1. (10 points) List the minterms and maxterms for the expression $A'B + (A + B')C + (A+C)'B$.

Express the complement of $A'B + (A + B')C + (A+C)'B$ in the simplest form you can using Boolean algebra. Show your work.
2. (10 points) For each VHDL code fragment shown below, write an equivalent set of simple signal assignments (each involving just a single signal, not a vector of signals).

(a) \[a \leftarrow '1'; \ b \leftarrow x \text{ xor } z;\]
    \[
    \text{case } c \text{ is}
    \begin{align*}
    \text{when } "00" & \Rightarrow a \leftarrow x \text{ or } y; \\
    \text{when } "01" & \Rightarrow b \leftarrow y; \\
    \text{when others} & \Rightarrow a \leftarrow c(1);
    \end{align*}
    \text{end case;}
\]

(b) \[x \leftarrow c;\]
    \[
    \text{for } i \text{ in } 0 \text{ to } 2 \text{ loop}
    \begin{align*}
    &z(i) \leftarrow a(i) \text{ xor } x; \\
    &x \leftarrow (\text{not } a(i)) \text{ or } x;
    \end{align*}
    \text{end loop;}
\]
3. (10 points) The simulation output shows selected signals from the processor introduced in section 1 of the course notes. The portions of the output corresponding to four different instructions are outlined. Note that some parts of the simulation output have been blanked out. For the labeled blanks, fill in the missing information below.

A. __________________________
B. __________________________
C. __________________________
D. __________________________
E. __________________________

0000    halt – halt execution
0001    negate – \( ACC := -ACC \)
1xxx    immediate load – if sign bit of xxx is 0 then \( ACC := 0xxx \) else \( ACC := fxxx \)
2xxx    direct load – \( ACC := M[0xxx] \)
3xxx    indirect load – \( ACC := M[M[0xxx]] \)
4xxx    direct store – \( M[0xxx] := ACC \)
5xxx    indirect store – \( M[M[0xxx]] := ACC \)
6xxx    branch – \( PC := 0xxx \)
7xxx    branch if zero – if \( ACC = 0 \) then \( PC := 0xxx \)
8xxx    branch if positive – if \( ACC > 0 \) then \( PC := 0xxx \)
9xxx    branch if negative – if \( ACC < 0 \) then \( PC := 0xxx \)
axxx    add – \( ACC := ACC + M[0xxx] \)
dxxx    and – \( ACC := ACC \ and \ M[0xxx] \)
4. (10 points) Consider the circuit shown below. Suppose this circuit is implemented directly, using CMOS, with each AND gate implemented using a NAND and an inverter (similarly for OR gates). What is the worst-case delay for the circuit, assuming that NAND and NOR gates have a delay of 2 ns each, and inverters have a delay of 1 ns? Highlight the path through the circuit that accounts for the worst-case delay.

Show an alternate implementation, using only NANDs, NORs and inverters that reduces the delay by at least 2 ns.

Compare the number of transistors required to implement each of these circuits.
5. (10 points) Use a Karnaugh map to find a simplest *product-of-sums* expression for \( F(X,Y,Z) = \Sigma m(0,1,4), d(X,Y,Z) = \Sigma m(2,6) \).

Use a Karnaugh map to find a simplest *sum-of-products* expression for \( F(A,B,C,D) = \Sigma m(1,3,4,5,9,10,12), d(A,B,C,D) = \Sigma m(8,11,14,15) \).

How many gates of each type are needed to implement this expression directly?
6. (15 points) The diagram at right shows a three digit, excess-3 adder. Complete the VHDL module shown below, so that it implements this circuit. Note that the digit type has been declared to be five bits, instead of four to make it easier for you to write the required VHDL. However, you will need to take this into account when you specify the logic.

```vhdl
package commonDefs is
  subtype digit is std_logic_vector(4 downto 0);
  type word is array(2 downto 0) of digit;
end package commonDefs;

library IEEE; ... use work.commonDefs.all;

entity xs3adder is port(
  A, B: in word;
  S: out word; carryOut: out std_logic);
end xs3adder;

architecture xs3arch of xs3adder is

begin

end xs3arch;
```
7. (10 points) Draw a schematic for a four bit version of the count-ones circuit from design problem 2, using four input LUTs. Each output of your circuit should have three bits. Use as few LUTs as you can and make the worst-case delay as small as possible.
8. (6 points) Define each of the following terms, with respect to a positive edge-triggered $D$ flip flop.

(a) propagation delay

(b) setup time

(c) hold time