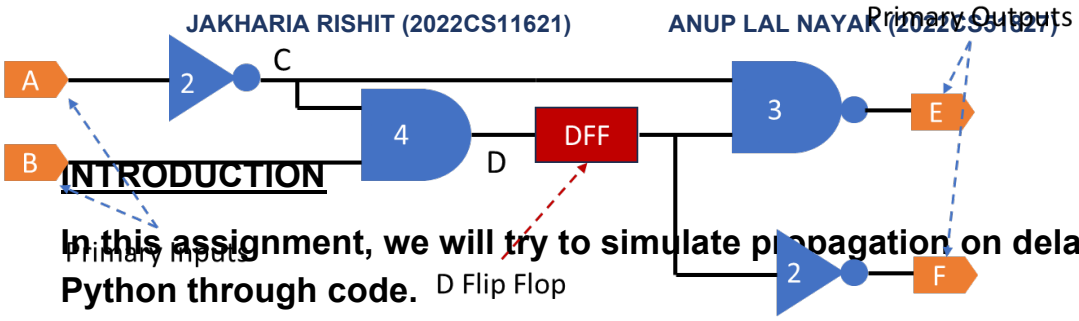
COL215, SUMMER 2023

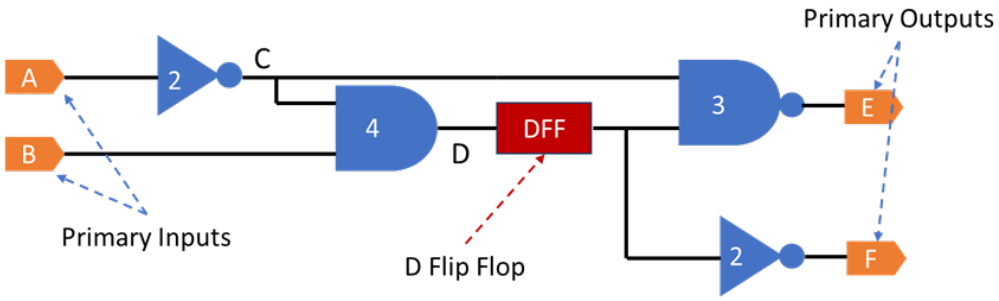
SOFTWARE ASSIGNMENT 2 REPORT

JAKHARIA RISHIT (2022CS11621) ANUP LAL NAYAK (2022CS11621)



INTRODUCTION

In this assignment, we will try to simulate propagation on delay in Python through code.



Problem Description

a) Consider the following variation of the circuits we handled in Assignment 1, with D Flip Flops (DFF in the figure below). The D Flip Flop has one input and one output. We will discuss the DFF internals in detail later in the course, but for now, let us assume that it has zero propagation delay. However, it has the property that the signal arriving at its input represents the END of a combinational path, and the signal at its output represents the START of a new combinational path. Note that the DFF output should be considered available at time 0 as it signifies the beginning of a new combinational path. In this scenario, what is the delay of the longest combinational path in a given circuit? We need to examine the following paths:

- From any primary input to any primary output (not passing through a DFF)
- From any primary input to any DFF input
- From any DFF output to any primary output
- From any DFF output to any DFF input

Input: Circuit file (see example file *circuit.txt* for the diagram above)
Input: Gate Delay file (see example file *gate_delays.txt* for diagram above)
Output: Longest Combinational Delay file (see example file *longest_delay.txt* for diagram above)

G

b) Suppose we have a choice of gate implementations. Each gate (e.g., AND2 gate) now has 3 implementation choices: fast/slow/moderate. Corresponding to these, the gates have 3 areas: large/medium/small. The fastest gate implementation also has the largest area. Design and implement

the solution to the following problem: find the smallest circuit that respects a given longest combinational path delay constraint. That is, select the implementation for each gate in the circuit, such that the total area (= sum of gate areas) is minimum and the longest combinational path delay is less than or equal to a given constraint.
Input: Circuit file (see example file *circuit.txt* for the diagram above)
Input: Gate Delay file (see example file *gate_delays.txt* for diagram above)
Input: Delay Constraint file (see example file *delay_constraint.txt* for diagram above)
Output: Minimum Area file (see example file *minimum_area.txt* for diagram above)

APPROACH AND LOGIC:

a) First, we create a data structure, "Gate," with the required class variables that help us calculate the delay in the given circuit, and then we use the following piece of the algorithm to find the final delay.
   a. Delay of nth signal = Delay of nth gate + Delay (n-1)th signal
   b. Time complexity is O(n), as we review every signal delay once.
   c. For this part, we use the minimum of the given time delays
   d. Ans Upon reaching a DFF, we store the value of the input and mark the output signal as having 0-time delay,
   e. Later, we take the max of DFF inputs and primary outputs
b) In part b, we adopt a brute force approach to check for every value of the gate delays.
   a. Now as we check for all the time delays, we take note of the time delay, area, and implementation
   b. Sort them according to the areas
   c. Now, we check from the start whether the given implementation satisfies the delay constraints

TEST CASES RUN:

a)

```
PRIMARY_INPUTS A B
PRIMARY_OUTPUTS K L
INTERNAL_SIGNALS C E F G H I J

INV A C
AND2 A B E
OR2 B E F
AND2 C E G
NAND2 G F I
NOR2 G I H
NAND2 H J K
AND2 J I L
DFF I J
```

```
NAND2_1 NAND2 3.5 11.2
NAND2_2 NAND2 3 13
NAND2_3 NAND2 4.5 5.3
AND2_1 AND2 16.2 9.5
AND2_2 AND2 7 12
AND2_3 AND2 4 19.6
NOR2_1 NOR2 3.5 10
NOR2_2 NOR2 3 12.5
NOR2_3 NOR2 2.5 15
OR2_1 OR2 4.5 12.8
OR2_2 OR2 7.5 10.3
OR2_3 OR2 3.75 17
INV_1 INV 2 7.33
INV_2 INV 3 4.6
INV_3 INV 3.5 2.5
```

longest delay = `16.5`

b)

```
PRIMARY_INPUTS A B
PRIMARY_OUTPUTS E F
INTERNAL_SIGNALS C D G H

INV A C
NAND2 B G H
AND2 C H D
DFF D G
NAND2 C G E
INV G F
```

```
NAND2_1 NAND2 3.5 11.2
NAND2_2 NAND2 3 13
NAND2_3 NAND2 4.5 5.3
AND2_1 AND2 16.2 9.5
AND2_2 AND2 7 12
AND2_3 AND2 4 19.6
NOR2_1 NOR2 3.5 10
NOR2_2 NOR2 3 12.5
NOR2_3 NOR2 2.5 15
OR2_1 OR2 4.5 12.8
OR2_2 OR2 7.5 10.3
OR2_3 OR2 3.75 17
INV_1 INV 2 7.33
INV_2 INV 3 4.6
INV_3 INV 3.5 2.5
```

Logest delay =

```
7.0
```

CONCLUSION:

Doing this assignment enhanced our understanding of optimizing the circuit for areas basics; we learned in depth about its structure and working.

Reflecting the understood logic into blocks of code was challenging, here we updated the earlier found data structure to now have the value of different time delays and areas.

Debugging was also a phase where we learned a lot about what conceptual or systematic mistakes we were making with our application of the solution, and it helped to refine our understanding even further.

We now have a useful code, when given the information can give us the best choice of implementation of the circuit from the given gate choices