(1) (16pts) Consider the following logic functions expressed in standard sum-of-products form:

\[ f_1(A, B, C, D) = \sum m(2, 3, 5, 7, 11, 14, 15) + \sum d(1, 10) \]

\[ f_2(A, B, C, D) = \sum m(2, 3, 7, 9, 11, 15) \]

(a) Construct a Karnaugh Map for each function and write the Boolean expression for each function as a minimum sum-of-products realization (minimum number of (gates + gate inputs.). Show the product terms you have chosen to implement each function on the maps.

1(a) 6pts

\( f_1(A, B, C, D) = \) _______________________ \( f_2(A, B, C, D) = \) _______________________

(b) If the two functions are to be implemented together (i.e. as a single, two-output, four-input function), construct a Karnaugh Map for each output and show the product terms that represent the minimum number of (gates + gate inputs) for the combined, two-output function on the maps. Write the Boolean expression for each output.

1(b) 6pts

\( f_1(A, B, C, D) = \) _______________________ \( f_2(A, B, C, D) = \) _______________________
(c) Considering the first function \( f_1(A, B, C, D) = \sum m(2, 3, 5, 7, 9, 11, 14, 15) + \sum d(1, 10) \) only, draw a schematic diagram for \( f_1 \) that uses the minimum number of logic gates. Use only AND, OR, and inverter gates and assume input complements are not available (i.e. an inverter counts as a gate.)
(2) (16pts) Consider the logic function shown below in Karnaugh Map form. In all parts to this question, assume input complements are not available (i.e. an inverter counts as a gate.)

\[
f(A,B,C,D) = \begin{array}{cccc}
AB & 00 & 01 & 11 & 10 \\
00 & 1 & 0 & 1 & 0 \\
01 & 0 & 1 & 0 & 1 \\
11 & 1 & 1 & 1 & 0 \\
10 & 0 & 0 & 0 & 1 \\
\end{array}
\]

(a) Derive an implementation of this function using a **single 8-input, 3-control-line multiplexer** and a minimum number of logic gates.

**2(a) (4pts)**

(b) Derive an implementation of the function using a **single 4-input, 2-control-line multiplexer** and a minimum number of two-input XOR gates and inverters only.

**2(b) (4pts)**
(c) Implement the function using a minimum number of simple logic gates (AND, OR, NAND, NOR, XOR, XNOR, and inverters only.)

2(c) (4pts)

(d) Construct the **PLA-format table representation** for the function in **AND-OR form**. How many **product terms** would an AND-OR PLA implementation require?

2(d) (2pts)

| Product terms: __________ |

(e) If the function were implemented using a ROM, how many ROM **storage locations** would be required? **How many address inputs** would the ROM have?

2(e) (2pts)

| Storage Locations: _________ | Address inputs: _________ |

Additional space for Problem 2
(3) (19pts) Consider the following state-transition graph which is to be implemented as a clocked, synchronous sequential circuit. If State d is the reset state and the states are encoded as State a = 01, State b = 11, State c = 10, and State d = 00:

(a) Construct a **state transition table** for the machine in terms of **input X** and **present-state** \((y_1 \ y_2)\) for **output Z** and **next-state** \((Y_1 \ Y_2)\).

(b) Use **Karnaugh map(s)** to obtain the **reduced next-state and output equations** for the machine. **Show your Karnaugh map(s).**
(c) Use the reduced set of equations to obtain a circuit diagram for an implementation of the machine. Use the minimum number of logic gates (AND, OR, NAND, NOR, XOR, XNOR, or inverters) and positive-edge-triggered toggle (T) flip-flops only.

3(c) (6pts)

(c) Using your next-state table from Part 3(a) above, construct an implication table and check for state equivalence. List the equivalent states. Enter the word NONE below if you cannot find any equivalent states from your table.

3(d) (4pts)

Equivalent States: ____________________

Additional space for Problem 3
(4) (14pts) A clocked, synchronous sequential circuit is to be designed as follows:

“The circuit is to have a single input, X, that is used to control two outputs Z_1 and Z_2. While X is 1, (Z_1, Z_2) = (00). When X becomes 0, the outputs (Z_1, Z_2) begin the sequence (00, 01, 11, 10). If they reach the value 10 they hold that value until the input changes to 1 again.”

(a) Construct a state transition graph (STG) for the machine in Mealy form.

(b) Convert your STG to Moore form and make a state assignment that minimizes the amount of output logic needed for a Moore implementation of the machine.
(c) Determine the **next-state and output equations** for your machine and **implement the circuit using positive edge-triggered D flip-flops** and a **minimum number of two-input NAND gates only**. Assume X comes from the output of another flip-flop and so the complement of X is also available.

Additional space for Problem 4
(5) (18pts) (a) What is a fundamental-mode asynchronous circuit? Give a concise definition.

(b) Consider the state diagram for the asynchronous sequential machine shown below.

(c) With the code for State a given as all 0’s derive a race-free state assignment for the machine using a minimum number of internal state variables. Show all working.
(d) Provide a **schematic diagram** for the machine using a **minimum number of logic gates** (AND, OR, NAND, NOR, XOR, XNOR, or inverters). Indicate your state variables as labels on buffer symbols. Ensure that the **output logic cannot produce glitches** and state why that is so.
(6) (17pts)
(a) Draw the block diagram for a sequential, 32-bit full-adder that uses a single, one-bit full adder and two 32-bit serial shift registers to perform addition of two unsigned, 32-bit binary numbers. Explain how the adder operates and how overflow would be detected if it occurred.

6(a) (6pts)

(b) With the addition of a single "add/subtract" input and logic gates, modify your design to perform addition or subtraction of two unsigned binary numbers.

6(b) (4pts)
(c) Derive a block diagram for a bit-serial sequential circuit that can be used for comparing the relative magnitudes of two 32-bit numbers, A<31:0> and B<31:0>, stored in sign-magnitude form, where bit 31 is the sign bit. **Do not use subtraction.** Describe how your circuit operates.

**6(c) (7pts)**

Additional space for Problem 6