DIGITAL DESIGN AND COMPUTER ORGANISATION LABORATORY

PROJECT REPORT

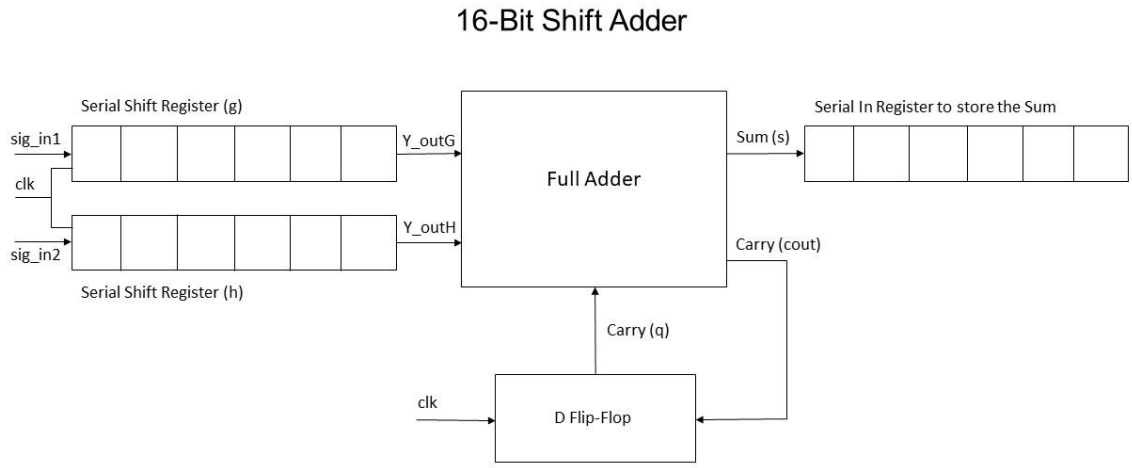
16 BIT SHIFT ADDER (SERIAL ADDER)

Avinash K Ratnam – PES1201800883
Harichandana Magapu – PES1201801041
Sanjna Chaturvedi – PES1201800722
Utsav M Jolapara – PES1201801086

ABSTRACT:

In this project, we attempt to simulate a 16-bit serial shift adder using Verilog. This adder has two input registers (to take two 16-bit binary numbers as input), a full adder (to add the numbers), and an output register (to store the 16-bit result).
As this is a serial adder, our input registers are SISO, i.e. Serial In Serial Out.
The full adder also has a d flip flop to store the carry of the previous operation (initially it is 0).
Our output registers are SIPO, i.e. Serial In Parallel Out. Serial In, because the input to the register comes from the full adder, one bit at a time. The input to the Full Adder, in turn comes from the SISO registers. Parallel Out because we want to access the result as a whole, not one bit at a time.

DIAGRAM:



IMPLEMENTATION:

To achieve the full functionality of the 16-bit serial shift adder, we have defined the following Verilog modules:

module shift(y,d,clk);
shift register to store the two inputs a and b to be added
y stores output, s is input, clk is clock

module sipo(y,s,clk);
serial in parallel out register to store the sum
y stores output, s is input, clk is clock

module fa(s,cout,a,b,cin);
full adder (adds two bits at a time, with carry)
s stores sum, cout stores carry out
a and b store input bits, cin stores carry in

module dff(q,d,clk);
d flipflop to store the cout of each stage
q stores output, d is input, clk is clock

module serial(sum,cout,a,b,clk);
main module
all other modules are instantiated in this module.
s stores sum, cout stores carry out
a and b store inputs
clk is clock