Lab 1 report

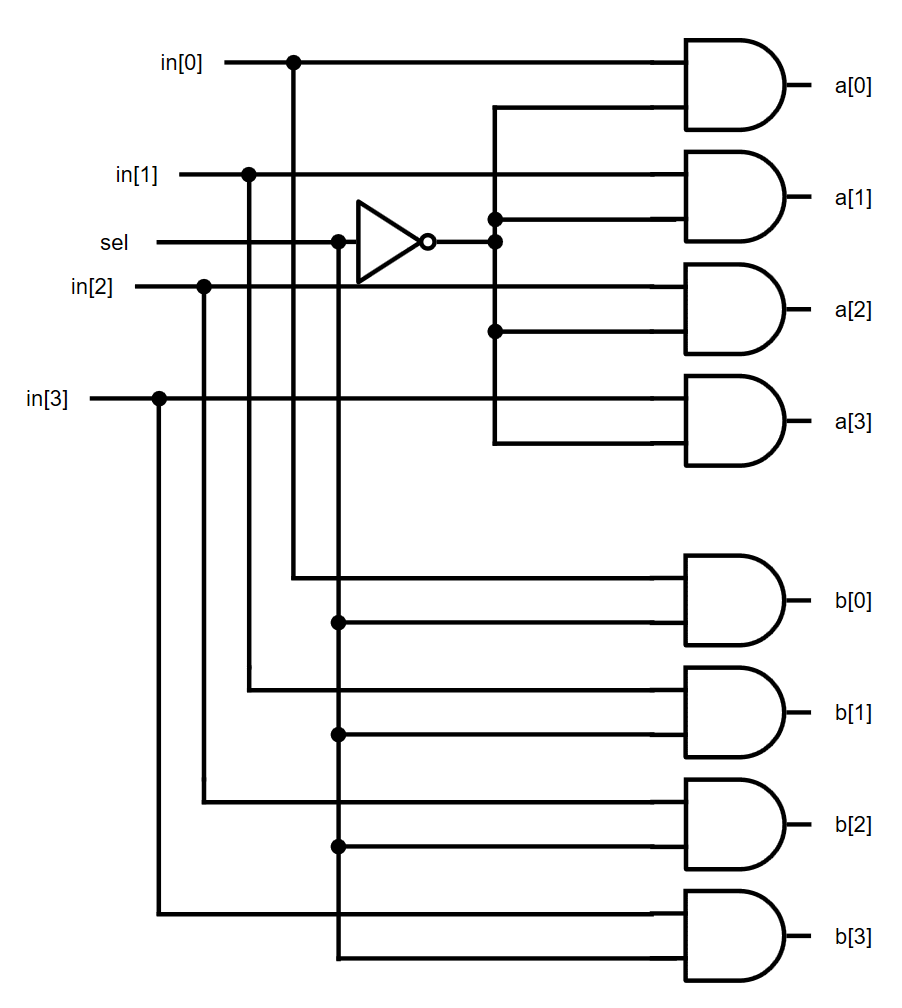
組員：

110062221 李品萱

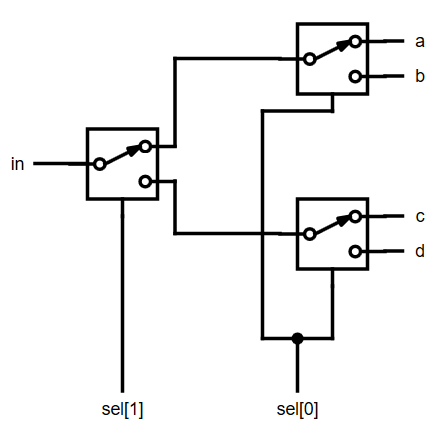
110062213 唐翊雯

1. **(Gate-level) 4-bit 1-to-4 de-multiplexer (DMUX)**

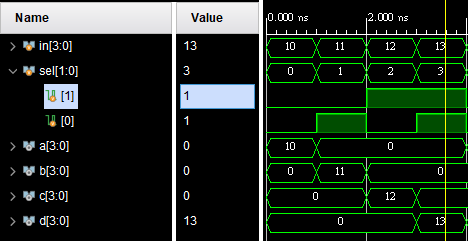
這題要用三個1-to-2DMUX接出1-to-4DMUX，所以我們先做出1-to-2 DMUX的部分，一個2-to-1 DMUX由一個not gate與8個and gate組成，gate-level circuit如下。



