Indian Institute of Technology Kharagpur

AUTUMN Semester, 2020
COMPUTER SCIENCE AND ENGINEERING

Computer Organization Laboratory

Assignment-5: Verilog Design of Bit-sliced Arithmetic Logic Unit (ALU) and Fast Adders

Full Marks: 30

Time allowed: 9 hours

INSTRUCTIONS: Make one submission per group in the form of a single zipped folder containing your Verilog source code files(s) and Verilog testbench(es). Name your submitted zipped folder as `Assgn_5_Grp_<Group_no>.zip` and (e.g. `Assgn_5__Grp_25.zip`). Inside each submitted source and testbench files, there should be a clear header describing the assignment no., problem no., semester, group no., and names of group members. Liberally comment your code to improve its comprehensibility.

1. **[4-bit Combinational ALU/Functional Generator (74181)]** The 74181 is a classic commercially available 4–bit combinational ALU/function generator IC. Its main inputs are: two 4–bit operands $A[3:0]$ and $B[3:0]$, an 1–bit *mode control signal* M which controls the mode of operation ($M = 1$ for logic operations and $M = 0$ for arithmetic operations), and a 4–bit *function select* signal $S[3:0]$. Its main output is a 4–bit *data out* signal $F[3:0]$. In the *logic mode* of operation, i.e., with $M = 1$, the i-th bit of F is given by:

$$F_i = A_i^{'} B_i S_0^{'} + A_i^{'} B_i^{'} S_1^{'} + A_i B_i^{'} S_2 + A_i B_i S_3$$

From the above expression for F_i, it is evident 74181 can be used as a *universal function generator* for two Boolean variables, for different values of $S[3:0]$, i.e., it can be used to generate all two–variable Boolean functions $F(A, B)$. On the other hand, for $M = 0$. the circuits performs various arithmetic operations on the input operands A (see datasheet mentioned below for details).

Refer to the datasheet of 74181 uploaded on *Moodle* (consider the *Active HIGH Operand* version), particularly the function table given on page-2 of the datasheet. Design, simulate (using a proper testbench) and implement in Verilog (on a target FPGA supported by your CAD tool) the function table. Note that as per the manual, for the add/subtract operation, full 4-bit carry lookahead is supported, and this should also be the case in your implementation. Come up with a proper interface of your design referring to the logic diagram for the *Active HIGH* version, as given on page-1 of the datasheet. DO NOT DIRECTLY IMPLEMENT THE LOGIC DIAGRAM GIVEN ON PAGE-3 OF THE DATASHEET, YOU WILL GET ZERO MARKS IF YOU DO SO. (6 marks)

2. **[16-bit Bit-sliced ALU/Functional Generator Using the 74181 IC]** The 74181 has carry-in and carry-out ports. This makes it possible to concatenate multiple instances of the IC to design larger ALU/function generator IC. This type of design belongs to the *Bit-slice design paradigm*, and was common in the early days of digital ICs, when it was difficult to implement in silicon digital circuits that could process operands with large number of bits. Design, simulate (using a proper testbench) and implement in Verilog (on a target FPGA supported by your CAD tool), an 16-bit ALU/function generator circuit by concatenating four instances of the 74181 circuit, which you designed to solve Question-(1). (4 marks)

3. **[Carry-Select Adder]** Design, simulate (using a proper testbench) and implement in Verilog (on a target FPGA supported by your CAD tool), a Carry-Select Adder circuit to add two 16-bit numbers. (10 marks)

4. **[Carry-Save Adder]** Design, simulate (using a proper testbench) and implement in Verilog (on a target FPGA supported by your CAD tool), a Carry-Save Adder circuit capable of adding nine 16-bit numbers. Your design should consist of several Carry-Save Adder (CSA) blocks connected in a tree configuration, and a carry-lookahead adder in the last stage. (10 marks)