

Lab 1

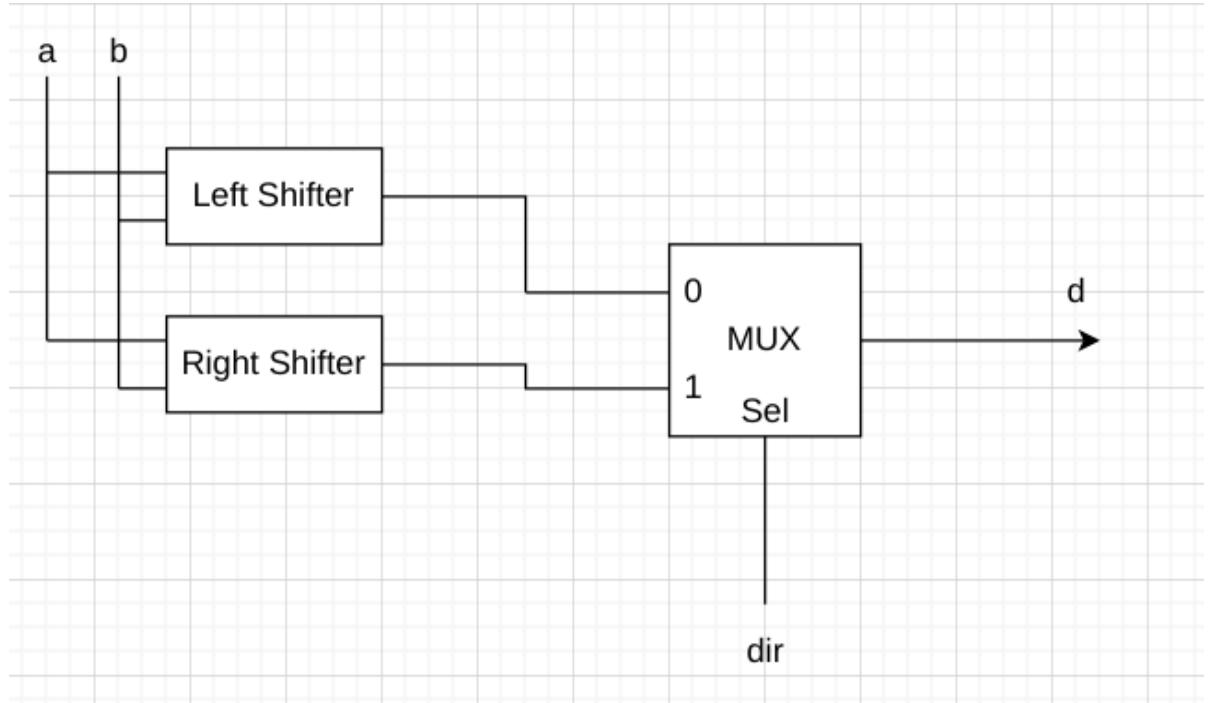
學號: 109062302

姓名: 郭品毅

1. 實作過程

- Lab 1_1

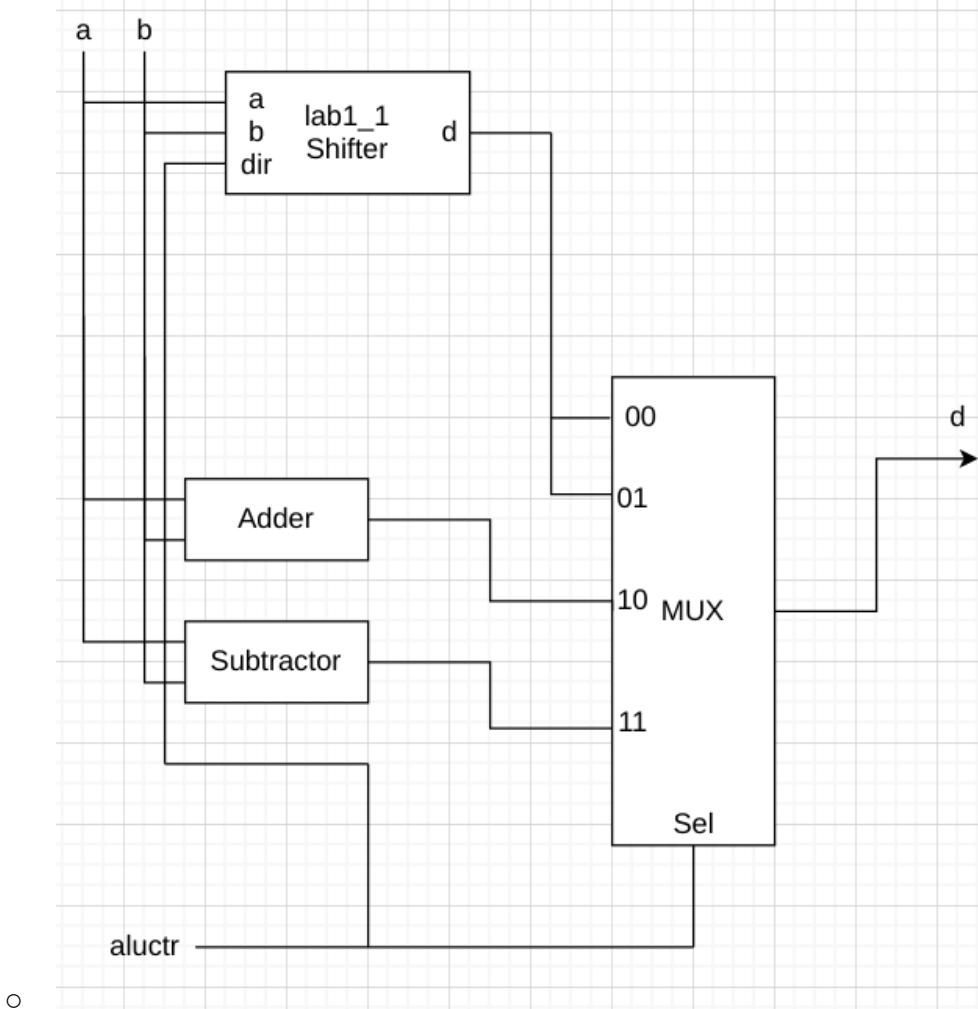
- 使用 assign, wire, ternary operator 或 always, register, if/else => 2x1 MUX
- 當 dir 為 0 時, 就會取 Left Shifter 的結果, 而輸出 $a \ll b$
- 當 dir 為 1 時, 就會取 Right Shifter 的結果, 而輸出 $a \gg b$



-

- Lab 1_2

- 使用 lab1_1 的 module, 因為 aluctr 的 LSB 與 dir 方向在 spec 上一致, 因此直接接入 lab1_1 當作 dir 的 input
- 其他的 case 也是由 MUX 來切換:
- 當 aluctr 為 00 時, 就會取 lab1_1 module 的輸出, 而輸出 $a \ll b$
- 當 aluctr 為 01 時, 就會取 lab1_1 module 的輸出, 而輸出 $a \gg b$
- 當 aluctr 為 10 時, 就會取 Adder 的輸出, 而輸出 $a + b$
- 當 aluctr 為 11 時, 就會取 Subtractor 的輸出, 而輸出 $a - b$



2. 學到的東西與遇到的困難

- 學會製作 Arch Linux 的軟體包來安裝 Vivado
- 學會開啟 Vivado