接著使用三個 1-to-2 DMUX組成一個1-to-4 DMUX，gate-level circuit如下，這邊我們會先判斷sel[1]是1或0，再判斷sel[0]對應到的output位置，也就是說，第一個DMUX會接出兩條線，然後分別接到兩個DMUX決定最後選到哪個訊號，這邊也可以看出DMUX的功能與MUX相反。

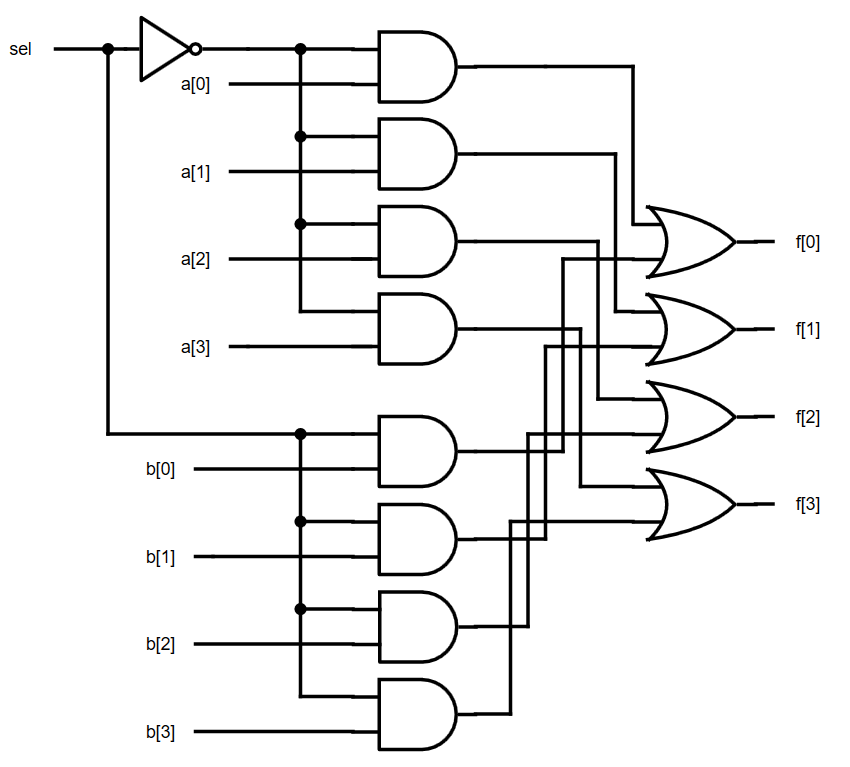


在simulation的結果中，我們也測試了每個sel，sel為00、01、10、11時分別會將output送到a、b、c、d，而未被選到的位置則會輸出0。

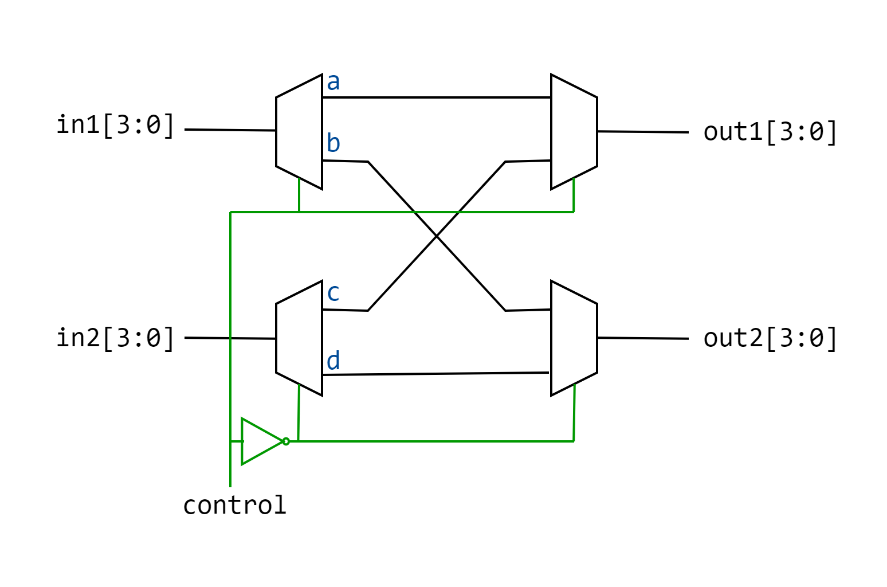
****

1. **(Gate-level) 4-bit simple crossbar switch with MUX/DMUX**

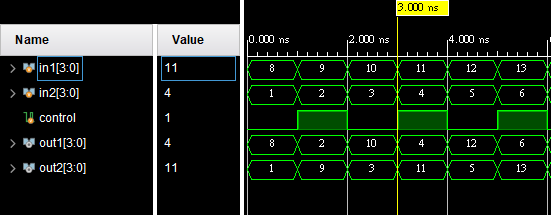
整個circuit由兩個1-to-2 DMUX與兩個2-to-1 MUX（皆為4 bit）組成。DMUX的gate-level circuit如前一題所示，而MUX的gate-level circuit如下：



