Alessandro Meucci and Sam Rojanasadakul
Professor Saab
VLSI Lab 6
12 December, 2023

Part 1:

Overview: The code successfully completes all three requirements for this part as seen in the pictures below for circuit s27, and in the attached output file for s35 (which is too large to include in this report).

Approach: In order to adequately satisfy the requirements, a scanner was implemented in the code, and for each word that was processed, the function would determine the type of gate, its name, and its fanin and fanout. Afterwards, the levels and a separate fanin array were created by iterating through nested linked lists that contained all of the gates and adding 1 to the maximum level of the gates in its fanin.

After this was completed, each gate was assigned a number, and all gates were printed to the user with their gate name, gate type, level, whether they were an output or not, number, fanin, and fanout. Using this data, the next objective of this part was to print the number of gates at each level, and finally the total number of gates. For this I have included the entirety of the small circuit's required outputs, and a part of the large circuit's output (the rest is in the attached file).

For small circuit:

```
Gate_records in the linked list:
GateName: G0 , GateType: 0, Level: 0, Output: false, Number: 0, Fanout: G0, Fanin: None
GateName: G1 , GateType: 0, Level: 0, Output: false, Number: 1, Fanout: G1, Fanin: None
GateName: G2 , GateType: 0, Level: 0, Output: false, Number: 2, Fanout: G2, Fanin: None
GateName: G3 , GateType: 0, Level: 0, Output: false, Number: 3, Fanout: G3, Fanin: None
GateName: G5 , GateType: 1, Level: 0, Output: false, Number: 4, Fanout: G5, Fanin: G10
GateName: G6 , GateType: 1, Level: 0, Output: false, Number: 5, Fanout: G6, Fanin: G11
GateName: G7 , GateType: 1, Level: 0, Output: false, Number: 6, Fanout: G7, Fanin: G13
GateName: G14 , GateType: 6, Level: 1, Output: false, Number: 7, Fanout: G14, Fanin: G0
GateName: G8 , GateType: 3, Level: 2, Output: false, Number: 8, Fanout: G8, Fanin: G6 G14
GateName: G12 , GateType: 4, Level: 1, Output: false, Number: 9, Fanout: G12, Fanin: G7 G1
GateName: G15 , GateType: 2, Level: 3, Output: false, Number: 10, Fanout: G15, Fanin: G8 G12
GateName: G16 , GateType: 2, Level: 3, Output: false, Number: 11, Fanout: G16, Fanin: G8 G3
GateName: G13 , GateType: 4, Level: 2, Output: false, Number: 12, Fanout: G13, Fanin: G12 G2
GateName: G9 , GateType: 5, Level: 4, Output: false, Number: 13, Fanout: G9, Fanin: G15 G16
GateName: G11 , GateType: 4, Level: 5, Output: false, Number: 14, Fanout: G11, Fanin: G9 G5
GateName: G10 , GateType: 4, Level: 6, Output: false, Number: 15, Fanout: G10, Fanin: G11 G14
GateName: G17 , GateType: 6, Level: 6, Output: true, Number: 16, Fanout: G17, Fanin: G11
```

```
Number of Gates at each level:
Level 0: 7 gates
Level 1: 2 gates
Level 2: 2 gates
Level 3: 2 gates
Level 4: 1 gates
Level 5: 1 gates
Level 6: 2 gates

Total number of gates: 17
```

For large circuit:

```
Number of Gates at each level:
Level 0: 1763 gates
Level 1: 1782 gates
Level 2: 927 gates
Level 3: 2880 gates
Level 4: 896 gates
Level 5: 608 gates
Level 6: 374 gates
Level 7: 603 gates
Level 8: 315 gates
Level 9: 288 gates
Level 10: 576 gates
Level 11: 288 gates
Level 12: 288 gates
Level 13: 288 gates
Level 14: 288 gates
Level 15: 288 gates
Level 16: 576 gates
Level 17: 576 gates
Level 18: 288 gates
Level 19: 288 gates
Level 20: 288 gates
Level 21: 288 gates
Level 22: 288 gates
Level 23: 544 gates
Level 24: 544 gates
Level 25: 544 gates
Level 26: 288 gates
Level 27: 288 gates
Level 28: 288 gates
Level 29: 288 gates

Total number of gates: 17828
```

