Do not start the exam until you are told.

Turn off any cell phones or pagers.

Write your name and PID at the top of every page. Do not separate the pages.

This is a closed-book, closed-notes, no-calculator exam. You may only refer to one page of your own notes.

Do not look at anyone else’s exam. Do not talk to anyone but an exam proctor during the exam.

If you have a question, raise your hand and an exam proctor will come to you.

You have 80 minutes to finish the exam. When the time is finished, you must stop writing.

Write your answers in the space provided.

To get the most partial credit, clearly and neatly show all steps of your work.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 20 points</td>
<td></td>
</tr>
<tr>
<td>2. 15 points</td>
<td></td>
</tr>
<tr>
<td>3. 15 points</td>
<td></td>
</tr>
<tr>
<td>4. 15 points</td>
<td></td>
</tr>
<tr>
<td>5. 20 points</td>
<td></td>
</tr>
<tr>
<td>6. 15 points</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong> (100pts.)</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1
(20 points)

Multiple choices are worth 2 points each and True/False are worth 1 point each.

1. What is the output of the following circuit?

\[ \text{Answer: E} \]

2. Which of the following expressions look the same in SOP and POS?

\[ \text{i. } X' \]
\[ \text{ii. } X \times Y + X' \times Z + X \times Y' \times Z' \]
\[ \text{iii. } X \times Y \times Z' \]
\[ \text{iv. } X + Y + Z' \]

\[ \text{A. (i) and (iii) only} \]
\[ \text{B. (i) and (iv) only} \]
\[ \text{C. (iii) and (iv) only} \]
\[ \text{D. (ii), (iii) and (iv) only} \]
\[ \text{E. (i), (iii) and (iv) only} \]

\[ \text{Answer: E} \]

3. The circuit below compares three unsigned 4-bit binary numbers \( X, Y, \) and \( Z \) using the 4-bit magnitude comparators. Which of the following values of \( X, Y, \) and \( Z \) will result in \( S \) being 1?

\[ \text{Answer: C} \]
4. A new flip-flop, called \textit{AB} flip-flop, is created as shown below. What does the flip flop do?

\begin{center}
\includegraphics[width=0.5\textwidth]{flip-flop}
\end{center}

A. Set command \quad A = 0 \quad B = 0
B. Reset command \quad A = 0 \quad B = 1
C. Hold command \quad A = 1 \quad B = 0
D. Toggle command \quad A = 1 \quad B = 1

Answer : D

5. Given the Boolean function \( S(P,Q,R) \) as implemented below, what is the function \( S \)?
(P \rightarrow \text{Most significant bit}; R \rightarrow \text{Least significant bit})

\begin{center}
\includegraphics[width=0.3\textwidth]{demux}
\end{center}

A. \( Q + R' \)
B. \( P + Q + R' \)
C. \( Q' \cdot R \)
D. \( P + Q' + R \)
E. 0

Answer : C

6. Given the Boolean function \( S(P,Q,R) \) as implemented below, what is the function \( S \)?
(P \rightarrow \text{Most significant bit}; R \rightarrow \text{Least significant bit})

\begin{center}
\includegraphics[width=0.3\textwidth]{mux}
\end{center}

A. \( \Sigma \ m (0, 6, 7) \)
B. \( \Sigma \ m (1, 6, 7) \)
C. \( \Sigma \ m (1, 2, 3, 4, 5) \)
D. \( \Sigma \ m (6, 7) \)
E. \( \Sigma \ m (0) \)

Answer : A
7. State True or False:

<table>
<thead>
<tr>
<th>Part</th>
<th>Question</th>
<th>True / False</th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>A 16-bit hierarchical carry look ahead adder, constructed using 4 4-bit carry look ahead adders is faster than a 16-bit carry look ahead adder.</td>
<td>T</td>
</tr>
<tr>
<td>b)</td>
<td>DRAM is the fastest and densest type of volatile memory.</td>
<td>F</td>
</tr>
<tr>
<td>c)</td>
<td>A state machine with 30 states when partitioned into 5 partitions needs 6 idle states.</td>
<td>F</td>
</tr>
<tr>
<td>d)</td>
<td>D flip flop can be implemented from a T flip flop using the following logic: ( D = T'Q' + TQ' )</td>
<td>F</td>
</tr>
<tr>
<td>e)</td>
<td>Hold time constraints is needed to find the maximum CPU clock frequency.</td>
<td>F</td>
</tr>
<tr>
<td>f)</td>
<td>RAM of 2Kbytes capacity needs 4 RAM chips of 4096x1 bit capacity</td>
<td>T</td>
</tr>
<tr>
<td>g)</td>
<td>8-bit sequential multiplier uses lesser adders than the combinational one.</td>
<td>T</td>
</tr>
<tr>
<td>h)</td>
<td>EEPROM needs UV light to be erased.</td>
<td>F</td>
</tr>
</tbody>
</table>
Problem 2  
(15 points)