在整個crossbar中，我們先用DMUX他們會送到哪一條線，再用MUX進行多選一的操作。舉例來說，當control是1時，in1會由b送到下圖中右下方的MUX，對於此MUX來說control是0，因此它會選擇wire b進來的訊號；而對於in2接到的DMUX來說control是0，因此會由wire c接出去，此時對於右上方的MUX control是1，c會接到out1。同理可知，當control為0時，in1會接到out1，in2會接到out2。

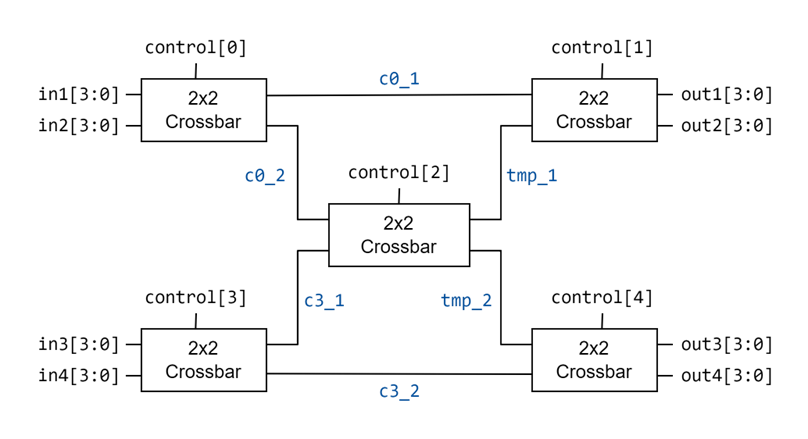


在testbench的部分，我們讓control在0、1交替變換，也可看出前面藉由電路圖得到的結果。

****

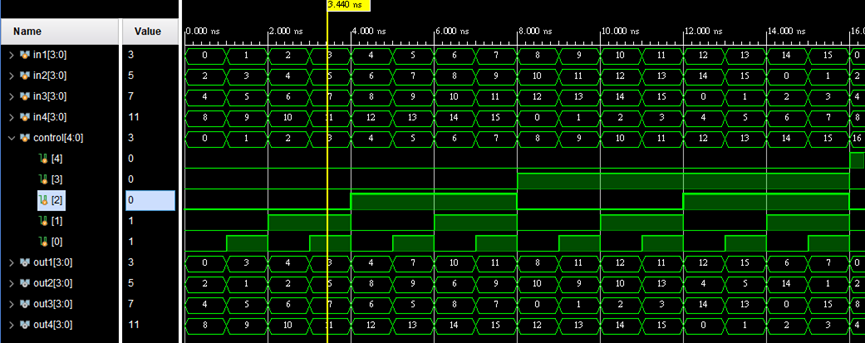
1. **(Gate-level) 4-bit 4x4crossbar with simple crossbar switch**

這題會用到前面實做過的2x2Crossbar，首先我們將input送進2x2Crossbar並用control訊號做select的動作，而後，c0\_2、c3\_1這兩條線接到的訊號會再送進2x2Crossbar做一次select，再藉由tmp\_1、tmp\_2分別送進兩個2x2Crossbar並select後送到對應的output。



