ASSIGNMENT SET – V

With universal logic gates, any Boolean function can implement without the need of any other logic gates. The NAND's and NOR 's are said to be universal logic gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate, and are universal logic gates used in all integrated circuits (IC) digital logic families.

Every computer is required to perform basic arithmetic operations like addition and subtraction as well as basic Boolean operations like bit-wise negation, bit shifting, and so forth. Here are some examples, written in typical machine language syntax:

ADD R_2, R_1, R_3 // $R_2 \leftarrow R_1+R_3$ where R_1, R_2, R_3 are registers

ADD R_2, R_1, foo // $R_2 \leftarrow R_1$+foo where foo stands for the value of the memory location pointed at by the user-defined label foo.

AND R_1, R_1, R_2 // $R_1 \leftarrow$ bit wise And of R_1 and R_2

A machine language program is a series of coded instructions. For example, a typical instruction in a 16-bit computer may be 1010001100011001. The language may be such that each instruction consists of four 4-bit fields: The left-most field codes a CPU operation, and the remaining three fields represent the operation's operands. Thus, the previous command may code the operation set R_3 to $R_1 + R_9$, depending of course on the hardware specification and the machine language syntax. Since binary codes are rather cryptic, machine languages are normally specified using both binary codes and symbolic mnemonics (a mnemonic is a symbolic label whose name hints at what it stands for, in our case hardware elements and binary operations). For example, the language designer can decide that the operation code 1010 will be represented by the mnemonic add and that the registers of the machine will be symbolically referred to using the symbols R_0, R_1, R_2, and so forth. Using these conventions, one can specify machine language instructions directly, as 1010001100011001.

In Assignment IV, you have clubbed ALU and the memory to implement complex mathematical functions using 16-bit integer numbers and fractions were discarded. The sequential chips that are used to implement the RAM in the assignment set III for developing the memory unit is used in building the current system. The only building blocks that you can use for developing the ALU of the current system are NAND gates and primitive Data flipflops (DFFs). Your computer should make use of 16-bit Program Counter (PC) as and when required.

The tool you need for this project is the ModelSim - Intel FPGA Starter Edition. All the chips should implement using HDL Verilog.

Please refer http://www1.idc.ac.il/tecs/book/chapter%2004.pdf for more details.

The objective of the assignment - V is to develop a 16-bit computer using Verilog. Your "Designed computer" should read the machine language instructions. Each instruction would be set of 16-bit binary string. Your system must carry out the processing of the entire program to do the following operations given below.

a. Multiplication of two numbers
b. Division of two numbers
c. Reminder of two numbers.