1. (6 points) List all the minterms for the expression \((B' + A)C + AC' + BC\).

Expanding the expression gives

\[
B'C + AC + AC' + BC = A'B'C + AB'C + ABC + A'B'C' + ABC' + A'BC
\]

All the minterms appear at right. Numerically, they are 1, 3, 4, 5, 6 and 7.

List the maxterms for the expression.

The “missing minterms” are 0 and 2. These correspond to the two maxterms of the expression 
\((A + B + C)\) and \((A + B' + C)\)

2. (5 points) Show that the following Boolean equation is true using algebraic simplification
(not Karnaugh maps). Show your work.

\[
AB + BC' + CD = (B + C)(B + D)(A + C' + D)
\]

Expanding the right side gives

\[
(B + C)(B + D)(A + C' + D) = (B + BC + BD + CD)(A + C' + D)
\]

\[
= (B + CD)(A + C' + D)
\]

\[
= AB + BC' + BD + (ACD + CD)
\]

\[
= AB + BC' + BD + CD
\]

\[
= AB + BC' + (BC'D + BCD) + CD
\]

\[
= AB + (BC' + BC'D) + (BCD + CD)
\]

\[
= AB + BC' + CD
\]
3. (8 points) The attached simulation output is from an execution of the simple processor introduced in section 1 of the lecture notes. The instruction set for the processor appears below. Answer the following questions using the simulation output.

What does the instruction stored at location 000C do? This instruction is A011. It adds the content of memory location 011 to the accumulator.

What is the value of memory location 0011 at time 6 µs? 0021

What is the value of the program counter at time 4450 ns? 000f

What is the value of memory location 0010 at time 7 µs? 0011

0000  halt execution
0001  negate the value in the ACC
1xxx  change the value of the ACC to xxx
2xxx  load the contents of memory location xxx into the ACC
3xxx  load the ACC from the memory location whose address is stored in memory location xxx
4xxx  store the value in the ACC in memory location xxx
5xxx  store the value in the ACC in the memory location whose address is stored in memory location xxx
6xxx  change the value of the PC to xxx
7xxx  change the value of the PC to xxx if ACC = 0
8xxx  change the value of the PC to xxx if ACC > 0
9xxx  change the value of the PC to xxx if ACC < 0
axxx  add the value in memory location xxx to the ACC
4. (10 points) Draw a logic circuit using the smallest possible number of simple gates (AND, OR and inverters, only) for the logic expression \( UX' + X(V + Z') + (V' + U)XZ' \).

![Truth Table]

How many transistors are required by a CMOS version of this circuit? Show how to improve it by using NAND and NOR gates. How many transistors does this version require?

The circuit above requires 36 transistors if implemented directly in CMOS. The circuit shown below uses just 24 transistors.

![Circuit Diagram]

5. (8 points) Use a Karnaugh map to find the simplest \textit{sum-of-products} expression for 
\( F(X,Y,Z) = \Sigma m(1,2,4), d(X,Y,Z) = \Sigma m(3,6) \)

Use a Karnaugh map to find the simplest \textit{product-of-sums} expression for 
\( F(A,B,C,D) = \Sigma m(1,2,6,7,8,9,15), d(A,B,C,D) = \Sigma m(3,4,5,11) \)

![Karnaugh Map 1]

\( F' = \overline{A' \overline{C} \overline{D}'} + \overline{B} \overline{C} + A \overline{C} \overline{D} \)

\( F = (A + C + D)(B' + C)(A' + C' + D) \)
6. (5 points) Show how to implement the function $F(A, B, C, D) = \Sigma m(0, 2, 5, 7, 9, 11)$, $d(A, B, C, D) = \Sigma m(3, 4, 6, 15)$ using an 8 input multiplexor.

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

![Multiplexor Diagram](image-url)
7. (10 points) The circuit below shows a combinational circuit that implements a 5 bit version of the parallel pulse-parity function from design problem 2. What is the worst-case propagation delay for a 64 bit version of this circuit, using only simple gates with a delay of 1 ns? (The worst-case propagation delay is the maximum time from when an input changes until all outputs reach their final value.)

The worst-case delay is $63+4=67$ ns.

The circuit outlined below is a lookahead version of the same circuit. The boxes represent a repeated sub-circuit. In the rightmost box, fill in this circuit. What is the worst-case delay for a 64 bit version of this circuit, implemented using simple gates?

The 64 bit version would have a delay of $6+3+6=15$ ns.
8. (15 points) The circuit shown below implements a ternary (base 3) half-adder. The pair of input bits \((A_i, B_i)\) represents a single ternary digit (the bit pair 00 represents the ternary digit 0, the bit pair 01 represents the ternary digit 1 and the bit pair 10 represents the ternary digit 2). Similarly for the outputs \((X_i, Y_i)\). We can build a ternary increment circuit by combining these ternary half-adder circuits together, in the same way as with a binary ripple-carry increment circuit. Suppose that the input presented to a ternary increment circuit with four ternary digits corresponds to the ternary value 1022. What are the values of the nine output bits? Fill in your answer below the output signals listed below.

\[ \text{Cout, } (X_3, Y_3), (X_2, Y_2), (X_1, Y_1), (X_0, Y_0) \]

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The partial VHDL module below implements a ternary increment circuit with 8 ternary digits. Complete the missing parts. Your VHDL should be complete and syntactically correct.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ternaryInc is
  Port (
    A, B : in std_logic_vector(7 downto 0);
    Cin : in std_logic;
    X, Y : out std_logic_vector(7 downto 0);
    Cout : out std_logic
  );
end ternaryInc;

architecture arch1 of ternaryInc is
  signal C: std_logic_vector(7 downto 0);
  begin
    process(A, B, Cin, C) begin
      X(0) <= (A(0) and (not Cin)) or (B(0) and Cin);
      Y(0) <= (B(0) and (not Cin)) or
              ((not A(0)) and (not B(0)) and Cin);
      C(0) <= A(0) and Cin;
      for i in 1 to 7 loop
        X(i) <= (A(i) and (not C(i-1))) or (B(i) and C(i-1));
        Y(i) <= (B(i) and (not C(i-1))) or
                ((not A(i)) and (not B(i)) and C(i-1));
        C(i) <= A(i) and C(i-1);
      end loop;
      Cout <= C(7);
    end process;
  end arch1;
```
9. (10 points) The figure shown below is a state diagram for a sequential circuit with one input A and two outputs X and Y. Is this a Moore model circuit or a Mealy model circuit? Fill in the values in the next state table.

It’s a Moore model circuit.

<table>
<thead>
<tr>
<th>$S_1S_0A$</th>
<th>XY</th>
<th>$D_1D_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>xx</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>xx</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>01</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>01</td>
</tr>
</tbody>
</table>

What are the next state equations for the circuit?

$D_1 = S_0 + S_1'A + S_1A'$

$D_0 = S_1'A + S_0A$