DD LAB5

一、地點：工學501A

二、準備時間：4/27 1924-1932

三、DEMO時間：4/27 1932-1940

四、評分方式

1. 利用 testbench於命令提示字元中顯示生日(20%)

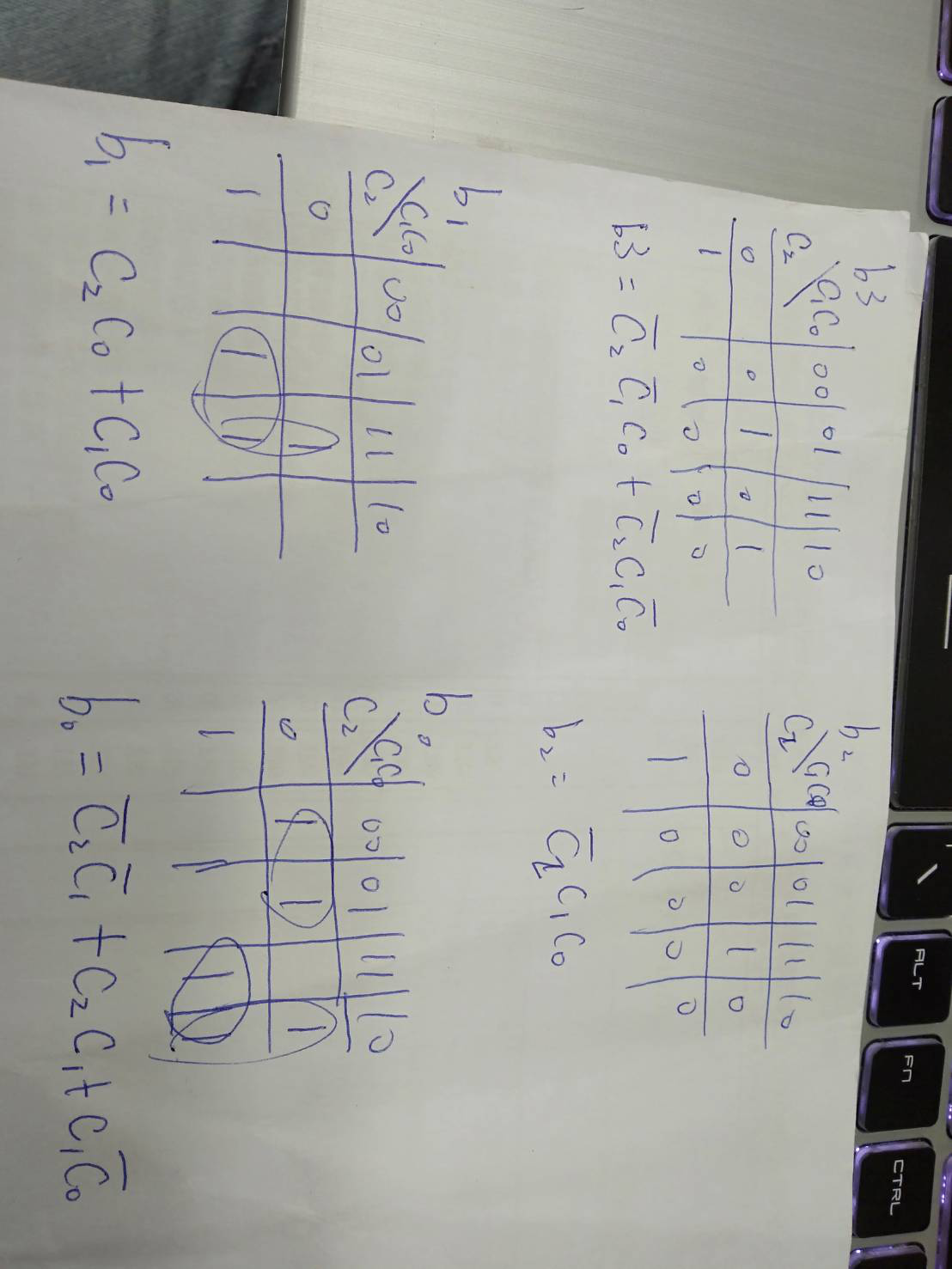
(1)路徑 C:\Users\User\Desktop\CO\2019\_COLAB2\iverilog\bin開啟cmd

