2025 Digital IC Design Homework 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | 黃啟桓 | | | |
| Student ID | P76134927 | | | |
| **Simulation Result** | | | | |
| Functional simulation | | Pass | Pre-Layout simulation | Pass |
| Cycle : 1043 | | | Please specify your clock width: 14 (ns)  Cycle : 1044 | |
| **Synthesis Result** | | | | |
| Total logic elements | | | 2416 | |
| Total memory bits | | | 0 | |
| Total registers | | | 1716 | |
| Embedded multiplier 9-bit elements | | | 60 | |
|  | | | | |
| **Description of your design** | | | | |
| 最大的改良是我讓每個stage都使用同樣的算法，每個stage都有8個需要乘Weight，8個不需要乘Weight。只要設計的電路不要按照圖示，即index不要按照順序 0 ~ 15，而是規律的變換。變動的是Weight，這樣就能使用更少的乘法器。  下圖是簡潔的示意圖，但為了縮短time cost，所以實際分成3步，使用non-blocking assignment。    https://cdn.discordapp.com/attachments/778196329250816000/1367631062699741316/image.png?ex=6815492e&is=6813f7ae&hm=78692ccdf73e8e99d475979f9a85924d3f3186b39726e5aedcc34938aa96e085&  可以理解成，中間的stage他的output的index是變化的，不是單純的 0 ~ 15，index保持一樣的規律，舉例來說: index:0永遠和8相乘，輸出是index 0。Index: 1永遠和9相乘，輸出永遠是2…。基本上定義完stage 1的狀態就能確定後面的輸出的狀態。所以stage 1的output index是[0, 2, 4, 6, 8, 10, 12, 14, 1, 3, 5, 7, 9, 11, 13, 15]。你會發現變化的index後，stage 2和stage 1的算法一樣，都是index 8~15需要乘上一個Weight。  而weight就是根據上面的規則變化。比如Index 8 他就要和 W[0] 相乘。至於W[0]是甚麼，看stage是甚麼。 | | | | |