(List of Gates included in the attached file)

Part 2:

Overview: The second part of this lab was to simulate a state, an output, and a runtime for each input given in the input files designated for each circuit. The output for the small circuit is included below (output for large is included in the attached file), and it can be seen that the output for the small circuit is correct and matches the designated requirements from the lab perfectly. For reference, this is what was expected:

```
input file for s27: G0, G1, G2, G3

        0000
        0010
        0100
        1000
        1111
```

```
output file: 4 represents undefined

        INPUT   :0000 // G0, G1, G2, G3
        STATE   :444  // G5, G6, G7
        OUTPUT  :4    // G17

        INPUT   :0010
        STATE   :044
        OUTPUT  :4

        INPUT   :0100
        STATE   :040
        OUTPUT  :4

        INPUT   :1000
        STATE   :041
        OUTPUT  :1

        INPUT   :1111
        STATE   :101
        OUTPUT  :1
```

Approach: For the code to correctly satisfy the requirements, the function needed to be able to not only obtain each new input from the input file, but for each input it needed to assign outputs to gates level by level, using the outputs from gates from previous levels. The code needed to constantly update outputs of gates as it traversed the circuit, and the only way to do this efficiently was to keep the outputs of every gate in an array that was sorted with respect to each gate's designated number. This way, the code could access outputs of previous gates immediately without having to iterate through the linked list each time, thus drastically reducing runtime. After each run the DFFs were updated using outputs of gates from the state before, and the process continued until the output was printed for each input.

Below is the output from the smaller circuit, which can be seen to match the above requirement exactly, and attached is the output of the large circuit. Both include their respective runtimes.

```
Inputs: 0000
State: 444
Output: 4

[Inputs: 0010
[State: 044
[Output: 4
[
[Inputs: 0100
[State: 040
[Output: 4

Inputs: 1000
State: 041
[Output: 1

Inputs: 1111
State: 101
Output: 1
[
[Execution Time: 0.000213 seconds
```

```
Inputs: 01111010101100011111010101101010110
State: 1000111111100100101010111000101101010010111001110001111110111001011110100110110110111011
10110100100010010110110001010000100011000010101000101101010100111101101001000001111111100001011
01111010101111100000011101111011000100011000100010101001100001111000101011000010011010001000111
01110001110101011000110010100001000110101101010000000011101011101100010101011110111111111111110
01110100001001011101101011011011110001010001111000000110000010011111111111111000000000000000001
01110001100001110111111101001101111100101110111111111111111000000000000000000111110101101101110
01100011100111100110010001011111111111111000000000000000000000001111001100110001011001101100101
10101100111111111111111000000000000000000110011100011101100101011111001110100111100000000000111
11111000000000000000001001100110000001010110111010001001100111011010010001111111111111010111111
00000010011111010011110000110101111111110111000001100111111110010011000110000100011110100000111
Output: 1000111111100100101010111000101101010010111001110001111110111001011110100110110110111011
11011010010001001011011000101000010001100001010100010110101010011110110100100000111111110000101

Execution Time: 14.486697 seconds
[atm125@g519d-04 Lab6]$
```

Comparing Runtimes: As can be seen from the pictures above, the runtimes vary quite a bit. For the small circuit the runtime is 0.000213 seconds or 213 microseconds. For the large circuit the runtime was 14.486 seconds. The discrepancy between the two is largely due to the number of gates, and how, with nested linked list iterations for gate assignments, the runtime for the large circuit increases exponentially with respect to the number of gates.