1. (4 points) Assuming 8 bit words, what is the binary representation of 79? 0100 1111

What is the hexadecimal representation of 79? 4f

What is the binary representation of –79 in 2s complement? 1011 0001

What is the binary representation of –79 in sign-magnitude form? 1100 1111

2. (4 points) How many simple gates (including inverters) are needed to directly implement the logic expression \( ACD + B'(C' + B'C) + (A' + C)(B + D) \)?

5 and gates, 5 or gates and 3 inverters, so 13 gates.

If the circuit is implemented with CMOS, how many transistors does it use?

\( 10 \times 6 + 3 \times 2 = 66 \).

3. (8 points) Write a Boolean expression for each output of the circuit shown below and give a truth table for it.

\[ Y = (A \oplus C)B + (C + D)' \]
\[ Z = B(C + D) \]

<table>
<thead>
<tr>
<th>( AB )</th>
<th>( CD )</th>
<th>( YZ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
<td>10</td>
</tr>
<tr>
<td>0001</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>0010</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>0011</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>0100</td>
<td>0000</td>
<td>10</td>
</tr>
<tr>
<td>0101</td>
<td>0000</td>
<td>01</td>
</tr>
<tr>
<td>0110</td>
<td>0000</td>
<td>11</td>
</tr>
<tr>
<td>0111</td>
<td>0000</td>
<td>11</td>
</tr>
<tr>
<td>1000</td>
<td>0000</td>
<td>10</td>
</tr>
<tr>
<td>1001</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>1010</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>1011</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>1100</td>
<td>0000</td>
<td>10</td>
</tr>
<tr>
<td>1101</td>
<td>0000</td>
<td>11</td>
</tr>
<tr>
<td>1110</td>
<td>0000</td>
<td>01</td>
</tr>
<tr>
<td>1111</td>
<td>0000</td>
<td>01</td>
</tr>
</tbody>
</table>
4. (12 points) Simplify the following two expressions using Karnaugh maps. Try to find expressions that share as many common terms as possible.

\[ X = ((AC)' + AB')D' + AB'CD \]
\[ Y = AC + (A+B+C)' + A'B + (C+D)' \]

Can these expressions both be implemented in a PLA with 4 inputs, 2 outputs and 5 product terms? If so, explain how. If not, explain why not.

\[ X = ((AC)' + AB')D' + AB'CD = A'D' + C'D' + AB'D' + AB'CD \]
\[ Y = AC + (A+B+C)' + A'B + (C+D)' = AC + A'B'C' + A'B + C'D' \]

These expressions can be implemented in a PLA with 5 product terms, since they can be expressed using just 5 unique product terms, as shown above.
5. (10 points) The figure below shows an up-counter with carry look-ahead. Explain what would need to be changed to convert this to a similar down-counter with carry look-ahead. Note that the equations for the decrement logic can be written as follows:

\[ D_i = Q_i \oplus \text{Borrow}_{in_i} = (Q_i \oplus \text{Borrow}_{in_i})' \]

\[ \text{Borrow}_{in_i+1} = \text{Borrow}_{out_i} = Q_i'B_{in_i} \]

So, \[ \text{Borrow}_{out_i}' = Q_i + \text{Borrow}_{in_i}' = Q_i + Q_{i-1} + B_{in_{i-1}}' \]

Change all the and gates to or gates.

Change all the xor gates to xnor gates.

Add an inverter at the enable input.
6. (10 points) Draw a state table for the sequential circuit shown below.

![Circuit Diagram]

\[ D_A = X \oplus B \quad Y = AB' \]
\[ D_A = X + (A \oplus B)' \quad Z = (AB)' \]

<table>
<thead>
<tr>
<th>AB</th>
<th>X</th>
<th>D_A</th>
<th>D_B</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>01</td>
<td>01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>11</td>
<td>01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>10</td>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>01</td>
<td>01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>00</td>
<td>11</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>11</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>11</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>01</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

7. (10 points) Answer the following questions, in connection with the circuit in problem 7. Assume that the flip flop setup time is 2 ns, hold time is 1 ns, maximum clock skew is 1 ns, flip flop propagation delay is between 1 and 3 ns and gate delay is between .8 and 2.5 ns.

(a) Is this circuit subject to hold time violations? If so, explain how to eliminate them.

