1) No calculators, hand-held or laptop computers, cellular phones or pagers allowed. 
2) If a box is provided, the contents of the box will be graded as the final answer. 
3) Show your work for consideration of partial credit. 
4) If you write anything on the back of a page and want it considered in the grading you must write “See back of page (give page number)” on the problem page. 
5) Two removable sheets with Boolean identities and theorems on one and Verilog operators on the other are given at the end of the exam.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. (14) A very simple circuit containing flip-flops and latches is given below. You are to sketch carefully the output signals that result from the applied input signals. Assume that the propagation times and setup times are much shorter than the intervals between changes in the inputs.

Corrected answer in red; wrong answer in dashes.
2. (a) (6) Find the flip-flop input equations in sum-of-products (SOP) form for the given circuit.

\[ J_A = B + X + Y \]

\[ K_A = X \bar{Y} \]

\[ D_B = \bar{X} A B + X \bar{A} + X B \]

(b) (5) Find sum of product next state equations \( A(t + 1) \) and \( B(t + 1) \) for the given circuit.

\[ A(t + 1) = J_A \bar{A} + K_A A = (B + X + Y) \bar{A} + (\bar{X} + Y) A \]

\[ A(t + 1) = \bar{A} B + X \bar{A} + \bar{X} A + Y \]

\[ B(t + 1) = \bar{X} A B + X \bar{A} + X B \]
3. (a) (9) Fill in the state table given for the equations given for $X(t+1)$, $Y(t+1)$, $W$ and $Z$.

$$X(t+1) = A X + \overline{B} Y$$
$$Y(t+1) = A B X + X Y \overline{B} + \overline{A} B Y$$
$$W = X \overline{Y} + \overline{X} Y$$
$$Z = \overline{X} Y$$

Note the 00, 01, 11, 10 order used in the table for $X$, $Y$ and $A$, $B$ to permit you to use K-map methods for entering values.

<table>
<thead>
<tr>
<th>$A B$</th>
<th>$A B$</th>
<th>$A B$</th>
<th>$A B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>1 1</td>
<td>1 0</td>
</tr>
<tr>
<td>$X(t)Y(t)$</td>
<td>$X(t+1)Y(t+1)$</td>
<td>$X(t+1)Y(t+1)$</td>
<td>$X(t+1)Y(t+1)$</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 0</td>
</tr>
<tr>
<td>1 1</td>
<td>1 1</td>
<td>0 1</td>
<td>1 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>0 0</td>
<td>1 1</td>
</tr>
</tbody>
</table>

(b) (9) Find the state diagram for the state table given below.

<table>
<thead>
<tr>
<th>$Y_1 Y_2$</th>
<th>$X_1 X_2$</th>
<th>$X_1 X_2$</th>
<th>$X_1 X_2$</th>
<th>$X_1 X_2$</th>
<th>$X_1 X_2$</th>
<th>$X_1 X_2$</th>
<th>$X_1 X_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>1 0</td>
<td>0 0</td>
<td>1 0</td>
<td>0 0</td>
<td>1 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>1 1</td>
<td>0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>1 1</td>
<td>0 0</td>
</tr>
</tbody>
</table>

![State Diagram](image.png)
4. (a) (7) For the state diagram given below, find the sequence of states and outputs that results from the applied input sequence.

(b) (3) A state-of-the art microprocessor has a clock frequency of 2GHz (2 x 10^9 cycles/second).

1) What is the clock period for the microprocessor in nano (10^-9) seconds? 0.5 ns
2) Timing parameters are given for the flip-flops in the Table 1. What is the maximum time delay that can be tolerated for a combinational logic path from flip-flop to flip-flop in the microprocessor? 0.35 ns

**Table 1: Flip-flop timing parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Max. Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagation Delay $t_p$</td>
<td>0.1 ns</td>
</tr>
<tr>
<td>Setup Time $t_{su}$</td>
<td>0.05 ns</td>
</tr>
<tr>
<td>Hold Time $t_h$</td>
<td>0.01 ns</td>
</tr>
</tbody>
</table>

$CP = \text{time delay} + t_p + t_{su} \Rightarrow \text{time delay} = CP - t_p - t_{su} = 0.5 - 0.1 - 0.05 = 0.35 \text{ ns}$
5. (6) The **state diagram** for a sequence recognizer for the subsequence 00, 01, 11 is to be determined. **10 never occurs!** The sequence is to be recognized by producing a 1 output on Z when the 11 is applied to X1X2 regardless of where the subsequence occurs in an overall sequence. The “backbone” of this diagram given below. You are to complete the diagram by adding the five arcs labeled with input/output value pairs needed. **No credit will be given for a solution with states added.**

