COL215 Software Assignment 3: Timing Optimisation in Gate Positioning

Deadline: 20th October 2024

1 Introduction

In this assignment, we will address timing optimisation in gate positioning. Assume that the input circuit given to us in Assignment 2 is a combinational circuit (no loops).

2 Critical path

- Primary input refers to an input pin to a gate that is not connected to the output pin of some other gate (it is an external input).
- Primary output refers to an output pin from a gate that is not connected to the input pin to some other gate (it is an external output). This is a simplification for our assignment – in reality, intermediate signals could also be primary outputs.
- Path delay from a point to another point is the sum of the gate delays and wire delays along that path.
  - Gate delay (units: nanoseconds) is specified in the input file separately for every gate
  - Assume that wire delay is proportional to the estimated length of the wire, and the delay per unit length is specified in the input file.
- If a wire connects a source with multiple destinations, then the signal arrives at all destinations simultaneously.
- Critical path delay for a circuit is defined as the longest path delay from any primary input to primary output of the circuit.

3 Problem Statement

Given:

- a set of rectangular logic gates $g_1, g_2 ... g_n$
- ***width*** and ***height*** of each gate g_i
- the input and output pin locations (x and y co-ordinates) on the boundary of each gate $g_i.p_1, g_i.p_2, ..., g_i.p_m$ (where gate g_i has m pins)
- gate delay $D_{g_1}, D_{g_2} ... D_{g_n}$
- wire delay per unit length D_{wire}
- the pin-level connections between the gates

write a program to assign locations to all gates in a plane so that:

- no two gates are overlapping
- the ***Critical path*** of whole circuit is ***minimised***.

3.1 Notes

- Assume that the gates cannot be re-oriented (rotated, etc.) in any way.
- Assuming that all wiring is horizontal and vertical
- Possible estimate for the wire length for a set of connected pins uses the ***semi-perimeter method***: form a rectangular bounding box of all the pin locations; the estimated wire length is half the perimeter of this rectangle.

4 Mathematical Formulation

Consider a structure consisting of gates and M wires with length L_m, each connecting a set of pins, gate delay as D_{g_i} and wire delay as D_{wire}. A path $\mathbf{P}$ is a sequence of gates and wires from the Input Pin to the Output Pin and $\mathcal{P}$ denote the set of all possible paths from input pins to output pins.

For a path $\mathbf{P}$, delay is calculated as

$$T_P = \sum_{(g_i, w_m) \in \mathbf{P}} (D_{g_i} + D_{wire} \cdot L_{w_m})$$

Where, g_i and w_m are the gates and wires in Path $\mathbf{P}$. Further, critical path delay is represented as:

$$T_{cp} = \max_{\mathbf{P} \in \mathcal{P}} T_P$$

Finally, the objective function is

$$T_{mincp} = \min T_{cp} = \min(\max_{\mathbf{P} \in \mathcal{P}} \sum_{(g_i, w_m) \epsilon \mathbf{P}} (D_{g_i} + D_{wire} \cdot L_{w_m}))$$

For estimating wire length use **Semi Perimeter Method**.

5 Formats

5.1 Input file format

For each gate, the input file contains the **width, height, delay and the corresponding pins co-ordinates.**

```
<name of gate> <width> <height> <delay>
<wire_delay> <delay>
<pins> <name of gate> <x_1, y_1> ... <x_m, y_m>
<wire> <g_x.p_x> <g_y.p_y>
```

5.2 Output file format

The output file begins with a specification of the bounding box, of the form:

```
bounding_box <width> <height>
critical_path <g_x.p_x> <g_y.p_y> ... <g_z.p_z>
critical_path_delay <delay>
```

After the above lines, each line of the output file from your program should have the location of the gate, specified as the x- and y-co-ordinates of the bottom left corner as follows:

```
<name of gate> <x co-ordinate> <y co-ordinate>
```

