1 Boolean Functions (10+10+10 = 30 points)

Boolean functions can be represented in several ways, such as a truth table, sum-of-products expression, product-of-sums expression, or a network of basic gates. This question deals with the worst-case (largest possible) sizes of common Boolean function representations (the worst case is across all possible functions with a given number of inputs).

Given a Boolean function with n inputs and 1 output, give expressions for the worst-case sizes of the following representations in terms of n. Also, give an example of a function that results in the worst-case size. Full credit will be given to sizes that are within a constant factor of the correct answer, *i.e.*, of the correct asymptotic complexity.

10 points Truth table

10 points Sum-of-products

10 points Product-of-sums

2 Two-level minimization (20 + 10 + 10 = 40 points)

In two-level minimization, a Boolean function is represented as a *covering table*, where columns correspond to prime implicants (or simply, primes) of the function and rows correspond to the function's on-set minterms. A 0 (or 1) entry in the i-th row and j-th column of the table implies that the j-th prime does not cover (or covers) the i-th on-set minterm. The objective is to find a minimum number of columns that covers all the rows. As a first step in two-level minimization, several *pruning techniques* such as *essential primes*, *row dominance*, and *column dominance* are applied to the covering table to result in a *reduced covering table*. The essential primes technique states that if a prime is the only one that covers an on-set minterm, the prime must be included in the final solution. Therefore, the column corresponding to an essential prime can be removed from the covering table (and directly included in the solution), along with any rows that it covers. The row and column dominance techniques work as follows. Row i is said to dominate row j if row i contains a 1 entry in every position where row j has a 1 entry. The same definition of dominance applies for columns. Dominating rows and dominated columns may be removed from the covering table, *i.e.*, they do not need to be considered when coming up with a solution.

Consider the covering table shown in the following figure:

	p1	p2	p3	p4	p5	p6	p7	p8
m1	1	0	0	0	0	0	0	0
m2	1	1	1	0	0	0	0	1
m3	0	1	1	0	0	0	0	0
m4	0	0	1	1	1	0	0	0
m5	0	0	0	0	1	0	1	1
m6	0	0	0	0	0	1	1	0
m7	0	0	0	0	1	1	0	1
m8	0	0	0	0	1	1	1	0

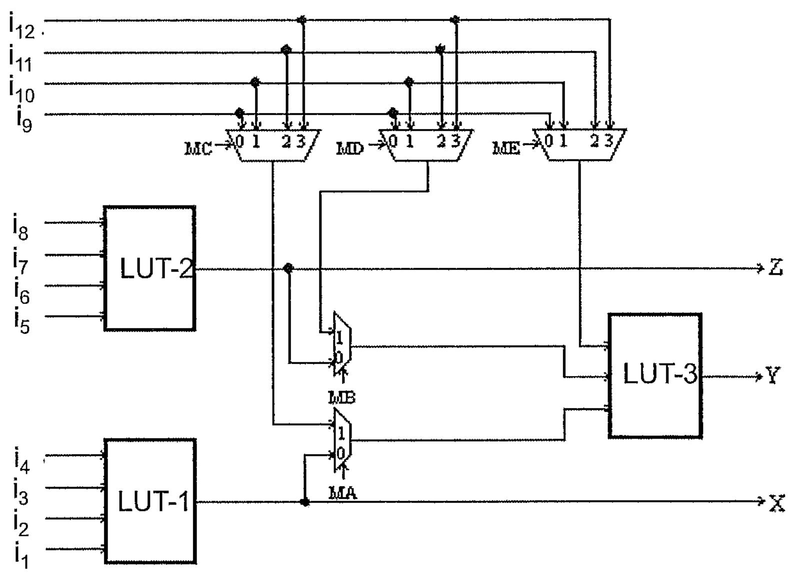
20 points Apply the pruning techniques recursively to the covering table until no further application of pruning techniques is possible. Your answer must clearly show the sequence of pruning techniques applied, and the final reduced covering table.

10 points Why is it OK to remove dominating rows? Why is it OK to remove dominated columns?

10 points How do the pruning techniques help speed up two-level minimization?

3 Field Programmable Gate Arrays (10+20=30 points)

Field programmable gate arrays (FGPAs) are programmable circuit substrates that can be re-configured to implement an arbitrary logic circuit. A fundamental building block of an FPGA is a configurable logic block (CLB), which can implement many different logic functions. Consider the CLB design shown in the following figure:



The CLB consists of (i) two 4-input look-up tables (LUT-1 and LUT-2), which can be programmed to realize arbitrary 4-input Boolean functions, (ii) a 3-input look-up table (LUT-3), which can be programmed to realize an arbitrary 3-input Boolean function, and (iii) five multiplexers (with select signals MA,MB,MC,MD,ME) that allow the inputs, LUTs, and outputs to be connected in different configurations. Note that signals MA and MB can assume the values 0 and 1, while MC, MD, and ME can assume the values 0, 1, 2, or 3. The CLB is configured (as part of the FPGA programming process) by writing the contents of the LUTs and setting the select signals of multiplexers to appropriate values.

Specify how to configure the CLB shown above so as to realize the following:

10 points An arbitrary function F of four input variables (x_1, x_2, x_3, x_4), plus another arbitrary function G of four unrelated input variables (x_5, x_6, x_7, x_8), plus a third arbitrary function H of three other unrelated input variables (x_9, x_{10}, x_{11}).

20 points An arbitrary function J of five variables $(x_1, x_2, x_3, x_4, x_5)$.