Laboratory 9
Verilog Application – Stack Calculator

Department of Electrical Engineering
National Cheng Kung University

Outline

- 後序式計算機簡介
- 實作一：算術邏輯單元
- 實作二：Stack計算機

Postfix Notation

- 後續式運算(Postfix Notation)是一種數學運算表達式，將運算子寫在操作數之後。主要優點是消除了括號的需求，同時使運算順序更加明確。後序式運算也常用於計算機科學領域和某些計算器上，因為它的處理方式相對簡單且容易實現。

Infix expression	Postfix expression
(A + B) * (C - D)	AB + CD - *

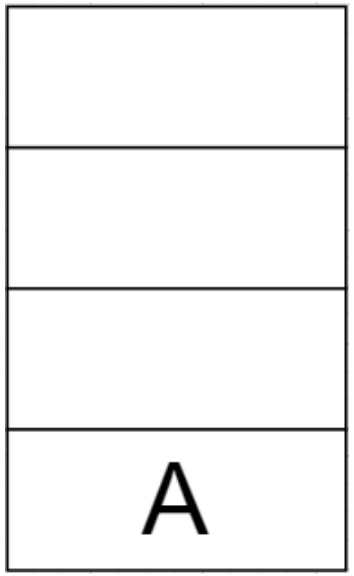
Stack

- 為了計算後序式運算，我們需要使用一種稱為後序式運算法的方法。該算法使用栈（stack）結構來存儲運算數和運算結果，並根據遇到的運算子進行計算。
  1. 創建一個空栈（stack）來存儲操作數和中間運算結果。
  2. 從左到右依次讀取後序式運算表達式中的元素。
  3. 如果讀取的元素是操作數（數字），將其壓入栈中。
  4. 如果讀取的元素是運算子，則從栈中彈出相應數量的操作數（根據運算子需要的操作數量），進行運算，並將運算結果壓入栈中。
  5. 重複步驟 3 和 4，直到遍歷完整個後序式運算表達式。
  - 最終，栈中只會剩下一個元素，這就是計算結果。

