

2025/10/16

實驗六

序向邏輯練習

姓名：林承羿

學號：01257027

班級：資工3A

E-mail：IanLin6225@gmail.com

※注意

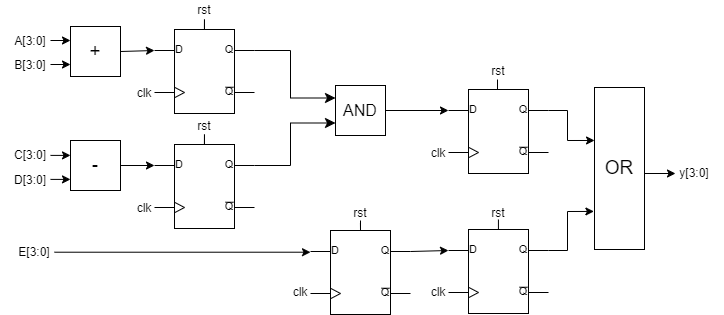
1. 繳交時一律轉PDF檔
2. 繳交期限為下周上課前
3. 一人繳交一份
4. 檔名請按照作業檔名格式進行填寫，未依照格式不予批改
5. 檔名範例：學號\_姓名\_HW6
6. **Pipeline**

* **實驗說明：**

1. 使用pipeline做10筆((A+B)&(C-D))|E的運算。
2. 測資：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 第1筆 | 第2筆 | 第3筆 | 第4筆 | 第5筆 | 第6筆 | 第7筆 | 第8筆 | 第9筆 | 第10筆 |
| Ａ | 6 | 4 | 1 | 8 | 6 | 11 | 6 | 11 | 7 | 2 |
| Ｂ | 7 | 8 | 9 | 7 | 10 | 9 | 8 | 10 | 4 | 8 |
| Ｃ | 8 | 7 | 6 | 3 | 3 | 6 | 2 | 4 | 10 | 11 |
| Ｄ | 3 | 3 | 3 | 7 | 3 | 5 | 7 | 3 | 7 | 13 |
| Ｅ | 10 | 1 | 5 | 2 | 10 | 6 | 1 | 2 | 9 | 5 |

* **系統硬體架構方塊圖（接線圖）：**



* **系統架構程式碼、測試資料程式碼與程式碼說明**

**截圖請善用win+shift+S**

|  |
| --- |
|  |
| **以上code註解也看得出來粗略地分成三個部分，主要以DFF作為切分點，並將\_設計成組合邏輯輸出，\_r表示經過一層序向邏輯的輸出，\_rr表示經過兩層序向邏輯的輸出。** |

* **架構圖：**

|  |
| --- |
|  |
| **上圖為quartus 19.1經過compile後產生的架構圖，為確認與架構圖相同，經過兩層DFF後OR出y的答案。** |

* **模擬結果與結果說明：**

**Testbench：**

|  |
| --- |
|  |
| **這是本人此題的testbench，配合pipe說明用到的第一筆輸入，且確認與題目所述的輸入相同。** |

**全部結果：**

|  |
| --- |
|  |
| **此圖為展示全部輸入跑出的結果。** |

**雙重結果確認：**

|  |
| --- |
|  |
| **上圖為一個本人依照此題撰寫的C++ code，從terminal輸出可看出與波形圖輸出一模一樣。** |

**說明pipe：**

|  |
| --- |
|  |
| **此圖加上了check欄，為證明有pipe，可發現在黃色線(表示得到and結果)一個clk後跑出第一個結果15，且同時算出第二筆輸入的and值4。**  **在初始化後，可看到於rst=1區間的posedge clk後，第一層pipe輸出全初始化為0，且rst=0後的第一個low level(#30)在testbench吃到第一筆輸入，經過兩個posedge clk後輸出第一筆y=15，也符合兩層DFF需要兩個clk才得知y的設計。** |

1. **Pipeline**

* **實驗說明：**

1. 完成下圖架構，Program\_Rom 已提供
2. 確保程式計數器 (pc) 正確計數，以及指令 (inst and inst\_r) 被正確提取。

* **系統硬體架構方塊圖（接線圖）：**
* 一張含有 螢幕擷取畫面, 文字, 多媒體軟體, 圖表 的圖片

  AI 產生的內容可能不正確。

**一張含有 螢幕擷取畫面, 多媒體軟體, 電路 的圖片

AI 產生的內容可能不正確。**

* **系統架構程式碼、測試資料程式碼與程式碼說明**

**截圖請善用win+shift+S**

* 1. 以下截圖按mycpu.sv中出現的module順序
     1. mycpu.sv

|  |
| --- |
|  |

* + 1. **Program\_rom.sv**

|  |
| --- |
|  |

* + 1. **INST\_DEC**

|  |
| --- |
|  |

* + 1. **Reg\_file**

|  |
| --- |
|  |

* + 1. **controller.sv**

|  |
| --- |
|  |
| **可以看到與上課demo的主要差別為，狀態s1的flush\_IFID\_ = 1;被註解，改取為default的flush\_IFID\_ = 0;。**  **這原因是觀察附的波形圖，在s1時inst\_r顯示不為13 (I\_NOP之值)，即在s1狀態未reset。而controller不能控制rst線，只能控制一同or的flush\_IFID\_，且rst此時為0，即只要flush\_IFID\_不為1就不會是13。** |

* **模擬結果與結果說明：**

|  |
| --- |
|  |
| **上圖為證明與助教給的範例圖相同(含轉換禁制、加上Divider、照例圖由上到排序波型、rst位置為 #40)。** |

* **架構圖：**

|  |
| --- |
|  |
| **就DFF來說，我起碼還看得出來是兩層Pipe，還透過自己命名的module，大致上可以確認結構跟例圖給的有87%像。** |

* **結論與心得：**

　　經過此次的實驗，我學會寫cpu中的fetch cycle，刻完了cpu大致上的架構(應該：Ｐ)，但從demo的情況~我學習狀況大概應該算是相當慘烈的。

　　但我只能說，牢我還是會做穿的，我大概也就只能每次慢慢的把進度追上，慢慢習慣寫硬體的步調，感謝助教跟朋友每次耐心的解決的紅色的線跟問題。