

CAD HW3

Static Timing Analysis

電子碩一 312510191 杜家漢

前言:

此次作業使用的程式架構為 C++，基本概念和前一次作業一樣，只是再多加入了邏輯正確性。

實作流程:

1. 讀取檔案內容:
2. Calculate Load:
3. 計算每個 input pattern 所對應到的 delay:
4. 計算 power
5. 計算 total power 以及 toggle

這次程式碼主要的概念都是沿用作業 2 的概念，包括讀檔案、計算 load。加入的 input pattern 也是需要透過逐行讀取的方式加入到程式碼中運行。所以在 struct 結構中加入了判斷目前輸入的部分。再來則是計算 delay 的部分，這次要加入新的判斷方式計算 delay，因為程式碼比較大，就沒特別截圖放入。

我寫的概念主要可以分為三個部分，也就是 3 個邏輯閘。Inv gate 相對簡單，因為只有一個輸入，不需要特別判斷是不是 controlling value，所以可以直接透過輸入 node 的值計算 delay。

Nand 和 nor gate 就要分成幾種情況判斷

1. 兩者都是 input node
2. 一個 input node(判斷何者是 controlling value)
3. 皆非 input node(判斷何者是 controlling value)

2 和 3 需要考慮 controlling value，並且加入到整體路徑的參數中記錄，後面的邏輯閘才可以正確地找到前面產生的 delay。

計算 power 也是沿用前面 delay 的計算方式，得到正確的結果。

最後計算 toggle 跟 total power，就是加入一個 gate value，判斷是不是 controlling value。對應甚麼輸入的值，如果跟原本的值不一樣就代表變換一次

`toggle+1`，如果達到 20 次時就不再變換 `toggle` 數值。最後計算 `toggle_rate`。

心得:

這次的作業再度使用了 `sta` 技巧，讓我練習到 EDA Tool 是如何運用在實際的電路模型中。此次作業有考慮到邏輯閘的正確性，所以實際要考慮的問題就變得非常多，而且這僅僅是三個 `gate` 而已，還有實際應用中會碰到的問題都沒有加進去。如果電路變大，整個運算時間就會變長，而且電路中就會有更多干擾影響到 `delay` 和 `power` 的計算方式。這也是我們值得思考並設法想解決的問題。