Report

COL215 Software Assignment 2

Jaskaran Singh Bhalla (2021TT11a139) Saurabh Arge (2022CS11610)

Algorithm

For Part A

We have used a recursion based approach to calculate the required longest delay. We pre-evaluated the delay-area combination with the minimum delay as mentioned in the assignment and used it as delay for each type to solve this problem. We have created one main function which makes calls to recursive helper functions. We make a function call for each primary output and each input end of the D-Flip Flop which, based upon the type of the gate, makes the same function call for its input nodes and then returns the maximum of the delays of its input nodes. Recursively, each node makes a call for its input node unless we reach the primary input node or output node of a D-Flip Flop which has a zero delay and we return 0. At each step the recursion call returns the max of the delays of the input nodes plus the delay of the node itself. So we have solved the problem in the backward direction of the circuit.

For Part B

We have implemented a backtracking algorithm to calculate the minimum combined area of the circuit under the given delay constraint. We have created one main function which makes calls to recursive helper functions. We make a function call for each primary output and each input end of the D-Flip Flop which, based upon the type of the gate, makes the same function call for its input nodes and then returns the calculated minimum area. For calculation of minimum area, we first choose the minimum area for the given component and then make the recursive call for the input component and we proceed in this manner. At the end of recursion at the last step we evaluate whether the delay of the chosen components is below delay constraint or not, if it is not we return back and choose a different combination of areas - delay for the previous component and then recheck for the next component. In this way we go through each and every combination for each component and evaluate the minimum area for each combinational part and combine the minimum area for each of them to get the total minimum area of the circuit.

Time Complexity

For Part A

Since we have used a recursion based approach, time complexity will largely vary based upon the design of the circuit. The worst case time complexity will arise when the circuit recursion tree is highly unbalanced leading to higher depth of recursion and multiple recursive calls per gate. Also in the worst case there will be no D-Flip Flop in the circuit. Since for each node there can be let us say M inputs and during recursion we visit each node once. So time complexity for the code will be of the order of O(M x N). But if we assume that each node can have at max only two input nodes i.e. M = 2 then the time complexity of our code will be O(2 x N) which is O(N). So we have arrived at a linear time complexity for the program.

For Part B

We have implemented a backtracking algorithm that uses a recursion based approach. The worst case time complexity will arise when the delay constraint is very restrained with respect to the delays of individual components. In this case we may have to choose each of the three time-area combinations for all the components in the circuit. If there are n components then there are 3 area-delay combinations for each component. Then we have a total of 3^n possible choices for designing the circuit. So if we visit all the cases then the worst case time complexity would be 3^n where n is the number of components in the circuit.

Note : This time complexity can be made better by reducing the number of computations by using memoization which we haven't been introduced to in the course, so we have refrained ourselves from implementing it.

Testing Strategy

We tested the code on the given sample test case. We have created test cases where we have given input to multiple gates and have multiple outputs. We have considered the cases where we have consecutive gates, two gates taking the same input, a single output going to multiple other gates to multiple outputs. A test case with all zero delays, a single input given. The test case where one output is received. As we have used a recursion based approach, so we have used test cases with deep recursion levels as well. We used test cases with randomised delays and gates as well. We have also considered the cases where there can be cyclic arrangements of individual combinational circuits which is only possible by using a D Flip-Flop.