This is a take-home exam. You are allowed to check books, notes, and conduct an internet search. However, you are expected to work on the solution by yourself. No discussion with anyone else is allowed.

There are two parts to the exam. Part A weighs 90 points. Part B weighs 10 points. However, in Part B, the total weight of the problems is higher than 10 points. You are encouraged to earn up to 10 points. In other words, there is no penalty on wrong solutions in Part B, but the maximum points to earn are limited to 10 points.

Please read the following instructions carefully. If you are unclear about any of the questions on the exam, make the most plausible assumption to answer the question. Instructors and proctors will not answer questions on the exam material. All the Best!

Part A: (90 pts)

A.1. (Decoders and Multiplexers) Given a four-input switching function

\[ f(a, b, c, d) = \sum m(1, 2, 4, 8, 11, 13, 14) + \sum d(0, 15) \]

1.1. Implement the function using a minimal network of 4:16 decoders and OR gates.
1.2. Implement the function using a minimal network of 2:4 decoders and OR gates.
1.3. Implement the function using a minimal network of 8:1 multiplexers.
1.4. Implement the function using a minimal network of 2:1 multiplexers.

A.2. (Sequential Arithmetic Operations) Design a sequential operator that performs arithmetic operation \( A + 2 \times B \) on two \( n \)-bit binary numbers \((A, B)\) using D flip-flops and a minimal network of AND, OR, NOT gates. Assume that NOT gates are free. Show your design process and the logic diagram of the circuit.

A.3. (Counter) Use standard modulo counters to design various counters.
3.1 Given a modulo 16 counter, construct a 3 to 11 counter.
3.2 Given a modulo 8 counter, construct a counter that counts in a sequence 1, 5, 6, 7, 2, 3 with a minimal network of AND, OR, NOT gates. Assume that NOT gates are free. Show your design process and the logic diagram of the circuit.
3.3 Construct a modulo 128 counter using modulo 4 counters.

A.4. System Designs: Implement the following algorithm:

```
Alg(X,Y,Z,start,U,done)
Input X[7:0], Y[7:0], Z[7:0], start;
Output U[7:0], done;
Local-object A[7:0], B[7:0], C[7:0];
S1: If start’ goto S1 \parallel done \leftarrow 1;
S2: done \leftarrow 0 \parallel A \leftarrow X \parallel B \leftarrow Y \parallel C \leftarrow Z;
S3: B \leftarrow Add(A,B) \parallel C \leftarrow Add(B,C);
S4: If A[7] goto S6 \parallel A \leftarrow INC(A);
S5: If C[7] goto S4 \parallel C \leftarrow Add(B,C);
S6: If B[7] goto S5 \parallel B \leftarrow Add(A,B);
S7: U \leftarrow C \parallel goto S1;
End Alg
```

4.1. Design a data subsystem that is adequate to execute the algorithm by answering the following questions.
4.1.1. Write the table that lists the instructions and the corresponding components that should be used in the data path subsystem.
4.1.2. Draw the schematic diagram to show the data path subsystem. Label the inputs, outputs, and control signals of all components.
4.2. Design the control subsystem by answering the following questions.
4.2.1. Write the table that lists the value of control signals for every state.
4.2.2. Draw the state diagram.
4.2.3. Implement the control subsystem using a one-hot encoding design. Draw the logic diagram.

Part B: (10 pts)
In Part B, the total weight of the problems is higher than 10 points. You are encouraged to earn up to 10 points. In other words, there is no penalty on wrong solutions, but the maximum points to earn are limited to 10 points. Note that some problems are open, i.e. exact (perfect) solutions may be challenging to derive. Nonetheless, the principles taught in the class can be used to start the exploration.

B.1. Sequential Arithmetic Operation: Design a sequential arithmetic operator that performs arithmetic operation \( W + X - Y \) on three \( n \)-bit binary numbers, \( W, X, \) and \( Y \), in \( n \) clock cycles using D flip-flops and a minimal network of AND, OR, NOT gates. Assume that NOT gates are free. Show your design process and the logic diagram of the circuit. (5pts)

B.2. Multistate Networks: With reference to Benes and Omega networks shown in slide 21 of Lecture 16 (System Design I), we label the left side ports from top to bottom \((0, 1, 2, 3, 4, 5, 6, 7)\) and the right side ports from top to bottom \((5, 6, 7, 0, 1, 2, 3, 4)\). We connect the ports of the same labels between the right and left ports. (5pts)
1. Draw the routes using the Benes network (top figure in slide 21).
2. Draw the routes using the Omega network (bottom figure in slide 21).

B.3. System Designs: Description and conversion of handshake protocols. (5pts)
1. Define the 4-phase handshake protocol. Show that the algorithm stated in Problem A.4 follows the 4-phase handshake protocol.
2. Define the 2-phase handshake protocol.
3. Rewrite the algorithm stated in Problem A.4 using 2-phase handshake protocol.

B.4. System Designs: For the System Designs of Problem 5 in HW4, suppose that the resources of the arithmetic operators are limited: we have no more than one Add operator and no more than one Inc operator. Show your revised solution through subproblems 5.1, 5.2 and 5.3. (5 pts)