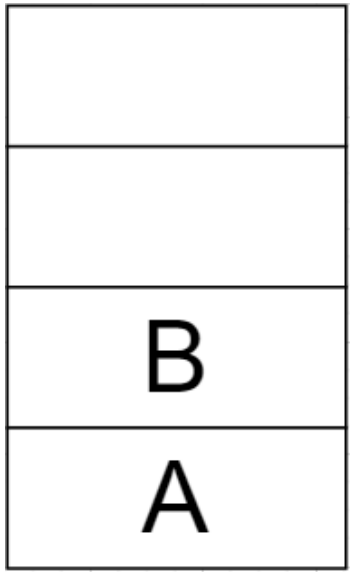
Postfix Calculation

- Postfix notation: AB+

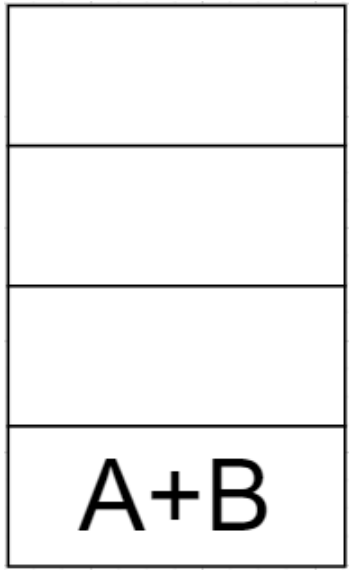
PUSH A



PUSH B

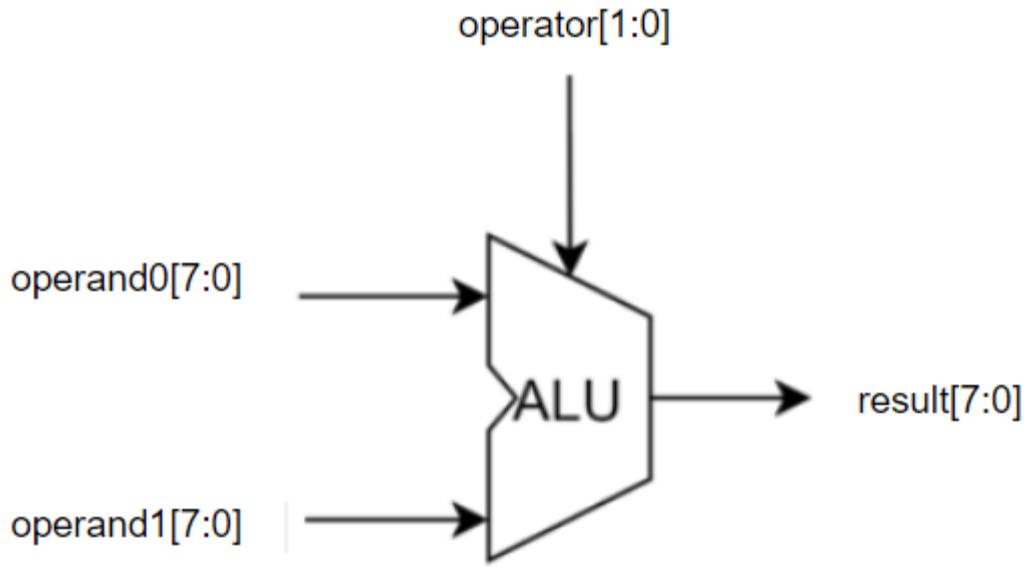


POP A
POP B
PUSH A+B



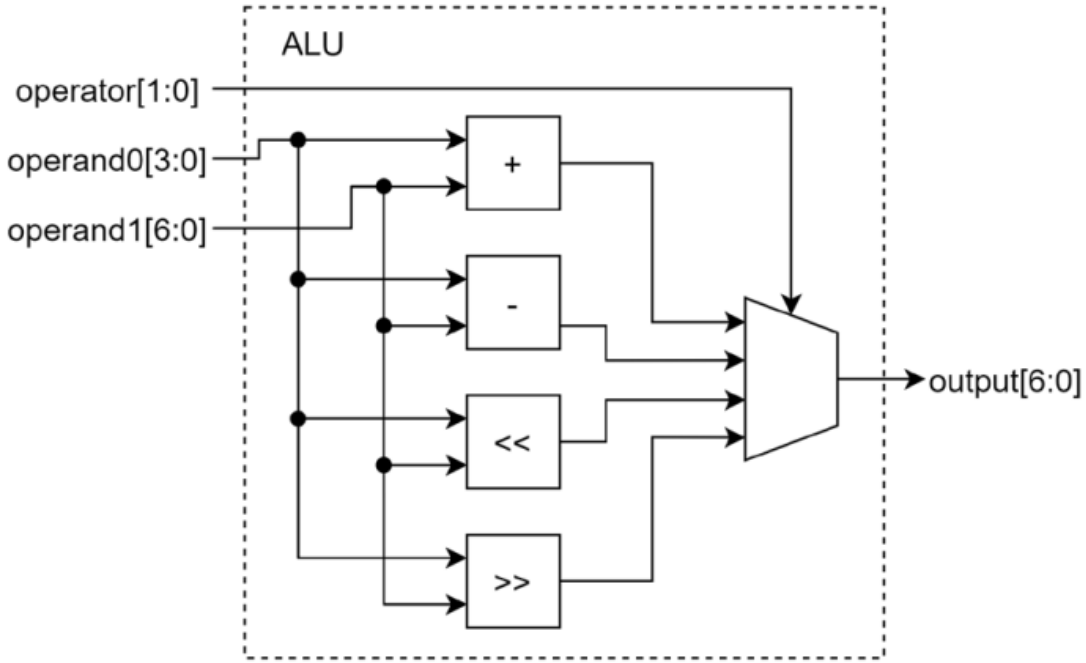
算術邏輯單元

- 輸入兩個運算元以及一個運算子以得到相應的運算結果
  - 其中 operand0 和operand1 皆為 8-bit 輸入
  - 運算子 operator 為 2-bit 輸入，選擇加(0)、減(1)、右移(2)、左移(3)
  - 其中在減法時以 operand0 – operand1 ；右/左移時以operand0 >> operand1 或是 operand0 << operand1 實現。



實作一

- 修改在 02_ALU 資料夾內的 ALU.v ，有一個現成的 ALU module
- 所有運算元都是無號數
- 測試時的結果一定會在 0~99 中間
- Hint: 因為電路無法停止的特性，實現時通常會讓所有結果都運算完畢，再使用一個多工器將正確的結果選出來