- 學會 initial block 和 always block 以及他們的差異
- 學會使用 assign 語法
- 學會使用 if/else 語法
- 學會製作 Combinational Logic
- 學會接入其他的 module
- 學會執行 Behavioral Simulation 以及查看波形圖
- CT Verilog Series 只看到了第 03 集就開始寫 lab1, 結果踩了一堆坑
 - 只會 assign 和 initial block
 - lab1_1 一開始還寫出了這種東西: `assign d = (dir == 0) && (a << b) || (dir == 1) && (a >> b)`
■ JavaScript 中毒 QQ
 - 發現 Ternary Operator 可以用就開始亂寫, 然後把 lab1_1 和 lab1_2 都做出來了 XD
 - 在 initial block 裡面寫 if / else, 結果輸出都是 X
■ 之後才知道要用 always block

- 透過嘗試發現了這些性質
 - assign 只能用在 wire
 - 使用其他的 module, 輸出只能接 wire
 - initial/always block 裡面 = 的左式只能放 reg
 - 之後看影片才發現影片裡面都有講了
 - 學會先把課程看完再開始實作
 - Vivado 好像沒有 Dark Theme 的樣子...
3. 想對老師或助教說的話
- 希望可以出更多有趣一點的題目～
 - 希望可以教更多 debug 的方法～
 - 希望我可以想出很有趣的笑話～