(b) (10) A **state diagram** is to be developed for a sequential code converter with input BCO and output G. The circuit is to convert a BCO (binary coded octal) code presented most significant bit (B2) first in input BCO to a Gray code that appears on output G most significant bit first (G2). The circuit is to be Mealy with the output bit value appearing when the corresponding input bit value is applied. When the conversion of a 3-bit BCO sequence to a 3-bit Gray code is completed, the circuit is to be back in the initial (reset) state ready to receive the next sequence. The sequence transformation is given below in a table. Find the state diagram. **Hint:** There are to be **no more than** 7 states in the diagram.
6. (a) (2) A state table has 14 rows and an engineer chooses to use 7 state variables to encode the states because it gives less complex combinational logic in the sequential circuit.

1) How many unused states will there be in the coded state table?

2) How many excess state variables has the engineer used beyond the minimum number required?

The following table is for the rest of the problem. Note the 00, 01, 11, 10 ordering for the values for \( X, Y \). This is to make it easy for you to construct K-maps using the information given.

<table>
<thead>
<tr>
<th>( X )</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( A(t) )</th>
<th>( A(t+1) )</th>
<th>( A(t+1) )</th>
<th>( A(t+1) )</th>
<th>( A(t+1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) (3) For the state table given above (outputs are irrelevant, so not given), find a minimum sum-of-products (SOP) flip-flop input equation for a T flip-flop for state variable \( A \).

\[
T_A = A \overline{X} \overline{Y} + \overline{A} Y + X Y
\]

(c) (4) For the state table given above, find a minimum sum-of-products (SOP) flip-flop input equations for a J-K flip-flop for state variable \( A \).

\[
J_B = Y \\
K_B = \overline{X} Y + X Y
\]
(d) (4) For the state table given above, find a minimum sum-of-products (SOP) flip-flop input equations for an **S-R flip-flop** for state variable **A**.

\[
\begin{array}{c|c|c|c}
S_A & X & 0 & 1 \\
A & 0 & X & 0 \\
& 1 & X & 0
\end{array}
\quad \begin{array}{c|c|c|c}
R_A & X & 0 & 0 \\
A & 1 & 0 & 1 \\
& 0 & 1 & 0
\end{array}
\]

\[
S_A = \overline{A} Y \\
R_A = \overline{X} Y + A X Y
\]

(d) (4) A new type of synchronous flip-flop has the following characteristic table. Find the corresponding excitation table with don’t cares used as much as possible for the new flip-flop type.

**Table 2: Characteristic Table for X-Y flip-flop**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Q(t)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Q(t)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q(t)</td>
</tr>
</tbody>
</table>

**Table 3: Expanded Characteristic Table for X-Y flip-flop**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Q(t)</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

From Table 3, \(Q(t) = Q(t+1) = 0\) results from \(X = 1, Y = 0\) or \(1\), so \(X = 1, Y = x\).
Q(t) = 0 and Q(t+1) = 1 result from X = 0 and Y = 0 or Y = 1, so X = 0, Y = x.
Q(t) = 1 and Q(t+1) = 0 result from X = 0 and Y = 1
Q(t) = 1 and Q(t+1) = 1 result from (X,Y) = 00, 10, and 11. This can be written as X 0 or 1 X.
1 X or 00, or X0 or 11. Full credit given for either one. Extra credit for both.

Table 4: Excitation Table for X-Y flip-flop

<table>
<thead>
<tr>
<th>Q(t)</th>
<th>Q(t+1)</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>

|          |        |     |
| X        | Y      |
| X        | 0      |
7. (a) (4) Draw the circuit diagram corresponding to the given Verilog description.

```verilog
module prob_7a (X, Y);
input [4:0] X;
output Y;
wire [2:0] N;

not N1 (Y, N[2]);
nand NA1 (N[2], N[1], X[2], N[0]);
nor NO1 (N[0], X[0], X[1]),
        NO2 (N[1], X[3], X[4]);
endmodule
```

Permutation of some connections is possible.