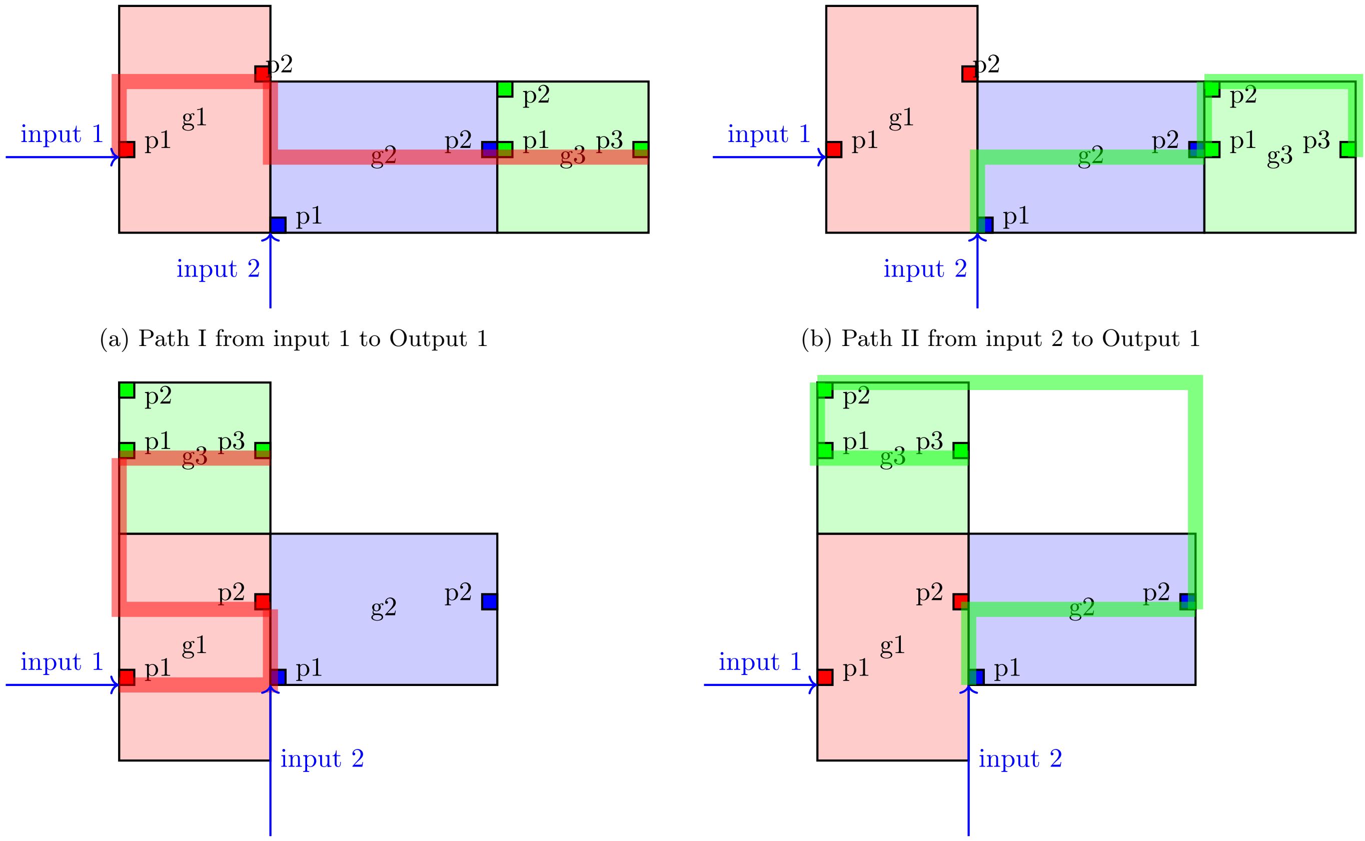
6 Test Case format

Input specifications for sample test is :

```
g1 2 3 5
pins g1 0 1 2 2
g2 3 2 3
pins g2 0 0 3 1
g3 2 2 6
pins g3 0 1 0 2 2 1
wire_delay  4
wire g1.p2 g3.p1
wire g2.p2 g3.p2
```

We are given 3 gates, $g1$, $g2$, $g3$ with delay 5, 3, and 6 ns respectively. The delay of wire(per unit length) is 4ns

Figure 1 shows 4 different configurations of placing the gates. The blue colored wires are primary input/output wires. The red and green path shows two different paths from multiple inputs to single output.



