Do not start the exam until you are told.
Write your name and PID at the top of every page. Do not separate the pages.
Turn off and put away all your electronics.
This is a closed-book, closed-notes, no-calculator exam. You may only refer to one 8½ x 11” page of your handwritten 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.

1. 10 points
2. 20 points
3. 15 points
4. 15 points
5. 15 points
6. 25 points
Total (100 pts.)
Problem 1

True or false?

1-1. ___ Mealy FSM generally has more states compared to an equivalent Moore FSM.
1-2. ___ State minimization always leads to a smaller implementation as it reduces the number of flip-flops required.
1-3. ___ Master-slave FF design does not solve instability problems in latches.
1-4. ___ Carry-lookahead adders are generally faster than ripple-carry adders.
1-5. ___ DRAM is larger than SRAM because of the large capacitor.

Multiple choice

1-6. One of the these is not like the others:
   a) $W(a, b, c) = ab + bc + ca$
   b) $X(a, b, c)$ = majority function of $a, b, c$
   c) $Y(a, b, c) = ab + c (a \oplus b)$
   d) $Z(a, b, c) = \prod_{M(3, 5, 6, 7)}$

1-7. Which of these has the least number of transistors (NMOS and PMOS combined):
   a) 1-bit SRAM cell
   b) 4-input NAND gate
   c) Level-sensitive SR latch
   d) D flip-flop

1-8. Forbidden input for an NOR-based SR latch is:
   a) $S = 0, R = 0$
   b) $S = 0, R = 1$
   c) $S = 1, R = 0$
   d) $S = 1, R = 1$

1-9. A 20-state FSM requires minimum 5 FFs to represent its states. How many FFs would be required if the 20-state FSM were partitioned into three FSMs, each containing 5-state, 7-state, and 8-states from the original 20-state FSMs?
   a) 8
   b) 9
   c) 10
   d) 11

1-10. Which of the following Boolean algebra statements is incorrect?
   a) $XY + XY' = X$
   b) $(X + Y + Z)' = X' + Y' + Z'$
   c) $X + X'Y + XY + Y' = 1$
   d) $XY + YZ + X'Z = XY + X'Z$
Problem 2

Design and implement a circuit using gates and flip-flops that outputs a 1 if the last three sequence of 1 bit input is a either 011 or 110. For example:

Input: 0 → 1 → 1 → 0 → 1 → 0 → …
Sequence: 0 → 01 → 011 → 11 → 110 → 101 → 010 → …
Output: 0 → 0 → 1 → 0 → 1 → 0 → 0 → …

Upon reset, the sequence is set to null. Implement using minimum number of flip-flops and as few as possible 1 or 2-input gates. Flip-flops have synchronous set and reset inputs that may be used. (Hint: this design can be implemented with less than 5 gates in addition to FFs)
Problem 3

Consider the following circuit with D flip-flops and 1-bit full adders.

![Circuit Diagram]

a) Assuming:
- Each gate inside the 1-bit full adder has delay of $T_{gate} = 0.5\text{ns}$
- $T_{setup} = 0\text{ns}$
- $T_{hold} = 0\text{ns}$
- $T_{propdelay} = 0.1\text{ns}$

Can the circuit function properly with CLK at 500MHz? If not, what is the maximum clock frequency?
b) Assuming:
   - Each gate used inside the 1-bit fuller adder has delay of $T_{\text{gate}} = 0.1\text{ns}$
   - $T_{\text{setup}} = 0.2\text{ns}$
   - $T_{\text{hold}} = 0.2\text{ns}$
   - $T_{\text{propdelay}} = 0.5\text{ns}$

Can the circuit function properly with CLK at 500MHz? If not, what is the maximum clock frequency?
Problem 4

For the state transition diagram below, assign states using three guidelines heuristic. Based on this assignment, formulate the Boolean equations for the next states and the output as a function of the current state and the input.
Problem 5

Implement the design shown in OR-AND PLA below onto AND-OR PLA pictured on the next page. Use the minimum number of connections.
Problem 6

Design a circuit that computes the \( i \)th Fibonacci number \((i < 256)\). The mathematical expression and example of Fibonacci sequence is shown below.

\[
\begin{align*}
F_n &= 0 \quad \text{if } n = 0 \\
&= 1 \quad \text{if } n = 1 \\
&= F_{n-1} + F_{n-2} \quad \text{otherwise}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-number</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>21</td>
</tr>
</tbody>
</table>

To speed up the average computation time, use a 256 x 256-bit memory to store the computed Fibonacci number, so that the data can be retrieved without computation on a subsequent request to that Fibonacci number.

a) Design a high level state machine that takes in two inputs: START (1-bit) and INDEX (8-bit), and returns two outputs: DONE (1-bit) and NUMBER (256-bit).
b) Create datapath using minimum number of the following components: RAM, adder, comparator, and register. Specify size for each component.

c) Connect the datapath and the controller.
(This page is intentionally left blank. Use as scratch paper or to provide additional answers)