It is. From the output of flip flop B to the input of flip flop A, there is just one gate delay, and so the propagation delay from the clock change to the input change can be as little as 1+.8 , which is less than the sum of the clock skew and the hold time. This violation can be eliminated by adding a pair of inverters on the path from B back to the input of the xor gate.

(b) Ignoring the inputs, what is the shortest clock period for which the circuit is not subject to setup time violations? Take into account the modifications made in part (a) to avoid hold time violations (if any).

With the above changes, both flip flop output to input paths have a maximum propagation delay of 7.5 ns. Adding in the flip flop propagation delay of 3 ns, the clock skew of 1 ns and the setup time of 2 ns, gives 13.5 ns, as the minimum safe clock period.

(c) If the clock goes high at time 0, during what time period must input X be stable to guarantee that there are no setup or hold time violations?

There is one gate delay from the input to both of the flip flops, so the input must be stable from 2+2.5 = 4.5ns before the clock changes until 1+.8 = .2 ns after the clock changes.

(d) If the clock goes high at time 0, during what time period can output Y be changing? Output Z? Y can be changing from 1.8 to 8 ns after the clock changes. Z can be changing from 1.8 to 5.5 ns after the clock changes.
8. (15 points) Draw a schematic for a sequential circuit corresponding to the state diagram shown below.

<table>
<thead>
<tr>
<th>AB X</th>
<th>D_A</th>
<th>D_B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 0</td>
<td>11 1</td>
<td>1 1</td>
<td></td>
</tr>
<tr>
<td>00 1</td>
<td>10 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>01 0</td>
<td>00 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>01 1</td>
<td>10 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10 0</td>
<td>11 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10 1</td>
<td>10 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11 0</td>
<td>00 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>11 1</td>
<td>01 1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[ D_A = B' + A'X \]
\[ D_B = B'X' + ABX \]
\[ Y = (A \oplus B)' \]
9. (6 points) Draw a state diagram corresponding to the VHDL program shown below.

```
entity foo is
  port (b, clk: in STD_LOGIC; u, v: out STD_LOGIC);
end foo;
architecture bar of foo is
  type state_type is (baseball, puck, birdie);
  signal state: state_type;
begin
  process(clk) begin
    if clk'event and clk = '1' then
      if state = baseball and b = '1' then
        state <= birdie;
      elsif state = puck and b = '0' then
        state <= baseball;
      elsif state = puck and b = '1' then
        state <= birdie;
      elsif state = birdie and b = '0' then
        state <= baseball;
      elsif state = birdie and b = '1' then
        state <= puck;
      end if;
    end if;
  end process;
  u <= '1' when state = baseball or state = puck else '0';
  v <= '1' when state = birdie else '0';
end bar;
```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity funky is
  port (
    clk: in STD_LOGIC;
    mode: in STD_LOGIC_VECTOR(1 downto 0);
    d:      in STD_LOGIC_VECTOR(5 downto 0);
    q:      out STD_LOGIC_VECTOR(5 downto 0)
  );
end funky;

architecture bad of funky is
  signal reg: STD_LOGIC_VECTOR(5 downto 0);
begin
  process(clk) begin
    if clk'event and clk = '1' then
      if mode = "00" then
        reg <= d;
      elsif mode = "01" then
        if reg(5 downto 2) = "000000" or
        reg(5 downto 2) = "000001" or
        reg(5 downto 2) = "000010" then
          reg <= not reg;
        else
          reg <= reg - "000011";
        end if;
      elsif mode = "11" then
        reg <= reg(0) & reg(5 downto 1);
      end if;
    end if;
  end process;
  q <= reg;
end bad;
11. (15 points) The simulation output below is from the simple processor running a program. The vertical lines mark the boundaries between instructions. Fill in all the missing waveforms. The processor instruction set appears at the bottom of the page.
12. (4 points) Consider a 256 Kbit SRAM with a word size of 16. How many address bits does it take to address all the words in the memory?

$2^{18}$ bits and $2^{18}/2^4=2^{14}$ words, so 14 address bits.

Assuming a square memory array, how many address bits are used by the row decoder?

9

13. (4 points) Draw a schematic for a positive edge-triggered D flip flop using just NAND gates and inverters.