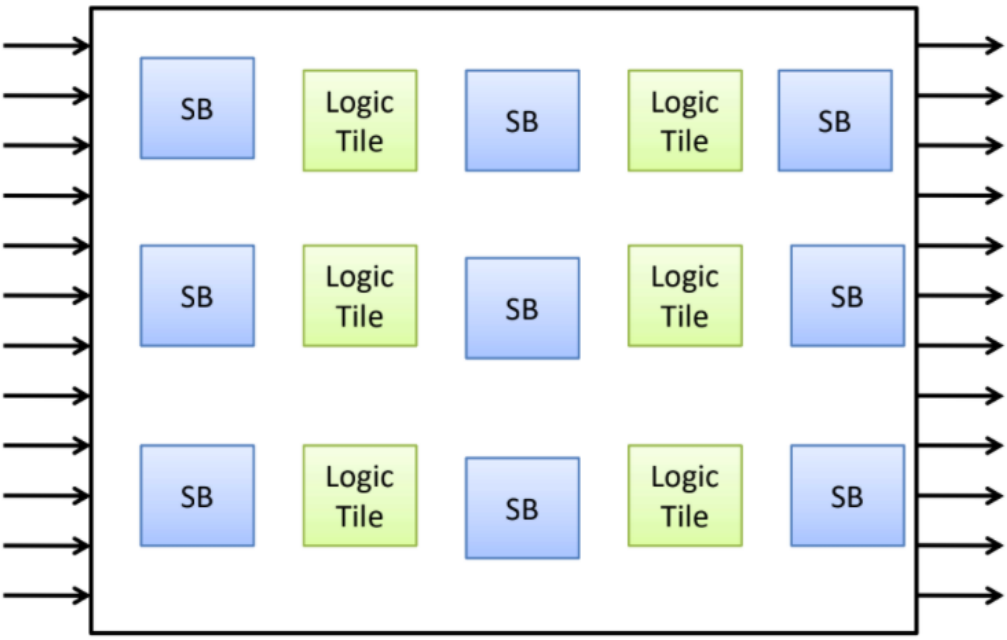
1) Logic Tile: - Logic tile contains LUT that is a one hot encoding for different set of inputs. It contains a flip flop that is connected to a clock for synchronous output.



2) Switch Box: - A switch box can take a configuration file and according to that it can route any input to any output.



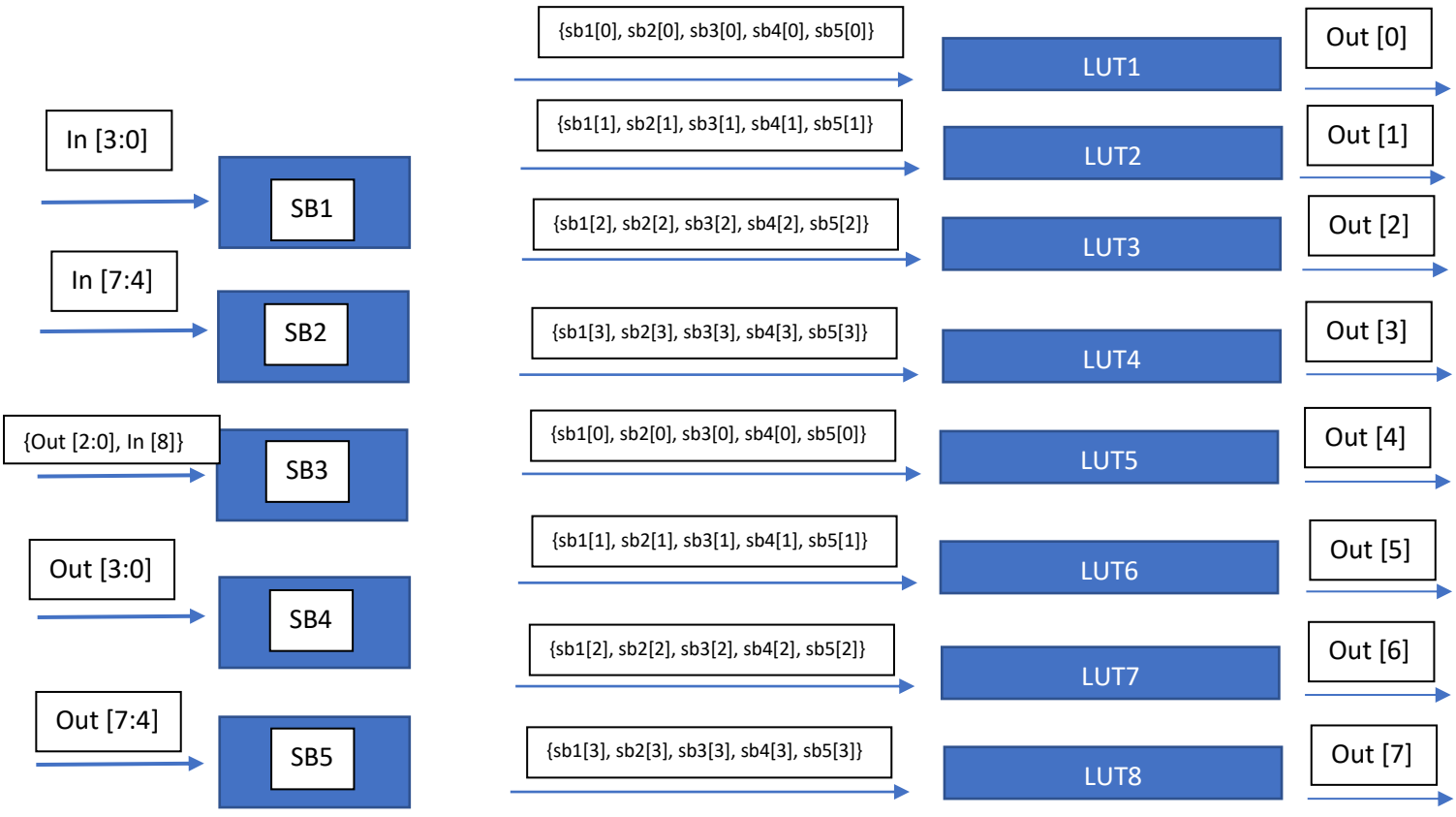
3) A configurable fabric: - Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing.



Implementation

1) We have to make our circuit generic so that we can implement any design which fits in the given input and output bit length
2) The designs we are thinking of implementing for this are
   i) 4-bit adder
   ii) 4-bit counter
   iii) 3-8 decoder
3) So, our maximum input requirement is 9 (for 4-bit adder, 2*4 input + 1 carry).
4) Maximum output requirements are 8 for the decoder.
5) So, in our design we need at least 8 logic tiles.
6) Now, each Logic Tile takes in 5 input so we need 8*5 outputs.
7) Each switchbox gives 4 output so we need 40/4=10 switchboxes.
8) Our input requirement is only 9 bits though. So we will take 5 switchboxes and give same output to a set of 4 Logic tiles.
9) What will we do about the rest of the inputs? We need to know what the previous output was, while implementing counters or adders. So, the rest of the inputs will be feedback loops from the output of the LUT.

Design



Making Config Files

1) Each input and corresponding output feedbacks are provided according to the above design.
2) Then a lookup table is made such that every possible input combination is routed to the output for that value and a one hot encoding is made for each circuit which gives the output accordingly.

References

1) Dr Neeraj Goel, Digital Logic Design