Computer Organization Test 1

Question 1. (20 points) For the Boolean function F specified in the following truth table (a "d" indicates a "don't care"), use a K-map to write the simplified sum-of-products (SOP) Boolean function.

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	d
0	1	1	0	1
0	1	1	1	1
1	0	0	0	d
1	0	0	1	d
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	1
1	1	1	0	d
1	1	1	1	d

b) Implement your above simplified Boolean function for F using AND, OR, and NOT gates. (Alternatively, if you had problems with part (a), you may implement the *unsimplified* Boolean function F. Treat the "don't cares" as "0"s).

c) Ignoring NOT gates, how many gate delays are in your above circuit?

d) What is the complexity (# of gates + # of inputs to those gates) of your circuit?

Question 2. (20 points)

a) Convert 106_{10} to a binary (base 2) value.

b) Convert 106_{10} to a hexadecimal (base 16) value.

c) Covert -106_{10} to a two's complement value.

d) Perform the following arithmetic operations:

1001011_2	1100110_2	$9\ 8\ 8\ E\ 3_{16}$	$9\ 8\ D\ C\ 5_{16}$
$+\underline{1100110_2}$	$-\ \underline{0101011_2}$	$+\ \underline{4\ C\ 9\ A\ 9}_{16}$	$-\ \underline{4\ 9\ 5\ E\ 9}_{16}$

Question 3. (10 points)

a) Convert the value 73.3125_{10} to its binary representation.

64 32 16 8 4 2 1 . .5 .25 .125 .0625

[ ] . [ ]

b) Normalize the above value so that the most significant 1 is immediately to the left of the radix point. Include the corresponding exponent value to indicate the motion of the radix point.

$1.$[ ] $\times 2^{[\ ]}$

c) Write the corresponding 32-bit IEEE 754 floating point representation for 73.3125_{10}.

Sign bit	8-bit Exponent (bias 127)	23-bit Mantissa (for normalized values, leading 1 not stored)

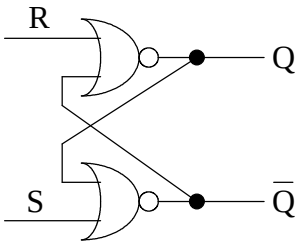
d) What is the purpose of a denormalized floating point number?

e) What is the purpose of “NaN,” not-a-number?

Question 4. (10 points) Consider adding $1.011_2 \times 2^{40}$ and $1.01_2 \times 2^5$.
a) How many places does the second number's mantissa get shifted?

b) After we add these two numbers and store the results back into a 32-bit IEEE 754 value, what would be the resulting sum?

Question 5. (10 points) a) If R = 1 and S = 0, then what will be the output on Q and $\overline{Q}$?



b) Now, if R goes to a 0 value, what happens to the output on Q and $\overline{Q}$?

Question 6. (10 points) If we consider implementing a 32-bit adder by "rippling" together smaller adders, then we get the following gate delays:

Type of Adders Used	Number of Adders Needed	Gate Delay per Adder	Total Gate Delays for 32-bit Adder
one-bit adders	32	2	64
two-bit adders	16	2	32
four-bit adders	8	3	24

a) Explain why the gate delay of a two-bit adder is the same as the gate delay for a one-bit adder.

b) Explain why the gate delay of a four-bit adder is more than the gate delay for a two-bit adder.

Question 7. (10 points) Suppose we have a register file with the following specifications:

- 16 registers numbered from R0 to R15
- each register has 64-bits
- one write ports
- two read ports

a) How many bits(/"wires") would be need for each of the following?

  - specifying the register number for either a read or write port
  - data output read from a read port

b) How many decoders would be needed in the implementation of the whole register file?

c) What type of decoders (i.e., number of inputs and number of outputs for each decoder) are needed?

d) How many multiplexers would be needed in the implementation of the whole register file?

e) What type of multiplexers are needed?

Question 8. (10 points) The register-file implementation does not scale well for large RAM memories because the number of gates in the address decoder (and MUXs) grows exponentially with the number of bits in the address. Therefore, large RAM memories use a square-array of bits and decode the address in two parts (row number then column number). For example, a 4 M x 4-bit memory would use two 11-to2^{11} decoders instead of one 22-to-2^{22} decoder needed for a register-file implementation.
a) How many gates are saved using two 11-to-2^{11} decoders instead of one 22-to-2^{22} decoder?

b) Why do RAM memory chips use dynamic memory (capacitor based) instead of static memory (SR-latch based) eventhough dynamic memory has the drawback that it needs to be refreshed?