Quiz # 1

Sept. 26, 2:35 - 3:40pm, in class

Books are allowed. Notes are allowed. Calculators are allowed – but only for arithmetic. Laptops and smart phones are not allowed.

1. **CMOS transistor circuits.**

   (a) Write the truth table for the output Y as a function of the inputs A, B, C implemented by the circuit in Figure 1.

   (b) For the circuit in Figure 2, for which input combination is there a short circuit? For which input combination is the output Y floating?

   (c) Draw a CMOS transistor circuit that computes the function $Y = \text{OR}(A, \text{NAND}(B, C))$

2. **Truth Tables from Logic Gates.**

   (a) Write the truth table for the Boolean function Y as a function of inputs A, B, C in Figure 3.

   (b) Write the truth table for the Boolean functions $f_1, f_2, f_3, f_4, f_5, f_6$ as a function of inputs x_1, x_2, x_3 in Figure 4.

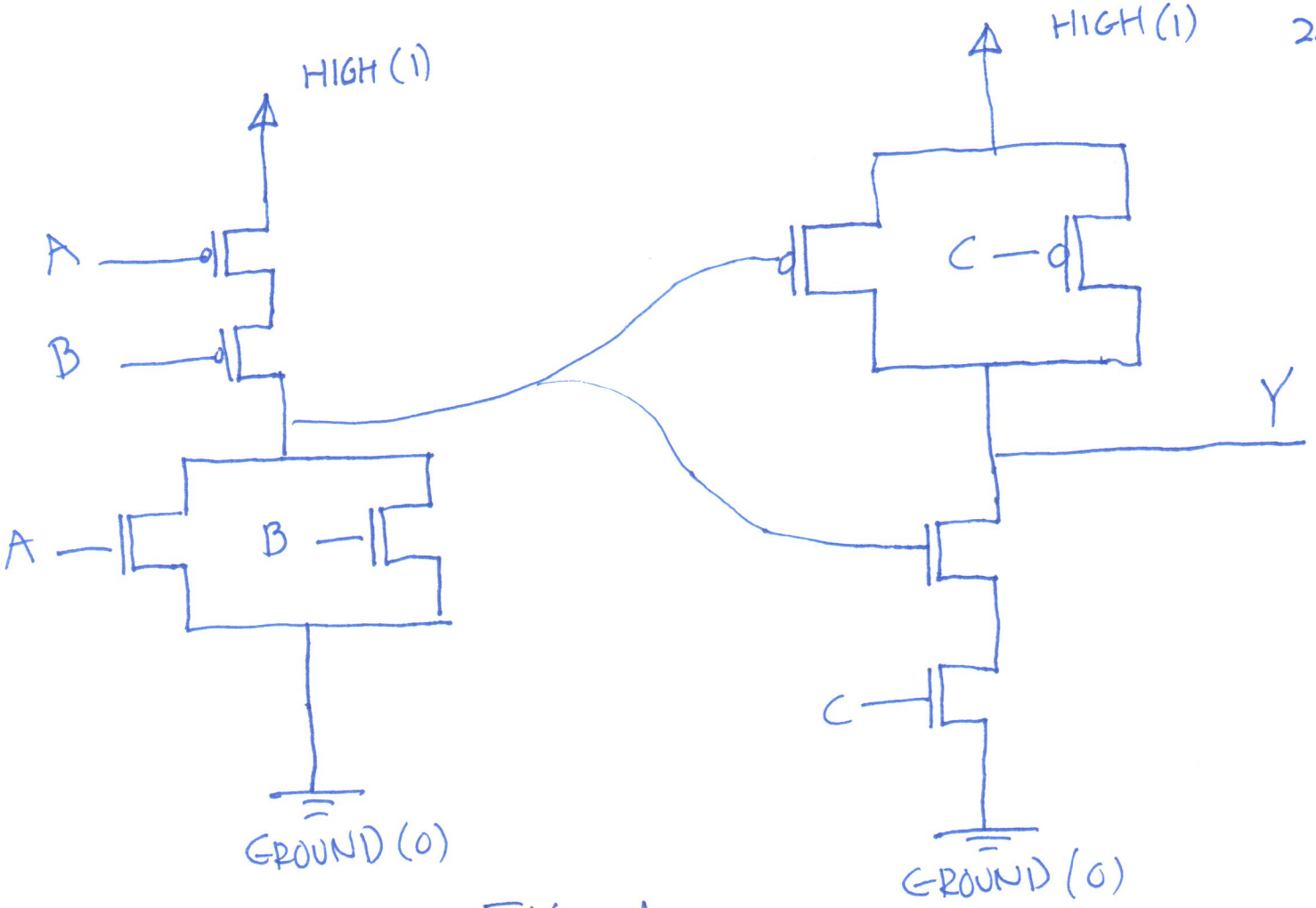

FIG. 1

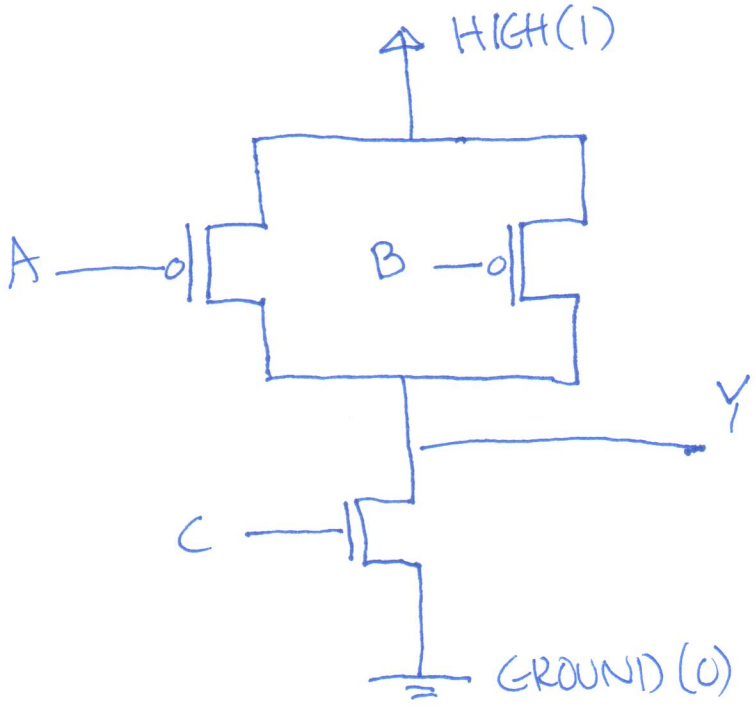

FIG. 2

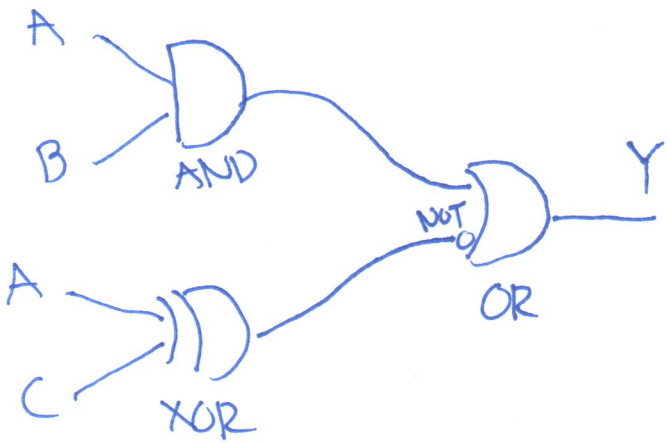

FIGURE 3.

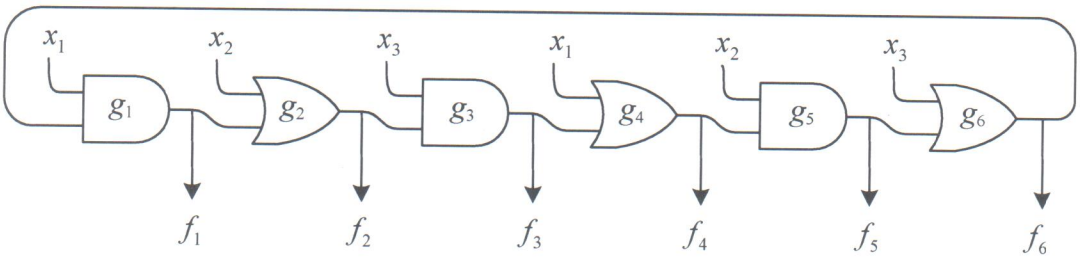

FIGURE 4.

3. **Recursive C Program.**

