**Digital System design Lab CSE-308L**

**Semester:6th**



**Lab Report # 2**

**Submitted By:** *Zainab Khalid*

**Registration No:***19PWCSE1743*

**Section: A**

**Submitted to:** *Mam Madeeha Sher*

**Department of Computer Systems Engineering**

**University of Engineering and Technology Peshawar**

# LAB No 2

**INTRODUCTION TO MODELSIM AND GATE LEVEL MODELING**

**Objectives:**

This lab will enable students to:

* + Learn top down and bottom up design methodologies
  + Data flow level modeling
  + Data gate level modeling

**Data Gate Level Modeling.**

Designing circuits using basic logic gates is known as gate-level modeling. A digital circuit is implemented using logic gates and interconnections between these gates.

**Data Flow Level Modeling.**

Dataflow modeling uses a number of operators that act on operands to produce the desired results. Verilog HDL provides about 30 operator types like &, |, ^, ==, >, <,? etc. Dataflow modeling uses continuous assignments and the keyword ***assign***. It is called continuous assignment because it remains active all the time.

***assign out = a&b;***

The left hand side variable must be of type wire. At the right hand side, there can be a single variable or expression and of type reg or wire.

**Operators used:**

***Arithmetic:******+,-,\*,/***

***Logical:******&&, ||***

***Bitwise:******&, |, ~, ^***

***Reduction:******~|, ~&, ~^***

***Shift:******<<, >>***

***Equality****:* ***==, !=, (case equality: ===, !==)***

***Conditional: ? , :***

***Concatenation:******{}***

*Example: a=01;b=110; c={a,b};*

***Replication:******{{}}***

*Example: A=111111000000*

*x={6{1}}; y={6{0}};*

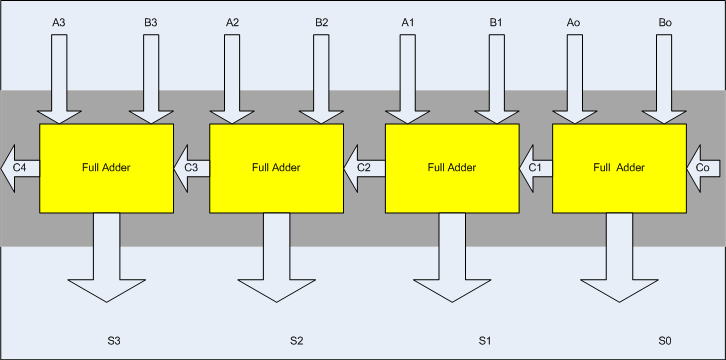
*A={x,y} or A={6{1},6{0}};*

**TASK 1:**

**4 bit Ripple Carry Adder**

**Block Diagram:**

Following is the block diagram of a 4 bit Ripple Carry Adder.

****

**I/O Connection:**

Ground “Co” Permanently and connect S0-S3 with four LEDs also connect C4 to another LED.

**Steps for task 1a:**

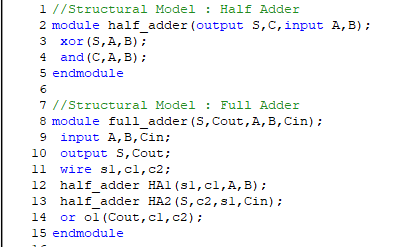
The following steps should be performed while designing a 4 bit RCA adder

