True or False
Write T for True or F for False in the spaces provided.

1. Different assembly language instructions are needed to perform two’s-complement and unsigned integer addition.
2. The x86 architecture is little-endian.
3. The decimal value 32 in binary is 11111.
4. xorl %eax, %eax sets the Zero Flag to true.
5. A P-type transistor is switched on when a zero is placed on its source terminal.
6. An N-type transistor is good at passing zeros, but bad at passing ones.
7. A ripple-carry adder circuit that adds two $n$-bit numbers takes time proportional to $O(\log n)$.
8. In x86 assembly language, the %esp register is often used as the return value from functions.
9. All instructions encoded in x86 machine language are four bytes long.
10. X86 assembly language supports an addressing mode with a base address, an index register, and scaling factors of 1, 2, 4, or 8.
CMOS Circuit

11. Consider the following circuit diagram.

Give a truth table for the Boolean function of \( a \) and \( b \) computed by this circuit.
Bitwise and Logical Operations and Endian-ness

12. Consider the following code fragment:

```c
unsigned int x = 0x89abcdef;
unsigned int y = 0x76543210;
unsigned int z = (x & 0xff) | (y & 0xffffffff);
unsigned char *p = (unsigned char *) & z;
printf ("%x\n", (unsigned int) *p);
```

Recall that %x means “print hexadecimal” for printf.

(a) What is the value of z at the end of this code?

(b) What is the output of this code on a little-endian machine?

(c) What is the output of this code on a big-endian machine?

(d) If we replaced the | operator with ||, what would be the value of z at the end of this code?
Truth table, Karnaugh Map, Circuit Diagram

13. Show how to build an efficient three-input majority circuit in three steps:

(a) Write the truth table for the majority function.

(b) Derive a minimal sum-of-products for the majority function using a Karnaugh map.

(c) Draw the circuit diagram for the three-input majority function using only inverters and NAND gates. Use no more gates than necessary. **Note:** if you can’t figure out how to do it with just inverters and NANDs, you may also use ANDs and ORs for partial credit.