Nine problems, 100 points.

Closed books, closed notes, no calculators. You would be wise to read all problems before beginning, note point values and difficulty of problems, and budget your time accordingly.

Please do not open the test until I tell you to do so.

Good luck!
1. (9 total points)
   a) (2 points) convert to hex and binary:

   \[ 37024 \text{ (octal)} = 01111100001000 \text{ (binary)} = 3E14 \text{ (hex)} \]

   b) (5 points) convert to binary, octal and decimal:

   \[ A55.6 \text{ (hex)} = 101001010101.0110 \text{ (binary)} \]

   \[ = 5125.3 \text{ (octal)} \]

   \[ = 2645.375 \text{ (decimal)} \]

   c) (2 points) convert to octal and hex

   \[ 4553712 \text{ (binary)} = 4553712 \text{ (octal)} = 928_{E5} \text{ (hex)} \]
2. (8 points) Your input is an XS3 signal and your output is:

\[
F = \begin{cases} 
1 & \text{if the input is a valid currency in US dollars, i.e. $1, $2, $5} \\
0 & \text{otherwise}
\end{cases}
\]

Find a minimal SOP form. Is this unique? Find a minimal POS form. Is this unique? (The first K-map is for the SOP form. The second is for the POS form. Clearly, the entries are the same, but it is much less messy to do the two forms on separate K-maps.)

\[
\text{SOP: } F = \overline{B} \overline{C} \overline{D} + \overline{A} \overline{E} \text{ unique}
\]

\[
\text{POS: } F = (\overline{A} + B)(B + \overline{D})\overline{C}
\]

\[
= (\overline{A} + B)(\overline{A} + \overline{D})\overline{C}
\]

\text{Not unique}
3. (11 total points)
(1 point) Convert 94 and -94 to 8 bit 2’s complement representation.

\[ 94 = 01011110 \quad -94 = 10100010 \]

(1 point) Convert 78 and -78 to 8 bit 2’s complement representation.

\[ 78 = 01001110 \quad -78 = 10110010 \]

(1 point) Convert 29 and -29 to 8 bit 2’s complement representation.

\[ 29 = 00011101 \quad -29 = 11100011 \]

(2 points) Perform the addition 78 + 29 in 8 bit 2’s complement representation.

\[
\begin{array}{c}
01001110 \\
00011101 \\
\hline
01101011 = 107
\end{array}
\]

(2 points) Perform the subtraction 78 - 29 in 8 bit 2’s complement representation.

\[
\begin{array}{c}
01001110 \\
11100011 \\
\hline
100110001 = 49
\end{array}
\]

discard

(2 points) Perform the subtraction 94 - 78 in 8 bit 2’s complement representation.

\[
\begin{array}{c}
01011110 \\
10110010 \\
\hline
100010000 = 16
\end{array}
\]

(2 points) Perform the subtraction 78 - 94 in 8 bit 2’s complement representation.

\[
\begin{array}{c}
01001110 \\
10100010 \\
\hline
11110000 = -16 \\
\text{2's comp} \\
00010000 = 16
\end{array}
\]
4. (7 points) Your input is in natural BCD. \( F = 1 \) if \( ABCD \) encodes a digit that is part of today's date (05/10/2000), and \( F = 0 \) otherwise. Find a minimal NAND-NAND logic for \( F \). Is this solution unique?

<table>
<thead>
<tr>
<th>Minterm</th>
<th>ABCD</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>X</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>X</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>X</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>X</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>X</td>
</tr>
</tbody>
</table>

\[
F = \overline{A} \overline{B} \overline{C} \overline{D} \overline{A} \overline{B} \overline{D} \overline{A} \overline{B} \overline{D} \\
= \overline{A} \overline{B} \overline{C} \overline{D} \overline{B} \overline{C} \overline{D} \text{ Not unique.}
\]
5. (15 points) The 8-bit shift register shown on the next page is able to realize the specified functions from the function table provided. Show how to connect the inputs and any necessary external logic to make it a bidirectional ring counter with an absorbing state. The behavior of the ring counter is completely specified in the Moore diagram provided. Note the two-variable input code that is needed to make the system function will require some (very simple) external logic to function.