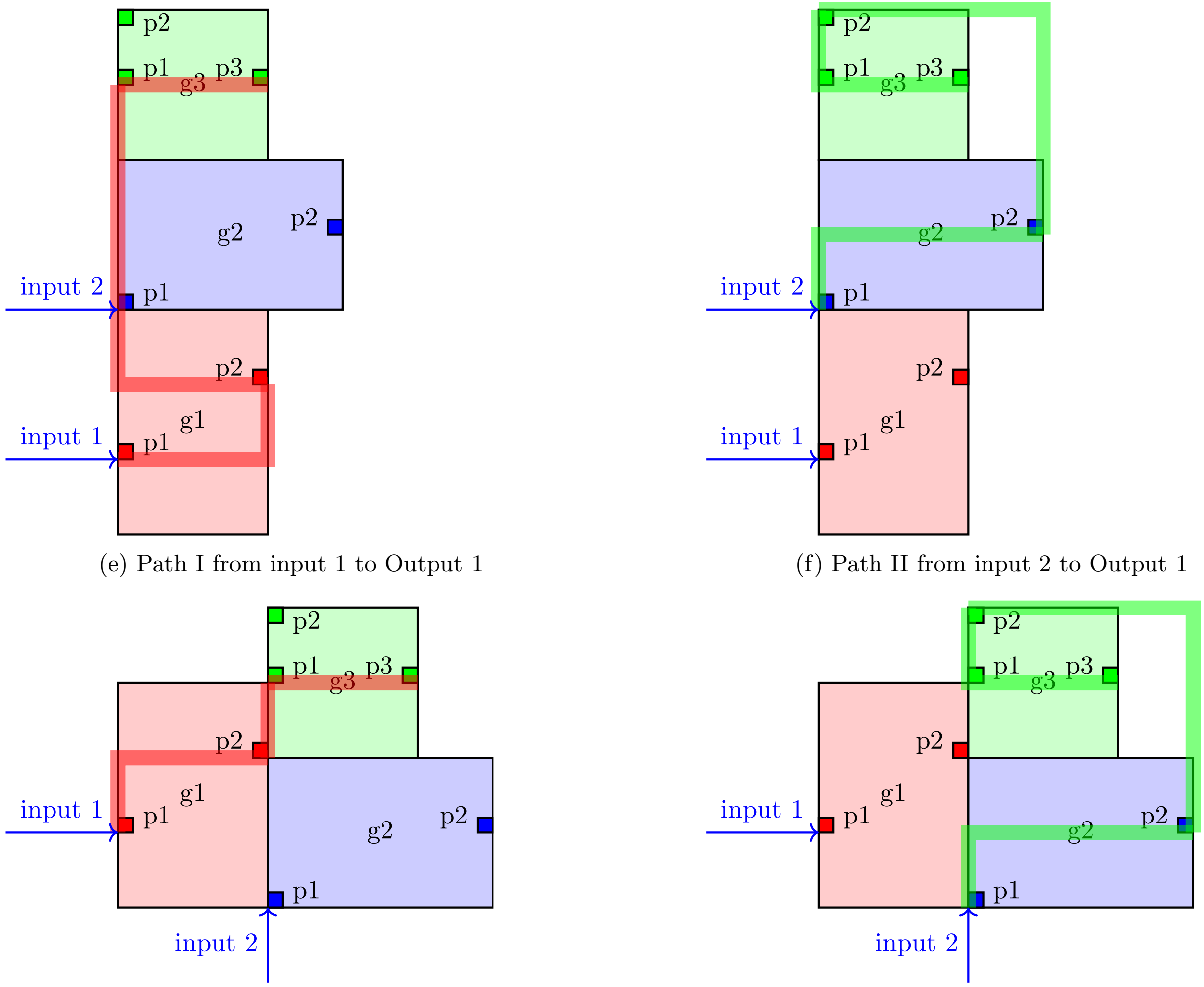
(a) Path I from input 1 to Output 1

(b) Path II from input 2 to Output 1

(c) Path I from input 1 to Output 1

(d) Path II from input 2 to Output 1

Figure 1: Various configurations for the given input specification



(e) Path I from input 1 to Output 1

(f) Path II from input 2 to Output 1

(g) Path I from input 1 to Output 1

(h) Path II from input 2 to Output 1

Figure 1: Various configurations for the given input specification

Please note that coordinates of a pin are relative to the bottom-left corner of that gate, which is assumed to be at (0, 0) to determine the position of pins for that gate. The Table 1 shows how to estimate the delay for each path in various configuration.

Table 1: Calculating Delays for Configurations in Figure 1

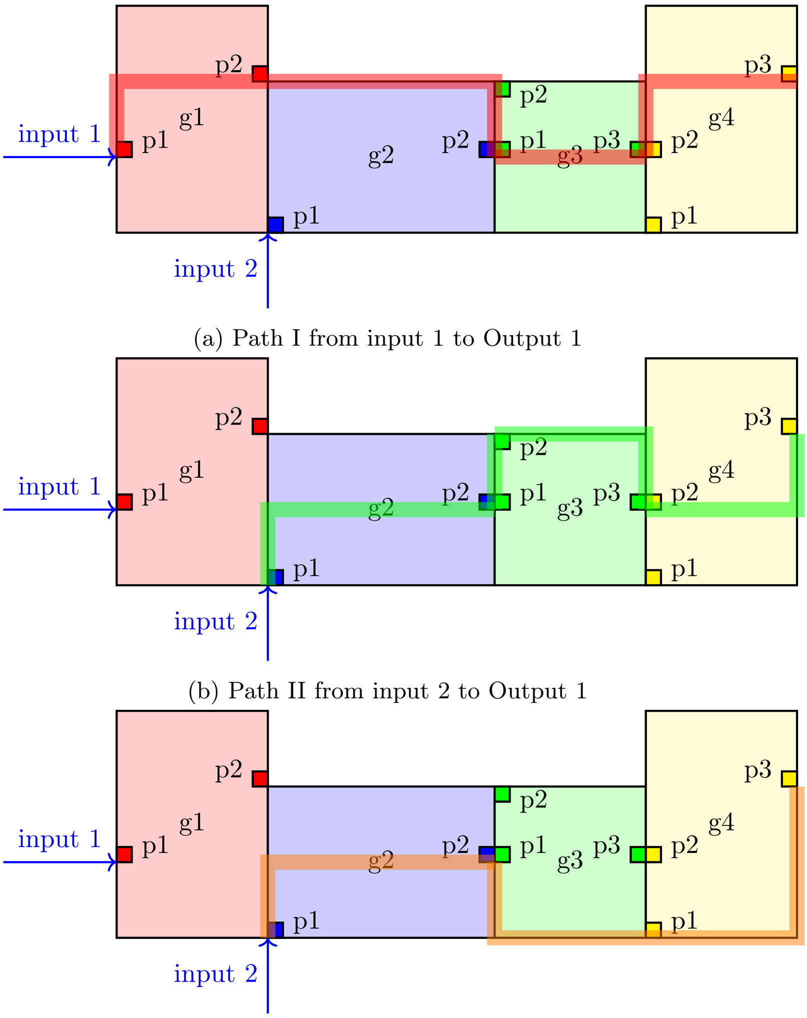