Cyclic Hanoi is a variation of the Tower of Hanoi problem in which each disk must be moved in the same cyclic direction, in most cases, clockwise. For example, given a standard three peg set-up, a given disk can be moved from peg A to peg B, then from B to C, C to A, etc. What will the following code print if called with the arguments listed at the bottom?

```
# include <stdio.h>

// move from a to c using b
int hanoi(int n, int a, int c, int b) {
   if (n == 1) {
      printf("Move disk from %d to %d\n", a, c);
   } else {
      // move from a to c using b
      hanoi(n - 1, a, c, b);
      printf("Move disk from %d to %d\n", a, b);
      // move from c to a using b
      hanoi(n - 1, c, a, b);
      printf("Move disk from %d to %d\n", b, c);
      // move from a to c using b
      hanoi(n - 1, a, c, b);
   }
}

int main(int argc, char **argv)
{
   int n = atoi(argv[1]);
   hanoi(n, 1, 2, 3);
}

./hanoi 3 1 2 3
```

4. **Iterative C Program.**

   What will the following C program print out?

```c
# include <stdio.h>
int fact(int n)
{
   int i, p;
   p = 1;
   for (i = 1; i <= n; ++i)
      p = p * i;
   return p;
}

int main ()
{
   int i, j, sum1, sum2;
   for (i = 1; i <= 5; ++i) {
      sum1 = 0;
      for (j = 1; j <= i; ++j) {
         sum1 = sum1 + (j - 1) * fact(j);
      }
      sum2 = 3*fact(i) - 1;
      printf("%d %d %d\n", i, sum1, sum2);
   }
}
```