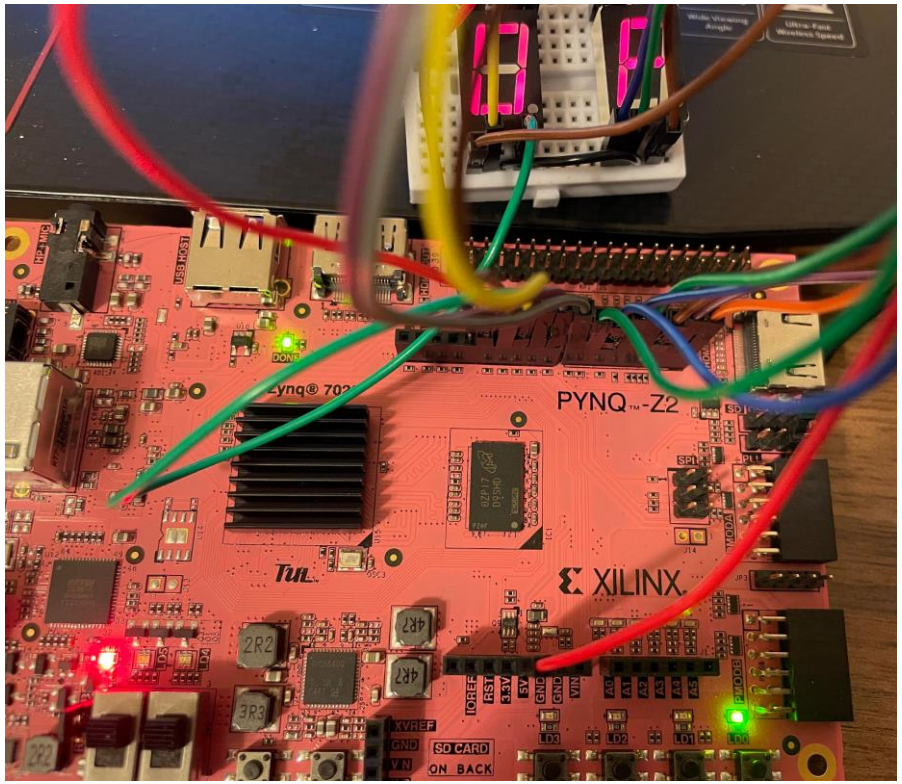


計算機介紹

- 本次實驗實作的計算機共有三種模式NUM mode、OP mode和Result mode。在每一種模式下的按鈕會有不同的功能，輸出的意義也有所不同。

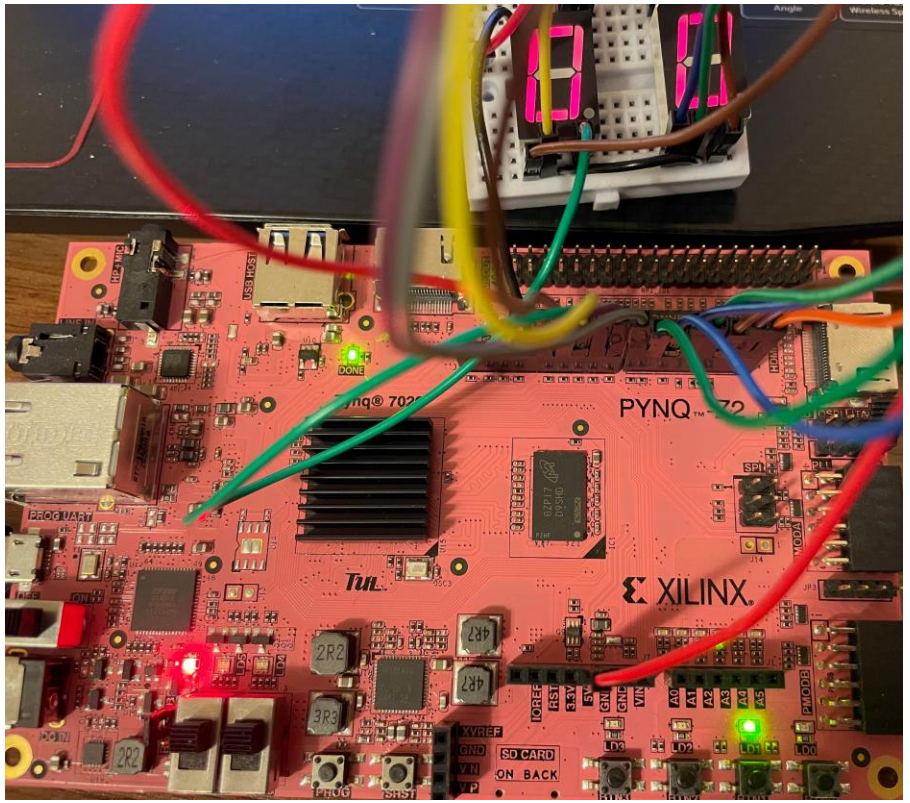
Num Mode

- Led[0]亮起表示目前為Num mode
- 七段顯示器顯示目前即將輸入的數字大小(不超過15)
- Btn[0]按下後會將目前的數字推入stack中並進到輸入下一個數字的Num mode