從下面simulation的結果及上面的電路圖中我們可以觀察到，當control[2]為0時，in1、in2必不會對應到out3或out4，當control[2]為1時，in1、in2必不會同時位於out1及out2或out3及out4，同理，in3、in4也不會同時位於out1及out2或out3及out4，所以它不可能出現的組合如下：

* [(in1, out3), (in2, out4), (in3, out1), (in4, out2)]
* [(in1, out3), (in2, out4), (in3, out2), (in4, out1)])
* [(in1, out4), (in2, out3), (in3, out1), (in4, out2)])
* [(in1, out4), (in2, out3), (in3, out2), (in4, out1)])



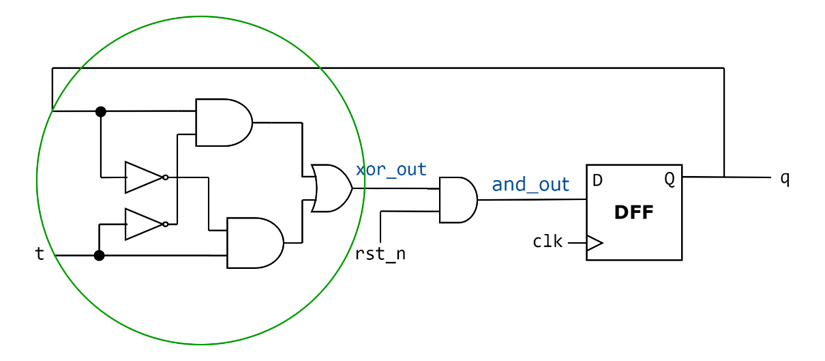
因為這題的Testbench主要目的是要確認control對各個input做怎樣的選擇，所以這部分我們用枚舉的方式跑過一次所有可能的control，確認不同的control訊號組合下出現的結果是正確的。

一張含有 文字 的圖片

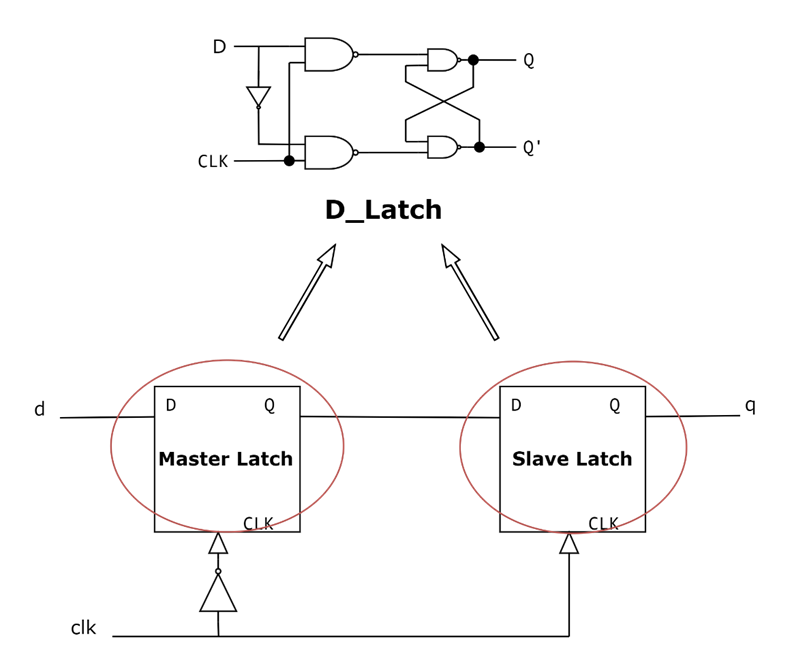
自動產生的描述

1. **(Gate-level) 1-bit toggle flip flop (TFF)**

在這題中，我們將t跟q接上xor gate再與rst\_n做and，再接上前面實做過的DFF，而xor gate的gate level circuit如下圖綠色圈起處所示，而此處TFF會將output Q再送進一開始的xor gate，這會使現在的q值對下一個clk時的q值產生影響，這部份我們也可以在後面的simulation結果中看到。