Given the state diagram in figure below, implement the state machine using a minimum number of connections on the PLA given below. Make sure to define all relevant signals/connections. Note that this PLA uses T-flip flops! Show all your work. The states are represented as $Q1Q0$; input as $X$ and output as $Z$.

The equations are:

\[ T0 = XQ1 + XQ0Q1' + X'Q0Q1' \]

\[ T1 = X'Q0Q1' + XQ0Q1 \]
Problem 3
(15 points)

The state diagram of a sequential circuit is given below:

a) Fill the circuit designed for the above state diagram using D flip flops with appropriate gates:
b) Complete the timing diagram for the state diagram given above. The propagation delay of both the flip flops is 2ns and the clock has the period of 8ns. Both flip flops provided are negative edge triggered.
Problem 4
(15 points)
Consider an ALU with two data inputs $A = A_n...A_1$ and $B = B_n...B_1$, two selection bits $S_1$ and $S_0$, one n-bit data output $F = F_n...F_1$, and one-bit carry out bit $C_{out}$ that implements the following four functions:

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>ALU Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$B - 1$ (Decrement $B$ by 1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-$B$ (Two’s complement of $B$)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$A$ XNOR $B$ (bitwise)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$A - B$</td>
</tr>
</tbody>
</table>

Implement one bit slice of this ALU, using exactly one full adders and with XOR, AND, OR and Inverter gates, without using multiplexers. Use minimum number of gates.
**Problem 5**  
(20 points)  
Consider the following *partially specified* state diagram

There are two pieces of known information about this FSM:  

- S3 is *equivalent* to S2  
- S0 is *NOT equivalent* to S5  

Please answer the following questions based on the given information.

1) Which of the following statements are correct? Circle the answer(s) that you think correct.
   - a. S0 is equivalent to S1  
   - b. S2 is equivalent to S4  
   - c. S1 is equivalent to S5  
   - d. S3 is NOT equivalent to S4

2) Please complete this FSM by specifying the values for all the “?” marks based on information provided thus far.
3) Use implication chart to reduce the states of the completely specified FSM you developed in Part 2)

4) Group the original states into maximal classes of compatibility. Use only as many groups as necessary.
Problem 6  
(15 points)

A circuit has a small register file which stores 6 numbers. This circuit receives an external input number and tries to find smallest register value range in the register file that contains the input number. After finding the value range, the circuit outputs the two register numbers that represent the **boundaries of the value range**.

For example, assume the following numbers are stored in the register file:

```
| 58 |
| 75 |
| 3  |
| 19 |
| 23 |
| 86 |
```

The following examples represent three typical cases that the circuit needs to handle:

1) The external input falls within the value range of the register file numbers but is not identical to any of them

   External input = 47

   ![Graph](image)

   Circuit output: **19, 58**

2) The external input is identical to one of the register file numbers

   External input = -23

   ![Graph](image)

   Circuit output: **-23, -23**

3) The external input falls outside the value range of the register file numbers

   External input = 110

   ![Graph](image)

   Circuit output: **86, 9999**  (Use 9999 to represent +infinity, and -9999 to represent -infinity)
a) Create a high-level state machine for this circuit

```
Addr = 0
upper = 9999
lower = -9999
ext_num = input

mem_num = Mem[addr]
Addr++

extend_num < mem_num < upper
Addr < 8

lower = mem_num
Addr < 8
lower < mem_num < extend_num
upper = mem_num

(mem_num > upper ||
mem_num < lower)
&& addr < 8

Addr = 8
```

b) Show details of the datapath, and label all signals. You can use register, comparator, counter and adder/subtractor in your datapath design. Draw the block diagram of how controller connects to the datapath.
This page is intentionally left blank. Use as scratch paper or to provide additional answer.