Note: You need to show all your steps in deriving a design or analysis. Just writing down the final solution is not enough, and a solution given that way will lose points.

1. (a) Use the divide-and-conquer (D&C) approach to design a combinational multi-bit left-shifting circuit MUBILS with inputs that are an n-bit number \( X \) (the shiftee) and a \( \log_2 n \) bit # \( m \) (the shift amount), and which outputs a \( (2n - 1) \)-bit number \( Z \) that is \( X \) shifted \( m \) times. E.g., if \( X = 11000111 \), and \( m = 110 = 6 \), then \( X \) needs to be shifted by 6 bits and \( Z = 011000111000000 \).

Show the general (for an \( n \)-bit \( X \)) 1st level D&C breakup and the stitch-up of this problem, and also show the leaf-node problem. Based on this, determine the complete D&C tree and circuit for \( n = 16 \), clearly showing the basic design for shifting \( X \) by a 1-bit number (this will be the smallest subproblem at the leaf of your D&C tree) and how two smaller designs are “stitched up” to provide the next larger design, and so forth. **Show all your work in clear and methodical steps.**

**Hint:** Perform the D&C of the design by breaking up the shift amount \( m \).

(b) Assuming the cost of a single small component with a small number of inputs is 1 unit, analyze (and show all your analysis work methodically) the cost of your design as a function of \( n \).

(c) Assuming the delay of a single small component with a small number of inputs is 1 unit, analyze (and show all your analysis work methodically including showing and describing the critical path in your design) the delay of your design as a function of \( n \).

Maximum points can only be obtained for correct, fast and hardware-cost-efficient designs. **Total 300**

2. Using the D&C approach, you need to design a combinational circuit that counts the number of 1’s in an \( n \)-bit number (bit-string) \( A \). The resulting circuit needs to be as fast as possible as far as the structure if the D&C breakup is concerned (i.e., not by using fast components as opposed to a fast D\&C structure). Assume that \( n = 2^k \), for some \( k \). As part of your design derivation, you need to clearly show the following aspects of your D&C approach.

(a) The breakup of the root problem into two or more subproblems, including a clear description of what those subproblems are.

(b) The **stitch-up** function. If the stitch-up function is a well-known or at least well-defined mathematical or logical function that you can describe mathematically and/or logically, then you need not derive the truth table (TT) or exact logic expressions and gate-level implementation of this function. Otherwise, you can just give the TT or logic expressions of the function. Note also, that in the former case, the stitch-up function can either be one well-defined function or a composition of two or more well-defined functions.

(c) The **leaf** function, which is the function that is needed at the bottom of the D&C tree. Again, the leaf function should be described in a similar way as specified for the stitch-up function. Note that the level of the D&C tree at which the leaf function’s realization is appropriate is dependent
on the design problem, and can typically range from a problem of size 1 to 5 bits. In this case, you need to judiciously determine this level so that it makes sense for the current problem. 25
Also note that in some designs the stitch-up and leaf functions are different and in others, they are the same. It is for you to determine what the case is for this problem (this is also partly dependent on your D&C approach).

(d) The schematic of the final design for \( n = 8 \) in terms of the interconnections between the leaf and stitch-up function blocks, so that the structure of the circuit is complete and clearly understandable. 50

(e) Determine the delay of this circuit as a function of \( n \), assuming that the delay of some well-defined basic function(s)/block(s) with a constant number of inputs (i.e., whose number of inputs is not a function of \( n \)) that you use in your stitch-up functions and leaf functions is 1 unit. 50

(f) Analyze the hardware cost of your designs in terms of the number of the aforementioned well-defined basic function blocks (each with a constant number of inputs), i.e., assuming the cost of each such basic block is 1 unit. 25

Maximum points can only be obtained for correct, fast and hardware-cost-efficient designs. Total 200