1. (6 points) Using 40 bits, how many different integers can be represented in

   (a) binary \(2^{40}\)
   (b) BCD \(10^{10}\)
   (c) 8-bit ASCII \(10^5\)

2. (12 points) The simulation output on the next page is from an execution of the simple processor introduced in section 1 of the lecture notes. The instruction set for the processor appears on the next page with the simulation output. Answer the following questions using the simulation output.

   What is the address of the instruction being executed starting at time 4.09 \(\mu s\)? 000F

   What is the signed decimal equivalent of the value stored in the accumulator at time 4.25 \(\mu s\)?
   \[\text{FFF6}_{16} = -10\]

   What is the value stored in location 000a at time 4.75 \(\mu s\)? \(4041_{16}\)

   At what location is there an indirect load instruction? From what location in memory does the value loaded into the accumulator come from? 0013, 0004

   Which component of the computer is putting a value on the data bus at time 4.27 \(\mu s\)? memory
0xxx load ACC with value stored in memory word xxx
1xxx store value in ACC into memory word xxx
2xxx add value in memory word xxx to value in ACC
3000 negate the value in ACC
2xxx add value in memory word xxx to value in ACC
3001 halt
4xxx change the value of PC to xxx
5xxx if the value of ACC is zero, change PC value to xxx
6xxx load ACC with value whose address is stored in word xxx
7xxx store ACC value into word whose address is in word xxx
8xxx change ACC value to xxx
9xxx add xxx to the value in ACC
3. (8 points) Write a Boolean expression for each output of the circuit shown below and give a truth table for it.

\[ X = AB + D \]

\[ Y = X + C' \]

\[ (C' + D)' \]

<table>
<thead>
<tr>
<th>ABCD</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0011</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0101</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0110</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0111</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1010</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1011</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1100</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1101</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1110</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1111</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4. (8 points) For each function given below, use a Karnaugh map to find the simplest sum-of-products expression for the function. Take full advantage of the don’t care conditions.

(a) \( F(X,Y,Z) = \Sigma m(1,2,5) \), \( d(X,Y,Z) = \Sigma m(3,6) \)

\[ F = Y'Z + YZ' \]
(b) \( F(A,B,C,D) = \Sigma m(1,2,3,6,8,9,12) \), \( d(A,B,C,D) = \Sigma m(4,7,11,14,15) \)

\[
\begin{array}{cccc}
CD \\
00 & 01 & 11 & 10 \\
00 & 1 & 1 & 1 & 1 \\
01 & x & 0 & x & 1 \\
11 & 1 & 0 & x & x \\
10 & 1 & 1 & x & 0 \\
\end{array}
\]

\[ F = B'D + A'C + AC'D' \]

5. (6 points) Use a Karnaugh map to find the simplest \textit{product-of-sums} expression for the function \( F(A,B,C,D) = \Sigma m(0,2,3,8,12,14) \), \( d(A,B,C,D) = \Sigma m(4,6,7,9,13) \).

\[
\begin{array}{cccc}
CD \\
00 & 01 & 11 & 10 \\
00 & 1 & 0 & 1 & 1 \\
01 & x & 0 & x & x \\
11 & 1 & x & 0 & 1 \\
10 & 1 & x & 0 & 0 \\
\end{array}
\]

\[ F' = CD + AD + AB'C \]

\[ F = (C+D')(A'+D')(A'+B+C') \]
6. (14 points) Consider the circuit shown below. Assume that all gates have a minimum propagation delay (for both rising and falling edges) of 1 ns, and a maximum propagation delay of 2 ns. In the timing diagram shown below, fill in the waveforms for $K$ and $L$, showing both min and max delay waveforms with shading to indicate time periods when the state of the signal is not known.

$K = EF + H$

$L = K + E'(GH)'$
7. (10 points) Give the Boolean equations for the outputs of a full-adder.

\[ S = A \oplus B \oplus C_i \quad C_o = AB + BC_i + AC_i \]

Draw a logic diagram for a full-adder.

Give a block diagram of a 4 bit adder using full-adders. Be sure to label all the inputs and outputs to your full adder blocks.
8. (10 points) The logic diagram below shows a 5 bit ripple-carry decrement circuit. Draw a logic diagram for a 5 bit borrow-lookahead decrement circuit. You may use gates with more than 2 inputs. What is the worst-case propagation delay for your circuit, if all gates with 1 input have a delay of 1 ns, all gates with 2 inputs have a delay of 2 ns, all gates with 3 or 4 inputs have a delay of 4 ns and all gates with 5 to 8 inputs have a delay of 6 ns?

The max delay is from the DECR input through the 5 input OR gate to Q4. This delay is 1+6+2=9 ns.
9. (6 points) Consider the schematic and simulation output shown below. What was the simulation precision used to produce this simulation output? Explain how you know.

It’s a unit delay simulation, so each gate gets the same delay. From the schematic, we see that when V drops at 3.5 ns, this is caused by the drop of Q and R at 3.2 ns. Since the path length from Q and R to V is 3 gates, and since the delay is .3 ns, each gate is being treated as having a .1 ns delay. This means that the simulation precision must be .1 ns.
10. (10 points) Draw a logic diagram using simple gates and a 4:1 multiplexor that is equivalent to the VHDL code shown below. Be sure to label all the inputs of your multiplexor correctly.

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity foo is
  port (curve, slider, wild: in std_logic;
        sign: in std_logic_vector (1 downto 0);
        strike, run: out std_logic);
end foo;

architecture bar of foo is
begin
  strike <= (curve or slider) and not wild;
  run <= (curve xor slider) when sign = "00" else wild when sign = "01" else slider and wild;
end bar;
```

![Logic Diagram](image-url)