- Btn[1]按下後會將目前的數字推入stack中並進到輸入下一個數字的OP mode
- Btn[2]按下後會將目前的數字加一，並在超過15時overflow回到0



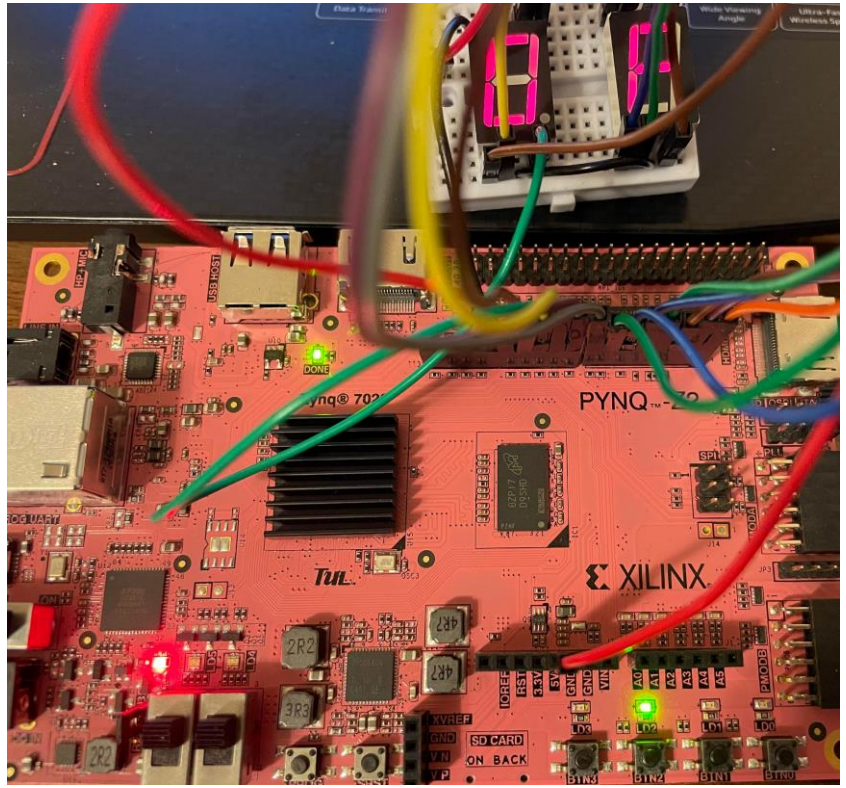
OP Mode

- Led[1]亮起表示目前為OP mode
- 七段顯示器顯示目前輸入的operator(0~3)
- Btn[0]按下後會將stack最上方的兩個數字彈出，並做完相對應的運算後存回stack上方並進入Result mode
- Btn[2]按下後會將目前的數字加一，並在超過3時overflow回到0

