Home Work Assignment - 1

Q1. Using postulates,

a) find mimimum SOP expression for the function,
$\boldsymbol{F\ A,B,C = A \cdot C + A \cdot B! + A! \cdot B \cdot C + A' \cdot B' \cdot C'}$

b) Prove that $\mathbf{B \oplus\ A \cdot B + B \cdot C + A! \cdot C = A! \cdot (B \oplus C)}$

Q2. The ***Chamber of Secrets*** at the ***Hogwarts School of Wizardy*** needs to be opened. ***Professor Dumbledore*** announces that it needs two students from the ***Gryffindor house*** (*G1, G2*) and a one from the ***Slytherin house*** (*SL*) to open the **chamber**. The chamber can be opened on working days either by the ***Gryffindor students*** together or by the ***Slytherin student***. On Saturdays, any one of the ***Gryffindor*** and the ***Slytherin*** can together open the **chamber**, while on Sundays, all three have to be present to open the **chamber**. Assign binary variables *X, Y and Z* to indicate a working day, Saturday and Sunday respectively, and obtain a Boolean expression for the condition to be satisfied for opening the chamber (*C*) in terms of the binary input variables *G1, G2, SL, X, Y and Z.*

Q3. A robot has four permitted directions of movement and three possible speed settings. Let the direction control and the speed control be applied through two bits each:

D1D0 = 00(forward) / 11(reverse) / 01(right) / 10(left), and
S1S0 = 00(zero) / 01(low) / 10(medium) / 11(high).

It is desired to have an electronic protection system to ensure that the robot can move at high speed for forward movement only, and reverse only at low speed. This will have to be achieved by generating an output bit P which should go HIGH if any of these two conditions is violated, and then using P to shut off the power to the robot. e down the Boolean expression for the output Pin the sum of products form in terms of D1, D0, S1, S0.

Q4. A IIIT student staying in one of the institute hostels has to make up his/her mind about his/her dinner. There can be three situations which may arise :

Situation 1 - If he/she has enough money (M) and at least three of his/her friends (F) also agree to go out for dinner, and it is not raining (R), he/she will have dinner with his/her friends in a restaurant at Civil Lines.

Situation 2 - If he/she is not able to go out, but at least three of his/her friends agree to join him/her (J), if the kind of food he/she wanted is available on home delivery (K) he/she will order home delivery of food from online food delivery services, Swiggy maybe.

Situation 3 - But, if the general feeling is that the food in the hostel mess is good on that day (G), he/she will have his/her dinner in the hostel mess.
Let his/her decision be denoted by a 2-bit output D1 D0:
D1 D0 = 00 : He/She eats in the hostel mess,
D1 D0 = 01 : He/She goes out to have dinner in a restaurant,
and D1 D0 = 10 : He/She orders food through home delivery app.
Assign binary variables to represent the three situations using the letters indicated in parentheses above, and obtain both POS expressions for D1 and D0 in terms of these variables, simplifying the expressions as far as possible.

Q5. (a) Represent -4 as a 4-bit and 8-bit two's complement number.
(b) Subtract 4 from 3 using 2's complement arithmetic. Consider both the numbers of size 4bits.Does the arithmetic operation operation generate overfow?

Q6. Write -6.125 in IEEE 754 single precision. Label all bits properly (sign bit, exponent, fraction).
(b) How can you tell if a 32 bit number is NaN. That is, describe what bits you look at to determine that it's NaN.
(c) Around 250 B.C., the Greek mathematician Archimedes proved that **223/71 < pi < 22/7**. Had he had access to computer and the standard library **<math.h>**, he would have been able to determine that the single-precision oating-point approximation of pi has the hexadecimal representation **0x40490FDB**. Of course, all of these are just approximations, since pi is not rational. What is the fractional binary number denoted by this oating-point value?

Q7. For the following problem, you will write code to implement floating-point functions, operating directly on bit-level representations of floating-point numbers. Your code should exactly replicate the conventions for IEEE floating-point operations, including using round-to-even mode when rounding is required.
Toward this end, we define data type float_bits to be equivalent to un-signed:

```
/* Access bit-level representation floating-point number */
typedef unsigned float_bits;
```

Rather than using data type float in your code, you will use float-bits. You may use both int and unsigned data types, including unsigned and integer constants and operations. You may not use any union, structs or arrays. Most significantly, you may not use any floating-point data types, operations or constants. Instead, your coade should perform the bit manipulations that implement the specified flating-point operations.

The following function illustrates the use of these coding rules. For argument f, it returns plus/minus 0 if f is denormalized (preserving the sign of f) and returns f otherwise.

```
/* if f is denorm, return 0. Otherwise, return f */

float_bits float_denorm_zero(float_bits f) {
     /* decompose bit representation into parts */
          unsigned sign = f >> 31;
          unsigned exp = f>>23 & 0xFF;
          unsigned frac = f & 0x7FFFFF;
          if (exp == 0) {
     /* denormalized. Set fraction to 0 */
          frac = 0;
     }
```

```
/* reassemble bits */
return (sign << 31) | (exp << 23) | frac;
}
```

Implement the function with the following prototype:

```
/* compute -f. if f is NaN then return f */

float_bits float_negate(float_bits f);
```

For floating-point number f, this function computes –f. if f is NaN, your function should wimply return f. Test your function by evaluating it for all 232 values of argument f and comparing the result to what would be obtained using your machine's floating-point operations.