Indian Institute of Technology (IIT-Bombay)

AUTUMN Semester, 2024
COMPUTER SCIENCE AND ENGINEERING

CS231: Digital Logic Design + Computer Architecture Lab

Lab 1

Full Marks: 100

Time allowed: (3 + ϵ) hours

1. Consider the following combinational circuit for *BCD-to-Seven-Segment* display. The truth table is provided. This is one of the most fundamental circuits that is used in most of the embedded boards that you see. You can also see some versions in our stations (even today !!!).

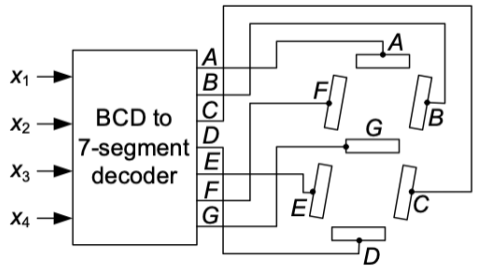
   The purpose of this lab is to help you understand why Verilog is called a hardware description language, even though many of the behavioural descriptions will just look like statements in C. Therefore, you are required to write both the structural and the behavioural descriptions of the same circuit.

   1. Construct the output logic equations using K-Map. You do not need to submit the K-Maps or the equations. Instead, realize them in your Verilog code.
   2. Write a structural verilog code to realize the circuit. Your circuit must use Verilog AND, OR, NOT gates.
   3. Realize the same logic with Verilog *assign* statements.
   4. Write a testbench in Verilog to exhaustively test your circuit with all possible input combinations. You must submit the design, the testbench and the (clear) screenshot of the waveform in GTKWave. Your testbench must have $*monitor* statements for monitoring all input and output signals. Also, it must have a $*display* statement to print your name and roll number.
   5. **Format:**
      (a) Roll_bcd_to_7_segment_struct.v, Roll_tb_bcd_to_7_segment_struct.v, Roll_waveform_struct.png
      (b) Roll_bcd_to_7_segment_behv.v, Roll_tb_bcd_to_7_segment_behv.v, Roll_waveform_behv.png

      You may make a .zip/.tar archive and submit it.

Seven-Segment Display

Seven-segment display: BCD to seven-segment decoder and seven LEDs



Seven-segment pattern and code:

Decimal	*BCD code*				*Seven-segment code*						
digit	x_1	x_2	x_3	x_4	A	B	C	D	E	F	G
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
0	0	0	0	0	1	1	1	1	1	1	0

Figure 1: K-Map for question 3