(2)$iverilog -o test tb\_lab5.v

(3)$vvp test

(4)化簡過程

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cnt 2 | Cnt 1 | Cnt 0 | birth\_num3 | birth\_num2 | birth\_num1 | birth\_num0 | Num |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 9 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 9 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 6 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 3 |



化簡成SOP

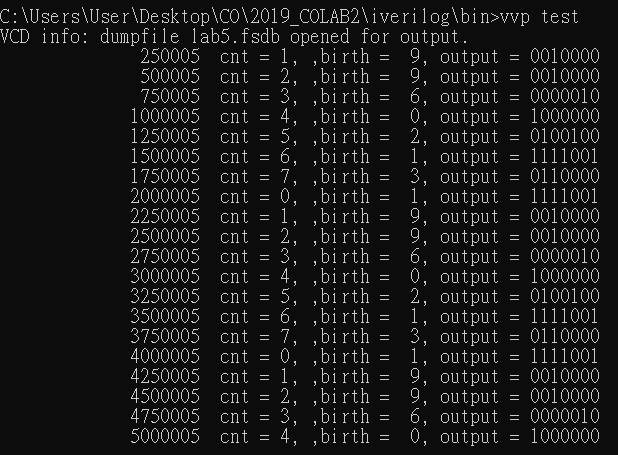
birth\_num[3] = (!cnt[2] & !cnt[1] & cnt[0]) | ( !cnt[2] & cnt[1] & !cnt[0] )

birth\_num[2] = (!cnt[2] & cnt[1] & cnt[0] )

birth\_num[1] = (cnt[2] & cnt[0]) | (cnt[1] & cnt[0])

birth\_num[0] = (!cnt[2] & !cnt[1]) | (cnt[2] & cnt[1]) | (cnt[1] & !cnt[0])

(5)結果

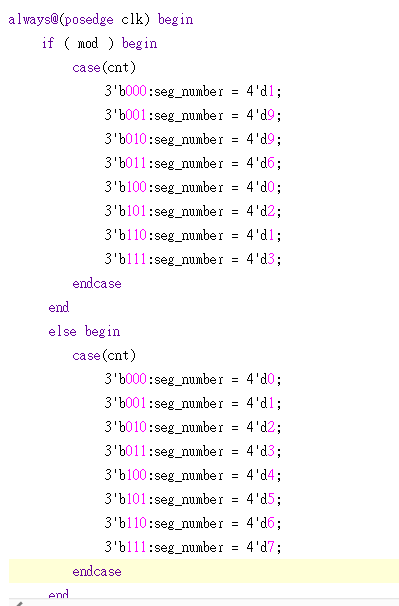


2.以sw14為開關，為0時最右邊的七段顯示器會依序顯示0~7，為1時則會依序顯示出自己的生

日 (60%)

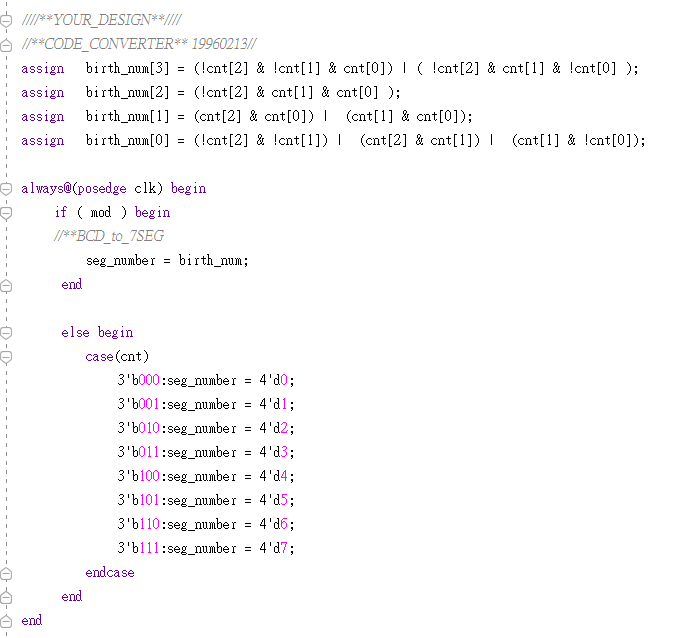
1. 須帶身分證
2. Sw 15為rst 用Sw14 on 顯示生日八碼 Sw14 off顯示0-7
3. Case版本設計：偵測到sw14 on (mod)就把cnt0-7對應到19960213，