### Function Table

<table>
<thead>
<tr>
<th>Decimal ABC</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 000</td>
<td>Shift right</td>
</tr>
<tr>
<td>1 001</td>
<td>Hold</td>
</tr>
<tr>
<td>2 010</td>
<td>Shift Left</td>
</tr>
<tr>
<td>3 011</td>
<td>Shift right twice</td>
</tr>
<tr>
<td>4 100</td>
<td>Shift left twice</td>
</tr>
<tr>
<td>5 101</td>
<td>Load parallel inputs</td>
</tr>
<tr>
<td>6 110</td>
<td>Parallel clear (reset)</td>
</tr>
<tr>
<td>7 111</td>
<td>Parallel set</td>
</tr>
</tbody>
</table>

### Inputs for ring counter

<table>
<thead>
<tr>
<th>XY</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Shift right</td>
</tr>
<tr>
<td>01</td>
<td>Shift left</td>
</tr>
<tr>
<td>10</td>
<td>Start</td>
</tr>
<tr>
<td>11</td>
<td>Clear</td>
</tr>
</tbody>
</table>

START (From any state) → CLEAR (From any state)
Show only necessary connections— not all 8.

C = x \bar{y}
6. (19 points) Implement the Moore diagram below in SOP logic and D flip-flops. The state variables are ABC where A is the most significant bit. The input variable is X. Start your design assuming the unused states are don’t cares. Then come back and ensure that these states are transient (i.e. you can’t get stuck in them.) (Hint: If you ignore the last sentence you will still get most of the problem correct.)

\[ D_A = \overline{A} + \overline{B} + \overline{C} \]

\[ D_B = AB + AC \]

\[ D_C = AB + \overline{C} + X \]
7. (12 points) You have an 8-bit word, $X_7 \ldots X_0$, and you need to generate a ninth bit such that the total 9 bits will have even parity. (Equivalently, you need to check the 8 bits for odd parity.) Do this with just three two-input XOR gates and the 74AS181 ALU, for which the data sheet is on the next two pages. Draw the resulting circuit. (Hint: You will need to break the 4-bit output S line into its four one-bit outputs.)

\[
A = X_7 - X_4 \\
B = X_3 - X_0 \\
(\text{or vice versa})
\]
Figure 4–11
A 4-bit universal arithmetic logic unit (74AS181) (Part b reprinted by permission of Texas Instruments)
### Advanced Design Techniques with Combinational Circuits

#### Table 4-11 Continued

**Selection**

<table>
<thead>
<tr>
<th>S3</th>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>M = H Logic Functions</th>
<th>M = L: Arithmetic Operations</th>
<th>( C_n = L ) (no carry)</th>
<th>( C_n = H ) (with carry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>( F = \overline{A} )</td>
<td>( F = \overline{A} )</td>
<td>( F = \overline{A} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>( F = 1 )</td>
<td>( F = 1 )</td>
<td>( F = 1 )</td>
<td>( F = 1 )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
</tr>
</tbody>
</table>

**Selection**

<table>
<thead>
<tr>
<th>S3</th>
<th>S2</th>
<th>S1</th>
<th>S0</th>
<th>M = H Logic Functions</th>
<th>M = L: Arithmetic Operations</th>
<th>( C_n = H ) (no carry)</th>
<th>( C_n = L ) (with carry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>( F = \overline{A} )</td>
<td>( F = \overline{A} )</td>
<td>( F = \overline{A} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>( F = 1 )</td>
<td>( F = 1 )</td>
<td>( F = 1 )</td>
<td>( F = 1 )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = A )</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
<td>( F = \overline{A} \cdot \overline{B} )</td>
</tr>
</tbody>
</table>

Figure 4-11 Continued
8. (10 points) Put the following into SOP form:

\[
F = \overline{AB CD} \cdot \overline{EF + GH} = \overline{AB} \cdot \overline{CD} + (EF + GH)
\]

\[
= (A + B)(\overline{CG} + D) + (EF + GH)
\]

\[\therefore F = \overline{A} \overline{C} + \overline{A} \overline{D} + \overline{B} \overline{C} + \overline{B} \overline{D} + EF + GH\]

\[
G = A \oplus B \oplus C \oplus D
\]

\[
G = \overline{ABCD} + \overline{ABC}D + \overline{AB}CD + \overline{A}BCD
\]

\[
+ \overline{ABCD} + \overline{A}B\overline{CD} + AB\overline{CD} + AB\overline{D}
\]
9. (9 points) Assume the system below always starts in state $A = 0, B = 0$. Analyze the system’s behavior, produce the Moore diagram, and state in words what the system is doing.

![Diagram of a JK flip-flop with inputs $J_A$, $K_A$, $J_B$, $K_B$, and clock $CLK$ connected to outputs $A$, $B$, $C$, and $D$.]

- $J_B = K_B = 1 \Rightarrow$ Always toggles
- $J_A = K_A = B \Rightarrow$ Hold if $B = 0$
  Toggle if $B = 1$

Counts by 3's to 12 & starts over.
The "real" state variables are just the first 2 bits.