Configuration	Path	Wire and Gates in the path	Delays	Critical Delay
1	Path I	$g_1 \rightarrow g_1p_2 - g_3p_1 \rightarrow g_3$	$5 + 4 \cdot 4 + 6 = 27$	$max(27, 13) = 27$
	Path II	$g_2 \rightarrow g_2p_2 - g_3p_2 \rightarrow g_3$	$3 + 1 \cdot 4 + 6 = 13$	
2	Path I	$g_1 \rightarrow g_1p_2 - g_3p_1 \rightarrow g_3$	$5 + 4 \cdot 4 + 6 = 27$	$max(27, 41) = 41$
	Path II	$g_2 \rightarrow g_2p_2 - g_3p_2 \rightarrow g_3$	$3 + 8 \cdot 4 + 6 = 41$	
3	Path I	$g_1 \rightarrow g_1p_2 - g_3p_1 \rightarrow g_3$	$5 + 6 \cdot 4 + 6 = 35$	$max(35, 33) = 35$
	Path II	$g_2 \rightarrow g_2p_2 - g_3p_2 \rightarrow g_3$	$3 + 6 \cdot 4 + 6 = 33$	
4	Path I	$g_1 \rightarrow g_1p_2 - g_3p_1 \rightarrow g_3$	$5 + 1 \cdot 4 + 6 = 15$	$max(15, 33) = 33$
	Path II	$g_2 \rightarrow g_2p_2 - g_3p_2 \rightarrow g_3$	$3 + 6 \cdot 4 + 6 = 33$	