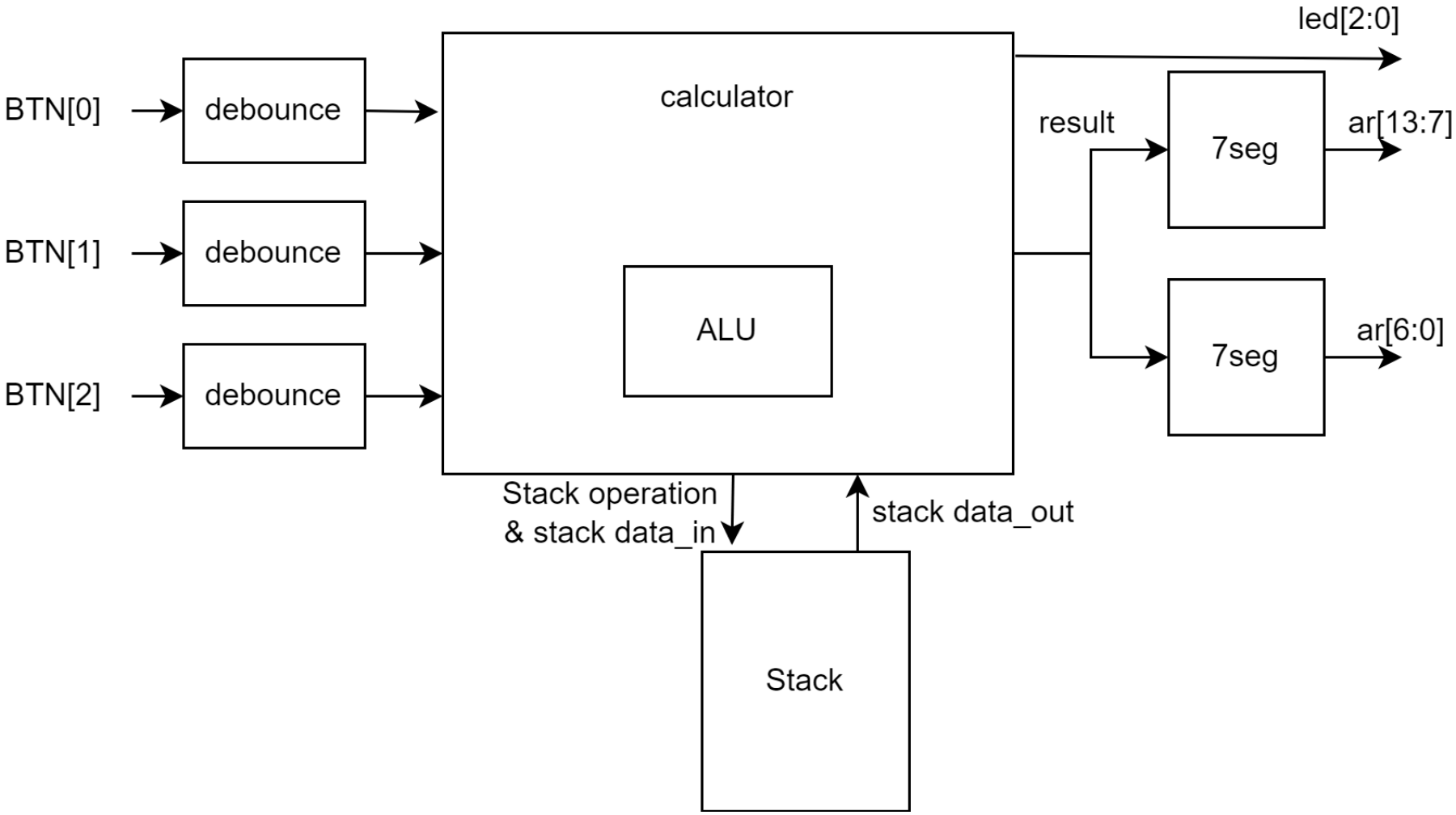


Result Mode

- Led[2]亮起表示目前為Result mode
- 七段顯示器顯示剛才計算出的數字大小(0~0xff)
- Btn[0]按下後會到輸入下一個數字的Num mode
- Btn[1]按下後會到OP mode



電路架構圖


led[2:0]
BTN[0]
debounce
BTN[1]
debounce
BTN[2]
debounce
calculator
ALU
result
7seg
ar[13:7]
7seg
ar[6:0]
Stack operation
& stack data_in
stack data_out
Stack

實作二

- 將實作一完成的ALU.v複製到實作二中
- 修改實作二中的calculator.v
- btn[3]當作序向電路的非同步reset（active-high）

實作二

- Stack的operation

operation	I	2	Operations. Idle = 2'b00, Push = 2'b01, Pop = 2'b10, Clear = 2'b11

課堂檢查與結報繳交內容

- 實作一
  - ALU的四種運算皆無問題
  - 結報挑3種不同輸入拍照
- 實作二
  - 任意輸入一串符合postfix notation規則的操作結果都會正確
  - 在結報中說明各種mode之間是如何進行切換，還有要在什麼條件下對stack進行操作