Lab 1: Gate-Level Verilog

Group 21: 陳克盈（112062205）、蔡明妡（112062224）

Table of Contents

1 NAND Gate to all other gates

此章節利用 NAND 所組成的所有邏輯閘，都將用來作為後續所有題目的邏輯閘使用。
在表示上為求可讀性，將會使用一般的邏輯閘符號來表示。

2 Full Adder vs. Half Adder

Half Adder 雖然能夠算出總和以及進位值，但由於缺少了 cin 的輸入，
導致他只能處理單一位元的加法，這也是為什麼他被稱作是 Half Adder。

3 Q1: 8-bit ripple carry adder

根據題目所求，建立八個 Basic Q3 撰寫的 Full Adder，並將他們依據順序將輸入輸出串接在一起，
便完成了 8-bit ripple carry adder。

4 Q2: Decode and execute

4.1 Universal gate

Universal gate 由 $a\&!b$ 組成，不難發現，當我們將 a 接上 True 信號，那麼這個 Gate 就會變成一個 NOT Gate，再利用這個 NOT Gate 接到 b 上，就變成了一個 AND Gate。最後，將這個 NOT Gate 接到 AND Gate 的輸出上，就完成了一個 NAND Gate。

剩下的所有基本邏輯閘都可以透過這個生成出來的 NAND Gate 組合出來。由於組合方法已經在章節**??**中提到，因此這裡就不再贅述。

4.2 Executor

4.2.1 SUB

首先，$rs - rt$ 可以被轉換為 $rs + (-rt)$，根據二補數的規則，可以再轉換為 $rs+ \sim rt + 1$。再搭配章節**??**實現的加法器，改變成 $4 - bitAdder$ 後，將 (a, b, cin) 設為 $rs, \sim rt, 1$，便是 SUB 的實現。

4.2.2 ADD

ADD 的實現同樣基於章節**??**實現的加法器，將 (a, b, cin) 設為 $rs, rt, 0$，便是 ADD 的實現。

4.2.3 BITWISE OR

利用 OR Gate 將每個位元分別運算。這裡放圖就好

4.2.4 BITWISE AND

利用 AND Gate 將每個位元分別運算。

這裡也是放圖就好

4.2.5 RIGHT SHIFT

由於一個 4-bit 的數字右移一位後，$out_0 \sim out_3$ 分別會對應到 in_1, in_2, in_3, in_0 因此只需要透過與 1 做 AND 能夠實現 assign 的特性，將輸出對應到正確的輸入即可。

4.2.6 LEFT SHIFT

左移實作方式與右移相同，唯一的區別在於 $out_0 \sim out_3$ 分別對應到的是 in_3, in_0, in_1, in_2。

4.3 Decoder

首先將 Executor 生成的結果依 OP_Code 順序導至 $0 \sim 7$，由於指令只有八種，因此我們可以使用七個 2-to-1 mux，並分成三個階段來實現 Decoder。

4.3.1 COMPARE LT

由於比較的結果決定於由高位數來第一個不同的位元，因此一個 4-bit 的比較小於將基於以下步驟：

(1) 比較 $rs[3]$ 是否 $< rt[3]$

(2) 若上個步驟為 False 且 $rs[3] = rt[3]$，則比較 $rs[2]$ 是否 $< rt[2]$

(3) 重複上述步驟，直到找到第一個不同的位元

這幾個步驟可以寫成一個表達式：$out = (rs[3] < rt[3]) \mid (rs[3] = rt[3] \& rs[2] < rt[2]) \mid (rs[3] = rt[3] \& rs[2] = rt[2] \& rs[1] < rt[1]) \mid (rs[3] = rt[3] \& rs[2] = rt[2] \& rs[1] = rt[1] \& rs[0] < rt[0])$

根據以上式子實作便可得到比較的結果，最後再加上題目規定的 $1010_{(2)}$ 即可。

(1) 第一階段，將 $sel[0]$ 作為 MUX 的 select bit，並依照 OP_Code 分成 $(0,1),(2,3),(4,5),(6,7)$，各自輸入到一個 MUX 中，便可將八種指令篩選成四種

(2) 第二階段，與第一階段相同，將 $sel[1]$ 作為 MUX 的 select bit，便可以將四種指令再篩選成兩種。

(3) 第三階段，將 $sel[2]$ 作為 MUX 的 select bit，便可以得到我們所需要的輸出。

4.3.2 COMPARE EQ

兩個 bit 要相等只有兩種可能：$0,0$ 或是 $1,1$，因此 $a = b$ 可以利用 $(!a \& !b) \mid (a \& b)$ 來表示。

分別對四個位元做以上操作，並確認是否都為 True，最終加上題目要求的 $1110_{(2)}$ 即為答案。

5 Q3: 4-bit carry-lookahead adder

CLA 的運算過程可以分為：PG Generator, CLA Generator, Adder 三個部分：

5.1 PG Generator

首先定義