若sw14 off (!mod)就把cnt0-7對應01234567



1. Gate版本設計：偵測到sw14 on (mod)就把算好的birth num給seg\_number

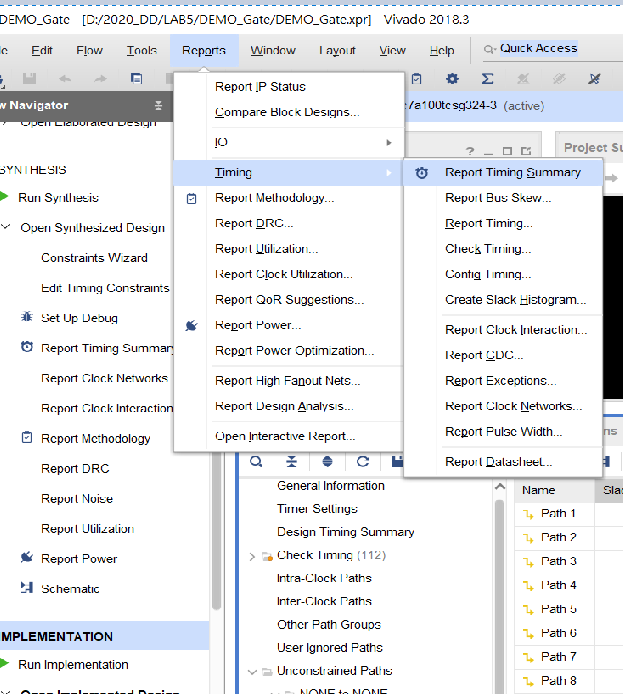
若sw14 off (!mod)就把cnt0-7對應01234567



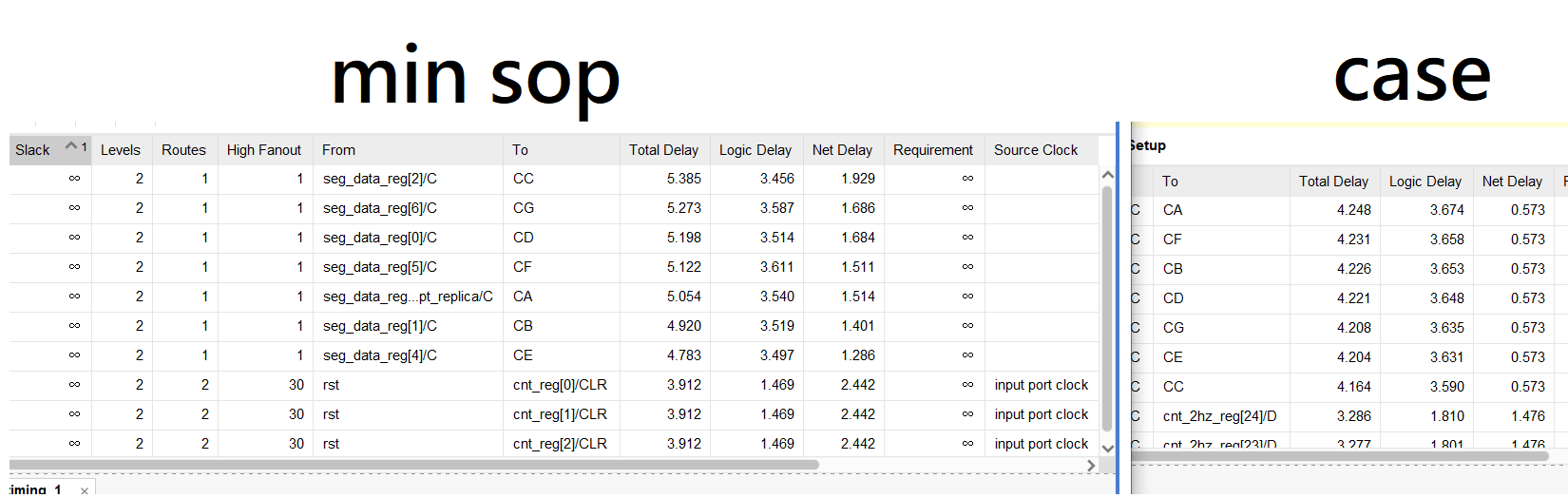
3.於回饋單上描述比較兩種方法在開發版上合成的差異 (20%)