From Table 1, we can see that the minimum delay is obtained in 1^{st} configuration and is equal to 27. The critical path changes based on the gate placement. THe critical path should include the pin connections starting with primary input and ending with primary output (omitting the gates). Therefore, output.txt (based on 1^{st} configuration will be:

```
bounding_box 7 3
critical_path g1.p1 g1.p2 g3.p1 g3.p3
critical_path_delay 27
g1 0 0
g2 2 0
g3 5 0
```

Referring to sample test 2 specification given below, there is a set of three pins (g2.p2, g3,p2 and g4,p1) connected together. Figure 2 and 3 shows two possible gate placement configuration and the corresponding paths. The table 2 shows the delay calculation for each path, it can be seen that semi-perimeter wire length is used to calculate delay between g2.p2 & g4.p1 and g2.p2 & g3.p2. While estimating wire delay for multiple connected pins, you need to use the semi-perimeter wire length as signal from output pin arrive simultaneously at each input pin.

```
g1 2 3 5
pins g1 0 1 2 2
g2 3 2 3
pins g2 0 0 3 1
g3 2 2 6
pins g3 0 1 0 2 2 1
g4 3 1 4
pins g4 0 0 2 1 2 2
wire_delay 3
wire g1.p2 g3.p1
wire g2.p2 g3.p2
wire g2.p2 g4.p1
wire g3.p3 g4.p2
```

