Birla Institute of Technology and Science Pilani
CS F342 Computer Architecture
First Semester 2021-11
Homework Exercise – 1 (31st August 2021)

Review of Digital Design

Instructions to the student:

1. There are six problems in this Homework.

The course computer architecture, has its foundation concepts rooted in two courses: Digital Design and Microprocessors. This Homework exercise is to revise the concepts the student has learnt in the course Digital Design. These are a few essential concepts which will help in building a basic processor. The reference to these topics and hints for solving the problems can be found in any good textbook on digital design.

Problem-1: A Ripple carry adder (RCA) is a basic adder circuit used to add two *n* digit binary numbers A and B and obtain a (*n+1*) digit SUM. Can you recall the structure of a RCA built only using Full-adders? If the propagation delay of a Full-adder in generating the *sum* and *carry* signals is T_{sum} and T_{carry} respectively, can you express the delay of a *n*-bit RCA in terms of T_{sum} and T_{carry} ?

Problem-2: Digital computers use two's complement arithmetic to deal with signed numbers. One common problem associated with this notation is the *overflow/underflow.* Can you give examples for each of these cases where the inputs are two signed 8-bit numbers? Come up with a combinational circuit built using standard logic gates to detect the condition in a practical implementation.

Problem-3: Revise the Radix-2 Booth's Algorithm and show the step by step procedure for calculating the product of two 8-bit signed numbers -25_{10} and $+43_{10}$. Verify your answer. What is the advantage in using a Radix-2 Booth's method over the conventional *shift-add* multiplier architecture?

Problem-4: Revise the concepts of Restoring and Non-Restoring algorithms. Perform the division operation (using both the algorithms) on two numbers $+48_{10}$ divided by $+17_{10}$. Division is one of the most complex (large area) and time consuming process in a digital computer. Can you come up with a few suggestion/s to improve this process.

Problem-5: Implement* a synchronous BCD counter which can count from 0-9 repeatedly in a loop. This synchronous counter has an active high RESET signal and works with a positive edge triggered CLOCK. You are free to choose any Flip-flop of your choice and any combination of logic gates.

Problem-6: Implement* a ***BCD up counter.*** The counter has a two-digit output (remember each BCD digit is 4-bit wide and valid BCD digits are from 0 to 9 only). The counter counts from “00” to “59” if it is counting in ***up*** mode. Choose a positive edge CLOCK and asynchronous active high RESET for this counter. Example output:

00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13 .... 59, 00,01 ...

Here, while implementing, reuse the counter you have designed in Problem-5.

***Implement** here means to draw a gate/system level diagram showing all the connections. HDL code is not expected in this Homework.*

******