(1)使用case的方法是流程控制結構並非像用邏輯閘一直計算去實現，相對硬體需求上會更加有效率跟省力，並且代碼更易讀，case比if else好，因為case沒有優先順序可平行運行，但是if else 有，會消耗掉大量組合邏輯

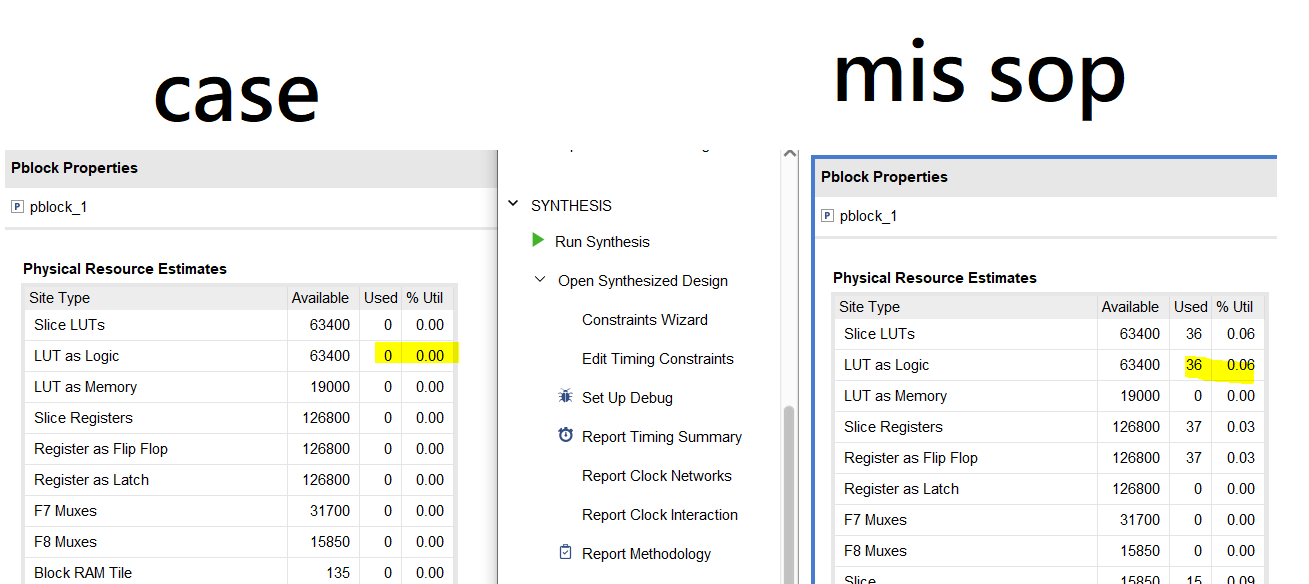
(i)timing report



Min sop delay 比case高



(ii)面積比較



4.問題

(1)回饋單的填寫20%是事後給嗎? 因為都是之後才寫回饋單? 是

(2)case和邏輯閘設計兩種方法都需要做嗎? 還是可以做一種就好? 一種會亮就好

(3)用sw14當作開關的意思是說用sw15 當作rst sw14當作mod去切換0~7或是生日8碼嗎?是

5.附錄：燒錄方法說明