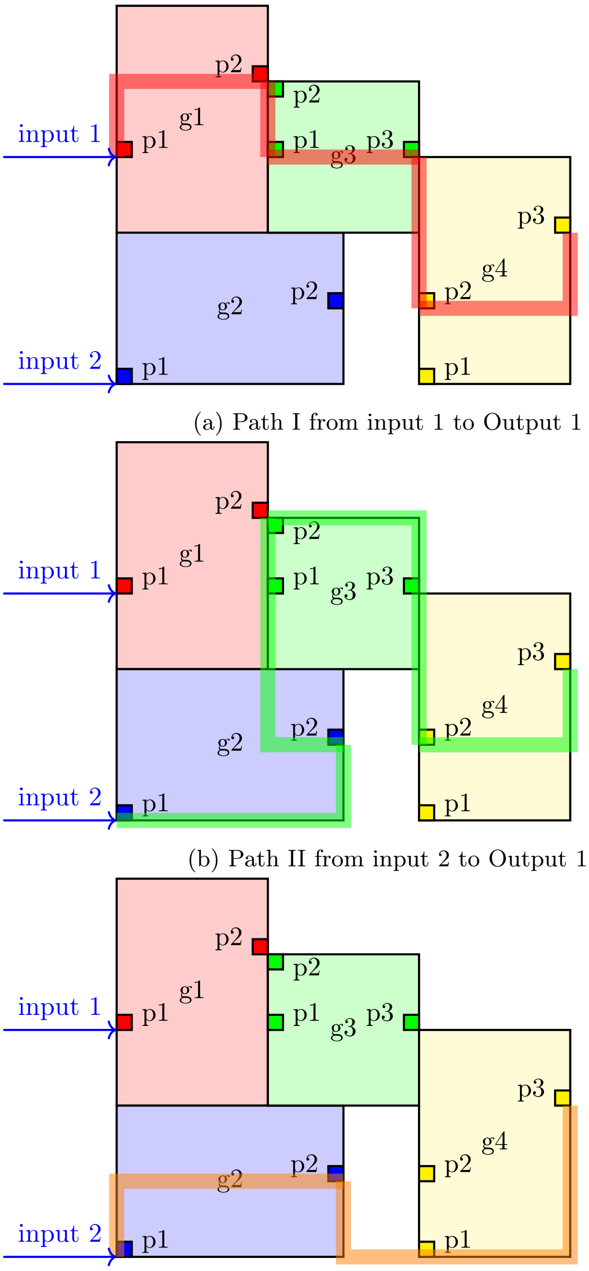


(a) Path I from input 1 to Output 1

(b) Path II from input 2 to Output 1

(c) Path III from input 2 to Output 1

Figure 2: Sample test 2: Configuration 1



(a) Path I from input 1 to Output 1

(b) Path II from input 2 to Output 1

(c) Path III from input 2 to Output 1

Figure 3: Sample test 2: Configuration 2

Table 2: Calculating Delays for Configurations in Figure 2 and 3

Configuration	Path	Wire and Gates in the path	Delays	Critical Delay
1	Path I	$g_1 \to g_1p_2 - g_3p_1 \to g_3 \to g_3p_3 - g_4p_2 \to g4$	$5 + 4 \cdot 3 + 6 + 4 = 27$	$max(27, 25, 19) =$ 27
	Path II	$g_2 \to g_2p_2 - g_3p_2 \to g_3 \to g_3p_3 - g_4p_2 \to g4$	$3 + 4 \cdot 3 + 6 + 4 = 25$	
	Path III	$g_2 \to g_2p_2 - g_4p_1 \to g_4$	$3 + 4 \cdot 3 + 4 = 19$	
2	Path I	$g_1 \to g_1p_2 - g_3p_1 \to g_3 \to g_3p_3 - g_4p_2 \to g4$	$5 + 1 \cdot 3 + 6 + 2 \cdot 3 + 4 = 24$	$max(24, 31, 25) =$ 31
	Path II	$g_2 \to g_2p_2 - g_3p_2 \to g_3 \to g_3p_3 - g_4p_2 \to g4$	$3 + 6 \cdot 3 + 6 + 2 \cdot 3 + 4 = 31$	
	Path III	$g_2 \to g_2p_2 - g_4p_1 \to g_4$	$3 + 6 \cdot 3 + 4 = 25$	

7 Testing instructions

Initial sample test cases will be uploaded on moodle. Additionally, you are required to generate your own test cases to verify the implementation. You need to provide justification (in report) for the generated test cases.

Following are the input constraints:

- **Corners of gate have integral coordinates**
- **Pins will have integral coordinates relative to the corresponding gate**
- **0 $<$ Number of gates $\leq$ 1000**
- **0 $<$ Width of gate $\leq$ 100**
- **0 $<$ Height of gate $\leq$ 100**
- **0 $<$ Number of pins on one side of a gate $\leq$ Height of Gate**
- **0 $<$ Total Number of pins $\leq$ 40000**
- **There is atleast 1 wire connecting a gate**

8 Assignment Submission Instructions

General assignment instructions that need to be followed for all assignments: only one partner needs to submit. Mention all team member names and entry IDs during the submission.

1. Name the submission file as entryNumber1_entryNumber2.zip or entryNumber1.zip
2. Go to Gradescope via moodle and upload the file under Software Assignment 3.
3. Only one submission per group is required. Gradescope will allow you to select the group partner.
4. The following files should be part of the zip folder:
    - Source files
    - Report as a .pdf file (handwritten report will be rejected). The report needs to state:
        - your design decisions
        - time complexity analysis
        - test cases