(b) (10) Write in the box beside each Verilog module, the name of the functional block represented by the module code. Select the names you write in from the following list (Some names may be used more than once):

- 4-bit Parallel Adder with CarryIn Variable
- 4-bit Parallel Adder with Fixed CarryIn
- 3-input Priority Encoder
- 3-input Non-Priority Encoder
- Quad 2-way Multiplexer
- Dual 4-way Multiplexer
- Quad 4-way Multiplexer
- 2-to-4 Decoder with Enable
- 2-to-4 Decoder without Enable
- None of the above

Sometimes some input vectors and some bits of the input vectors may not be used.

```verilog
module prob7b1(A, W, X, Y, Z, Q);
input A;
input [3:0] V, W, X, Y; Z
output [3:0] Q;
    assign Q = A ? X : Y;
endmodule
```

Quad 2-way Multiplexer
module prob7b2 (A, W, X, Y, Z, Q);
input A;
input [3:0] V, W, X, Y; Z
output [3:0] Q;
    assign Q = X[1] ? (X[0] ? V : W) : (X[0] ? Y: Z);
endmodule

module prob7b3 (A, W, X, Y, Z, Q);
input A;
input [3:0] V, W, X, Y; Z
output [3:0] Q;
assign Q[0] = A & ~X[1] & ~X[0];
assign Q[1] = A & ~X[1] & X[0];
endmodule

module prob7b4 (A, W, X, Y, Z, Q);
input A;
input [3:0] V, W, X, Y; Z
output [3:0] Q;
wire [3:0] U;
    assign Q = V ^ W ^ {U[2:0], 1'b0};
    assign U = ({U[2:0], 1'b0} & (V | W)) | (V & W);
endmodule

module prob7b5 (A, W, X, Y, Z, Q);
input A;
input [3:0] V, W, X, Y; Z
output [3:0] Q;
always@ (X)
endmodule

Quad 4-way Multiplexer
2-to-4 Decoder with Enable
4-bit Parallel Adder with Fixed Carry In
3-input Priority Encoder
Boolean Algebra Identities and Theorems

1. \( X + 0 = X \) 
2. \( X \cdot 1 = X \) 
   Existence of 1 and 0
3. \( X + 1 = 1 \) 
4. \( X \cdot 0 = 0 \) 
   Existence of 1 and 0
5. \( X + X = X \) 
6. \( X \cdot X = X \) 
   Idempotence
7. \( X + \bar{X} = 1 \) 
8. \( X \cdot \bar{X} = 0 \) 
   Existence of Complement
9. \( \bar{X} = X \) 
   Involution

----------------------------------------------------------------------------------------------------------------

10. \( X + Y = Y + X \) 
11. \( X \cdot Y = Y \cdot X \) 
   Commutative Laws
12. \( X + (Y + Z) = (X + Y) + Z \) 
13. \( X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z \) 
   Associative Laws
14. \( X \cdot (Y + Z) = X \cdot Y + X \cdot Z \) 
15. \( X + Y \cdot Z = (X + Y) \cdot (X + Z) \) 
   Distributive Laws
16. \( X + \bar{Y} = \bar{X} \cdot \bar{Y} \) 
17. \( X \cdot \bar{Y} = \bar{X} + \bar{Y} \) 
   DeMorgan’s Laws

----------------------------------------------------------------------------------------------------------------

18. \( X + X \cdot Y = X \) 
19. \( X \cdot (X + Y) = X \) 
   Absorption
20. \( X \cdot Y + X \cdot Y = Y \) 
21. \( (X + Y) \cdot (X + Y) = Y \) 
   Minimization
22. \( X + \bar{X} \cdot Y = X + Y \) 
23. \( X \cdot (X + Y) = X \cdot Y \) 
   Simplification
24. \( X \cdot Y + \bar{X} \cdot Z + Y \cdot Z = X \cdot Y + \bar{X} \cdot Z \) 
25. \( (X + Y) \cdot (\bar{X} + Z) \cdot (Y + Z) = (X + Y)(\bar{X} + Z) \) 
   Consensus

----------------------------------------------------------------------------------------------------------------

Selected Verilog Operators

Bitwise Logical Operators:

\(~\) not
\(&\) and
\(|\) or
\(^\wedge\) xor
\(~^\wedge\) or \(~^\wedge\) xnor

Logical and Relational Operators:

\(!\) not
\(&&\) and
\(||\) or
\(=\) equal
\(!=\) not equal
\(>=\) greater than or equal to
\(<=\) less than or equal to
\(>\) greater than
\(<\) less than

Conditional:

\(A \ ? B : C\) If A is true, then B, else C