**ModelSim:**

1. First implement a Full adder using data gate level modeling.

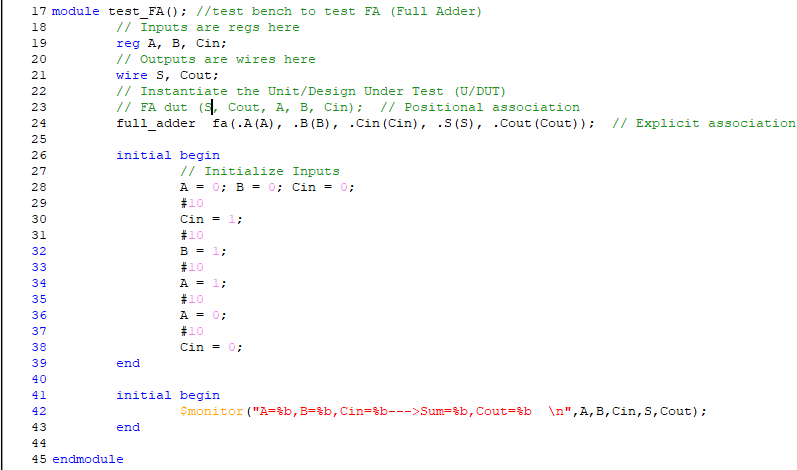
**Data Gate Level Modeling.**

**Code (Full Adder):**

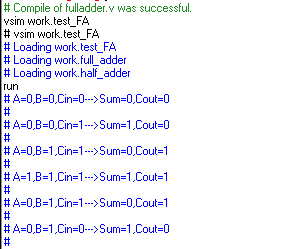


1. Simulate the Full adder with a test bench.

**Code (Test Bench For Full Adder):**

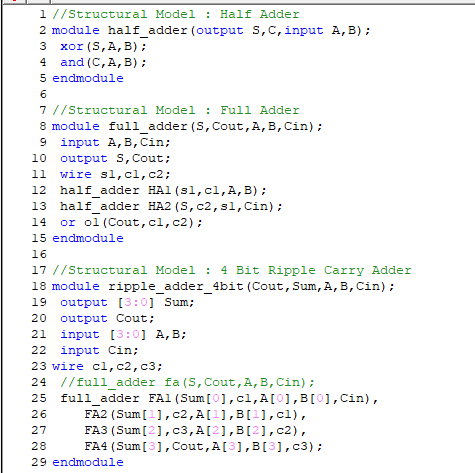


**Output (Full Adder):**



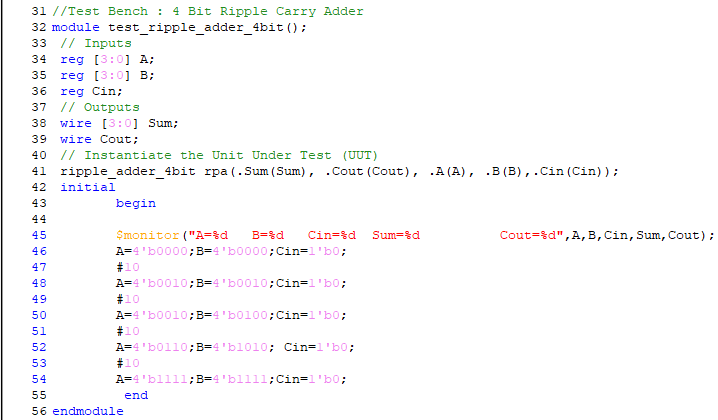
1. Instantiate the Full adder four times and connect the circuit as shown.

**Code (Ripple Adder):**

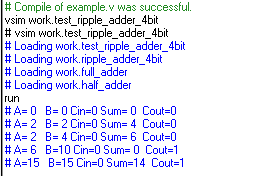


1. Now again write a test bench and simulate the 4 bit RCA.

**Code (Test Bench For Ripple Adder):**



**Output (Ripple Adder):**



**Xilinx:**

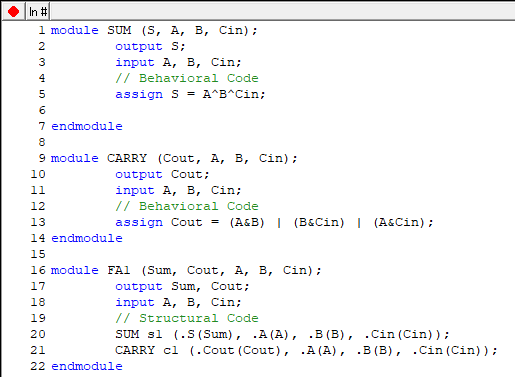
1. Make new project in Xilinx and add the files that you simulated in ModelSim.
2. Add User Constraint File inputs should be locked with the switches, C0 should be permanently “0” while S0-S4 and C4 with LEDS

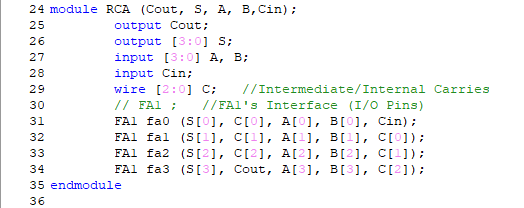
**Task 1b:**

Design the 4 bit full adder using data flow level modeling.

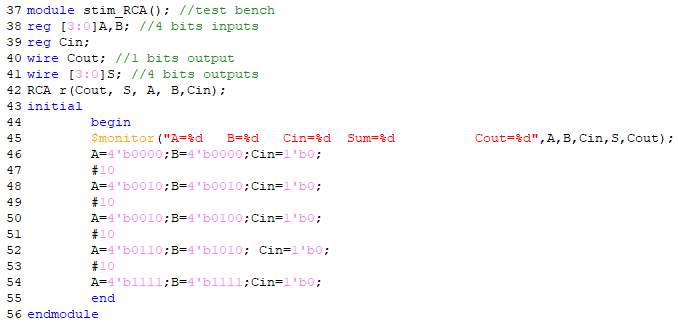
**Data Flow Level Modeling:**

**Code (Ripple Adder):**

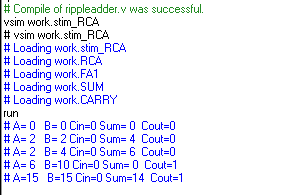




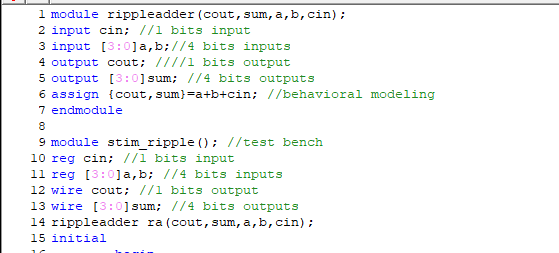
**Code (Test Bench):**

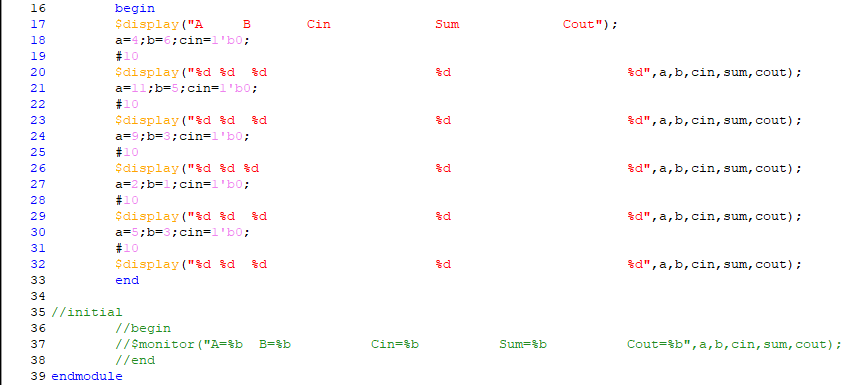


**Output:**



**2nd Code:**





**Output:**