其中，由下圖可知D flip flop是由Master Latch、Slave Latch及clock組成，clock在low時會將訊號從master latch送至slave latch，clock high時再由slave latch送出，即為positive edge的DFF。



經過simulation後，我們可以從waveform中觀察到，當clk為high且rst\_n為1時，q與t做xor的值就會被送出，維持一個clock cycle；而當clk為high但rst\_n為0時，q值會為0，因為此時相當於把0送進DFF，其結果為0。此外，在waveform中，我們可以看到q一開始有紅色的區塊，此為電路尚未給值時呈現的狀態。

一張含有 文字, 計分板 的圖片

自動產生的描述

在testbench的部分，我們參考了basic lab DFF的testbench，對應不同的clock改變rst\_n及t的值，這邊也有刻意使rst\_n的變化頻率與t不同以方便觀察q的結果，雖然看起來頗為冗長，但至少有達到我們確認其特徵的目的。

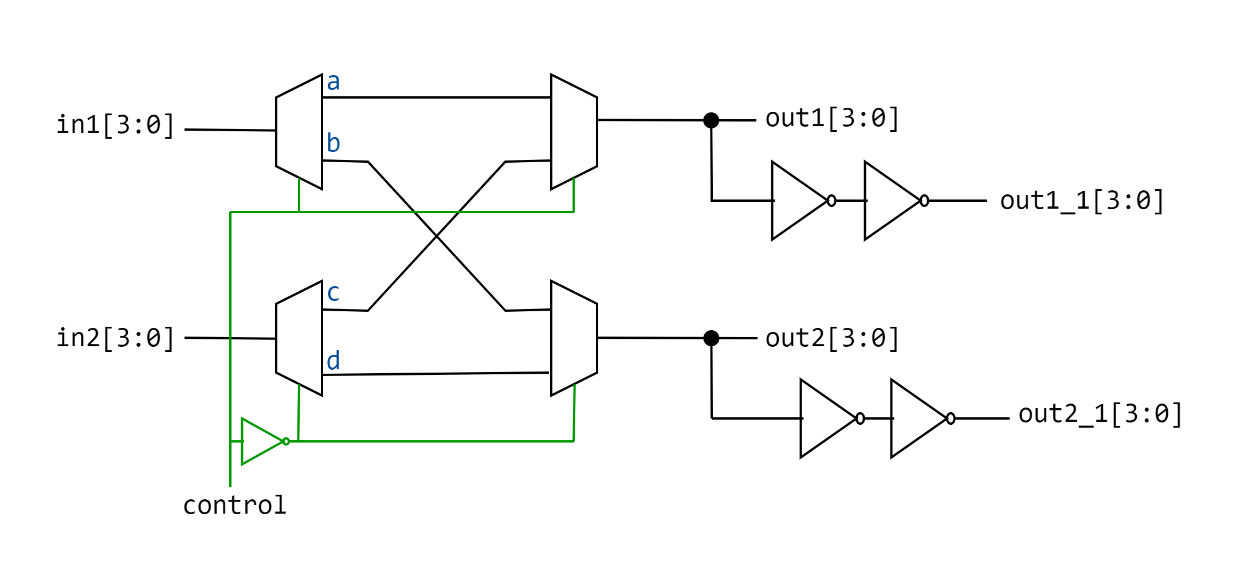
**一張含有 文字 的圖片

自動產生的描述**

1. **FPGA demonstration: (Gate-level) 4-bit simple crossbar switch with MUX/DMUX**

這一題Code的部分只有基於4-bit simple crossbar的code做一些修改：由於題目規定一個output要對應到兩個LED，這部份我們兩個人對fan-out有不同的做法，如下：

我們的第一種作法是在原本的out1及out2各自再接出兩個not gate，並再開out1\_1、out2\_1這兩個output接原本的output做完兩次not的結果，這個方法是參考講義上fan-out那一段的作法。

****

在寫fan-out的過程中，我們原本的寫法是下圖中註解的部分，但後來在網路上剛好看到語法的資料，發現一樣的事可以精簡成一行，如下圖。

一張含有 文字 的圖片

自動產生的描述

第二種方法是我們另外開了兩個4 bit的output變數，並將其值分別設為原先的output 1與output 2，再分別對應到LED燈上。

1. **Summery**
2. **Contributions**