

Special Topics in Computer Aided Design - HW2

B101016 曹寓恆

Net 和 Wire 是在讀取 verilog file 時的輔助，因為不確定 verilog 的語法會有後級和前級交替出現的情況，所以在讀取時只有先建立 Gate 的空殼並把資訊先暫存在 wire 中，並用 input 及 output 儲存所有的輸入級和輸出級，在讀取完成後再將中間的各個 gate 接線。在完成接線後整個電路的接線資訊會由各個 Gate 的 I_ptr、A1_ptr、A2_ptr、ZN_ptr 儲存連接關係。因為算 delay 需要考慮到前後級的順序，所以會再將 topological sort 的結果存放在 topo_order 中。

```
struct Wire
{
    string FromGateID;
    vector<pair<int, Gate *>> Pin_ToGate;
};
struct Net
{
    map<string, Gate *> input; // net_id,
    map<string, Gate *> output; // net_id,
    map<string, Wire> wire; // net_id,
    map<string, Gate *> gate; // gate's
    list<Gate *> topo_order; // result
};
```

接著要算各個 gate 的輸出電容，Gate 中的 ZN_ptr 存有 fan out 資訊以便完成計算。再來要算 delay，在算完每一個 gate 後會將當前的 path delay 存至 Sensitive_Delay 還有把 delay 來源腳位存在 sensitization 中方便最後尋找 longest path，在算完整個電路之後，由 output 擁有最大的 Sensitive_Delay 的 gate 就是 Longest path 的輸出級，再由 sensitization 一級一級回推出 longest path 完成輸出。

```
struct Gate
{
    string ID, Output_Net_ID;
    int type, indegree, sensitization; // indegree is for topological sort, sensitization
    double C_ZN, Propagation_Delay, Transition_Time, Longest_Path_Delay, Sensitive_Delay;
    bool ZN;
    Gate *I_ptr, *A1_ptr, *A2_ptr; // ptr of from_gate, can get value by C_A1 == A1_ptr
    vector<pair<int, Gate *>> ZN_ptr; // Pin_ToGate
}
```

最後，有同學提到會有誤差的情況，我認為不是運算精度問題，插值順序不同結果就可能不同，以 example.v / g1 / 第一組 input 為例(先算黃色再算橘色)兩種插值方式一個能整除一個是循環小數，採用下圖右的方式可以得到和 golden 完全一致的結果，建議之後可以提供統一的插值方式。

| 0.0191325195555555 | | | | | 0.01913250625 | | | | |
|--------------------|--------|----------|----------|----------|---------------|--------|----------|----------|----------|
| D | E | F | G | H | D | E | F | G | H |
| 0.003 | 0.0062 | 0.0125 | 0.017337 | 0.0251 | 0.003 | 0.0062 | 0.0125 | 0.017337 | 0.0251 |
| | | 0.016677 | 0.019133 | 0.023075 | | | 0.016677 | 0.019133 | 0.023075 |
| | | 0.022481 | 0.02603 | 0.031726 | | | 0.022481 | 0.02603 | 0.031726 |
| | | 0.026053 | 0.030274 | 0.03705 | | | 0.026053 | 0.030274 | 0.03705 |