

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)

1. (Decoders) Given three four-input Boolean functions

$$f_1(a, b, c, d) = \sum m(2, 4, 6, 8) + \sum d(12, 13, 14, 15),$$

$$f_2(a, b, c, d) = \sum m(3, 7, 9) + \sum d(2, 4, 8, 14),$$

$$f_3(a, b, c, d) = \sum m(4, 6, 9, 15) + \sum d(3, 11, 12).$$

1.1. Implement the three functions using a minimal network of 4:16 decoders and OR gates.

1.2. Implement the three functions using a minimal network of 2:4 decoders and OR gates.

2. (Multiplexers) Assume a dual-railed system, where you have access to any variable and its complement. Implement the following four-input Boolean function as indicated in each of the following subproblems.

$$f(a, b, c, d) = \sum m(1, 3, 6, 11, 12) + \sum d(2, 4, 7, 9, 14).$$

2.1. Implement the function using a minimal network of 8:1 multiplexers.

2.2. Implement the function using a minimal network of 2:1 multiplexers.

3. (Sequential Arithmetic Operations) Design a sequential subtractor that performs arithmetic subtraction on two n -bit binary numbers, X and Y , i.e. $X - Y$, in n clock cycles using a D flip-flop and a minimal network of AND, OR, NOT gates. Show your design process and the logic diagram of the circuit.

4. (Counter) Use standard modulo counters to design various counters.

4.1 Given a modulo 16 counter, construct a 5 to 14 counter.

4.2 Given a modulo 8 counter, construct a counter that counts in a sequence 2,3,6,7,4,5, with a minimal network of AND, OR, NOT gates.

4.3 Construct a modulo 64 counter using modulo 16 counters.

5. 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 || done $\leftarrow$ 1;

S2: done $\leftarrow$ 0 || A $\leftarrow$ X || B $\leftarrow$ Y || C $\leftarrow$ Z;

S3: If A[7] goto S7 || A $\leftarrow$ Add(A,B);

S4: If B[7] goto S7 || B $\leftarrow$ Add(B,C);

S5: If C[7] goto S7 || C $\leftarrow$ Add(A,C);

S6: goto S3 || A $\leftarrow$ Add(A,B);

S7: U $\leftarrow$ C || goto S1;

End Alg

- 5.1. Design a data subsystem that is adequate to execute the algorithm by answering the following questions.
- 5.1.1. Write the table that lists the instructions and the corresponding components that should be used in the data path subsystem.
- 5.1.2. Draw the schematic diagram to show the data path subsystem. Label the inputs, outputs, and control signals of all components.
- 5.2. Design the control subsystem by answering the following questions.
- 5.2.1. Write the table that lists the value of control signals for every state.
- 5.2.2. Draw the state diagram.
- 5.2.3. 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: A sequential four-at-a-time adder inputs a_i, b_i, e_i, f_i the i 'th bit of four binary numbers in each clock cycle for $i = 0$ to $n - 1$ and produces the sum s_i . Draw the logic diagram of your design using a minimal number of D flip-flops and a minimal network of Full Adders. Note that you are allowed to use AND, OR, NOT gates only when it is necessary. (5 pts)

B.2. Multistate Networks: With reference to Benes and Omega networks shown in slide 21 of Lecture 16, arrange the routes so that every $output_i$ (on the right side of the figure) is connected to $input_j$ (on the left side of the figure), where $i + j = 7$ for all $0 \leq i, j \leq 7$. (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: Suppose that we are given a 16-bit multiply module, $Multiply_{16}(X, Y, Z, start, done)$, that multiplies two 16-bit numbers (for example a module as described in slide 5 of lecture 17). Design a system that multiplies two 32-bit numbers using the 16-bit Multiply module. (5 pts)

1. Write the program. Explain the interface with module $Multiply_{16}(X, Y, Z, start, done)$.
2. Describe the data subsystem with a schematic diagram. You can treat $Multiply_{16}(X, Y, Z, start, done)$ as a black box.
3. Draw the state diagram of the control subsystem and describe the control signals using a truth table.