1.1. Represent the following numbers as directed:

a. 12 as a 4-bit unsigned number **1100**
b. 12 as a 5-bit unsigned number **01100**
c. +1 as a 4-bit 2's complement number **0001**
d. –1 as a 4-bit 2's complement number **1111**
e. –1 as a 5-bit 2's complement number **11111**
f. +1 as a 4-bit signed-magnitude number **0001**
g. –1 as a 4-bit signed-magnitude number **1001**

1.3. What is the 16-bit FP number representation of –5.375 in hex with 1-bit sign, 4-bit biased exponent, and 11-bit fraction, where bias = 7?

1011.0110 → $1.0110110 * 2^{10}$ → 1, 1010, 0110110

(1, 1010, 01101100000) = 0xd360

1.5. What is the real number equivalent to FP number 0x3400 with 1-bit sign, 4-bit biased exponent, 11-bit fraction, and bias = 8?

0x3400 → 0,0110,10000000000 → bias_exp = 6 so bias = -2 → 0000.0110 = **0.375**

1.7. What is the biggest positive FP number that can be represented in 16-bit format using 1-bit sign, 4-bit biased exponent, and 11-bit fraction, where bias is 8? **6**

1.8. Do the following assuming 16-bit FP numbers with 4-bit bias exponent, bias = 7, and 11-bit fraction:

a. What real number does an FP number with sign = 0, bias exponent = 1, and fraction = 0 represent?

0,0001,00000000000 → 0000.00000100 → 1/64 → **0.015625**

b. What real number does an FP number with sign = 1, bias exponent = 14, and fraction = (11111111111)2 represent?

1, 1110, 11111111111 → 11101111.11111110 → **-31.9953125**

1.9. Represent the following real numbers as 16-bit FP numbers with 4-bit biased exponent, bias = 7, and 11-bit fraction:

a. 1.0 **(0, 0111, 00000000000) = 0x3800**
b. 0.5 **(0, 0110, 00000000000) = 0x3000**
c. 0.25 **(0, 0101, 00000000000) = 0x2800**

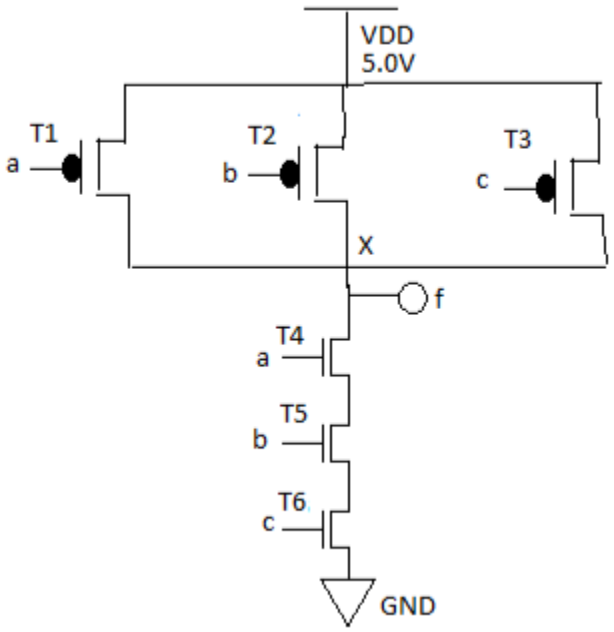
1.10. Do the following assuming 16-bit FP numbers with 4-bit bias exponent, bias = 8, and 11-bit fraction:

a. What real number does an FP number with sign bit = 0, bias exponent = 1, and fraction = 0 represent?

(0,0001,00000000000) → 0000.0000001 → 1/128 → **0.0078125**

b. What real number does an FP number with sign bit = 1, bias exponent = 14, and fraction = (11111111111)2 represent? (1, 1110, 11111111111) → 11110111.1111 → **-8.05**

1.15. Draw a transistor-level schematic of a three-input CMOS NAND gate and determine its truth table in terms of transistor ON and OFF positions.



1.18. What is the difference between pipelining and parallelism architectures? Identify their application areas.

Pipelining is the concept of organizing a CPU's data path into stages to execute programs faster (like an assembly line). Pipelining is most useful when tasks are dependent upon a prior task's completion. Parallelism is the concept of executing tasks in parallel, but only truly works when tasks are not reliant on the output/result of other tasks.

1.21. Explain why a further increase in performance comes from parallel processing.

You can combine parallelism architecture and pipelining to execute nondependent tasks at the same time in different registries/on different processors, then stagger the execution of dependent tasks (still in parallel) to maximize efficiency. It's like having four sets of assembly lines instead of just one assembly line set.