Combinational Digital Circuit Delay Analysis

Abhishek Amrendra Kumar

2022CS11598

Ayush Gupta

2022CS11114

Date: August 20, 2023

Contents

1 Part 1: Computing Output Delays

1.1 Algorithm Description

In this section, we analyze the algorithm designed to compute the output delays of a combinational digital circuit. The algorithm takes into account the gate delays provided in the 'gate_delays.txt' file, as well as the description of the circuit provided in the 'circuit.txt' file.

The algorithm follows these steps:

1. Read the gate delays from 'gate_delays.txt' for different gate types.
2. Parse the circuit description from 'circuit.txt', including primary inputs, primary outputs, and internal signals, along with their associated gate types and connections.
3. Create a list of signals to delay, along with their corresponding gate delay.
4. Calculate the maximum delay for each signal by considering the propagation delays through its connected gates.
5. Store the computed max delays for each signal in a map for future reference.
6. Print the delays for each primary output signal.

1.2 Time Complexity

The time complexity of the algorithm depends on the number of gates and connections in the circuit. The algorithm processes each signal only once, and for each signal, it iterates through its connected gates to calculate the delay. Therefore, the time complexity can be approximated as O(N), where N is the number of gates/signals in the circuit.

1.3 Testing Strategy

To test the algorithm, various circuits with different gate types and signal connections were created. These circuits were designed to cover different cases, including simple cases with a few gates and more complex cases with multiple interconnected gates. Test cases were written to compare the output delays computed by the algorithm with expected results for each primary output signal. The testing strategy ensured that the algorithm

correctly handles different gate types and circuit configurations. curate delay calculations across different circuit complexities.

1.4 Test-Case-1

Circuit Description

```
circuit.txt
PRIMARY_INPUTS A B
PRIMARY_OUTPUTS H
INTERNAL_SIGNALS C D E F G
OR2 A B C
AND2 A C D
NOR2 B C E
NAND2 C E F
OR2 D F G
INV G H

gate_delays.txt

NAND2 4
AND2 1
NOR2 3
OR2 2
INV 2
```

1.5 Test-Case-2

Circuit Description

```
circuit.txt
PRIMARY_INPUTS A B
PRIMARY_OUTPUTS G H
INTERNAL_SIGNALS C D E F
OR2 A B C
INV B E
AND2 C E D
NOR2 E C F
NAND2 D F G
INV F H
```

```
gate_delays.txt
    NAND2 3
    AND2 2
    NOR2 5
    OR2 3
    INV 100
```

The test cases cover various scenarios, including circuits of moderate to larger sizes, to ensure accurate delay calculations.

2 Part 2: Computing Input Delays

2.1 Algorithm Description

In this section, we analyze the algorithm designed to compute the input delays of a combinational digital circuit. The algorithm follows these steps:

1. Read the gate delays from 'gate_delays.txt' for different gate types.
2. Parse the circuit description from 'circuit.txt', including primary inputs, primary outputs, and internal signals, along with their associated gate types and connections.
3. Create a list of signals to delay, along with their corresponding gate delay.
4. Calculate the delay for each input signal by considering the propagation delays through the circuit to reach the required delays at primary outputs.
5. Store the computed delays for each input signal in a map.
6. Print the delays for each primary input signal.
7. Write the computed input delays to 'inputs_delays.txt' for future reference.

2.2 Time Complexity

Similar to Part 1, the time complexity of the algorithm for Part 2 also depends on the number of gates/signals in the circuit. The algorithm processes each signal only once, and for each signal, it iterates through its connected gates to calculate the required delay. Therefore, the time complex-

ity can be approximated as O(N), where N is the number of gates/signals in the circuit.

2.3 Testing Strategy

The testing strategy for Part 2 is similar to that of Part 1. The algorithm was tested on various circuits with different gate types and signal connections, and the computed input delays were compared with expected results for each primary input signal. The testing strategy ensured that the algorithm correctly computes the required input delays to satisfy the specified criteria.

2.4 Test-Case:1

Circuit Description

```
PRIMARY_INPUTS A B
PRIMARY_OUTPUTS H
INTERNAL_SIGNALS C D E F G
OR2 A B C
AND2 A C D
NOR2 B C E
NAND2 C E F
OR2 D F G
INV G H
```

Gate Delays

```
gate_delays.txt

NAND2 4
AND2 1
NOR2 3
OR2 2
INV 2
```

Required Delays

```
H 17
```

2.5 Test-Case-2

Circuit Description

```
PRIMARY_INPUTS A B
PRIMARY_OUTPUTS G H
INTERNAL_SIGNALS C D E F
OR2 A B C
INV B E
AND2 C E D
NOR2 E C F
NAND2 D F G
INV F H
```

Gate Delays

gate_delays.txt

```
NAND2 3
AND2 2
NOR2 5
OR2 3
INV 100
```

Required Delays

```
G 150
H 250
```