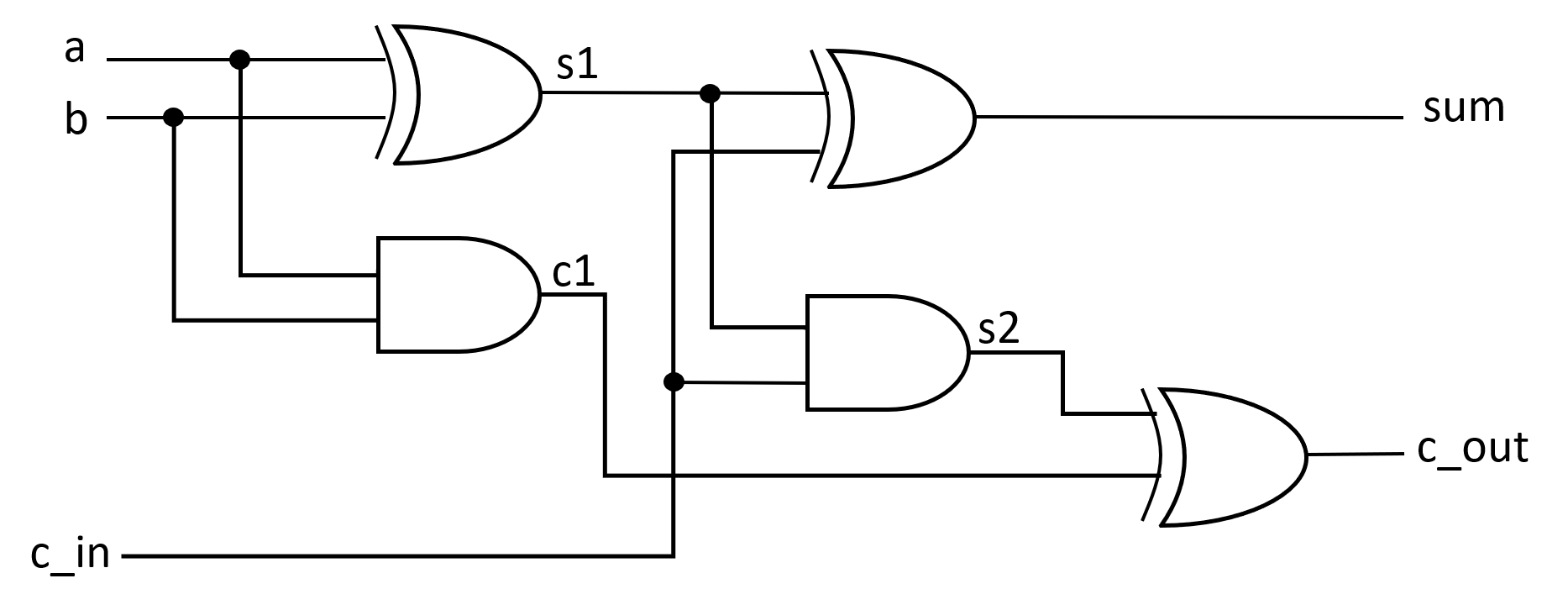
計算機組織Midterm Project

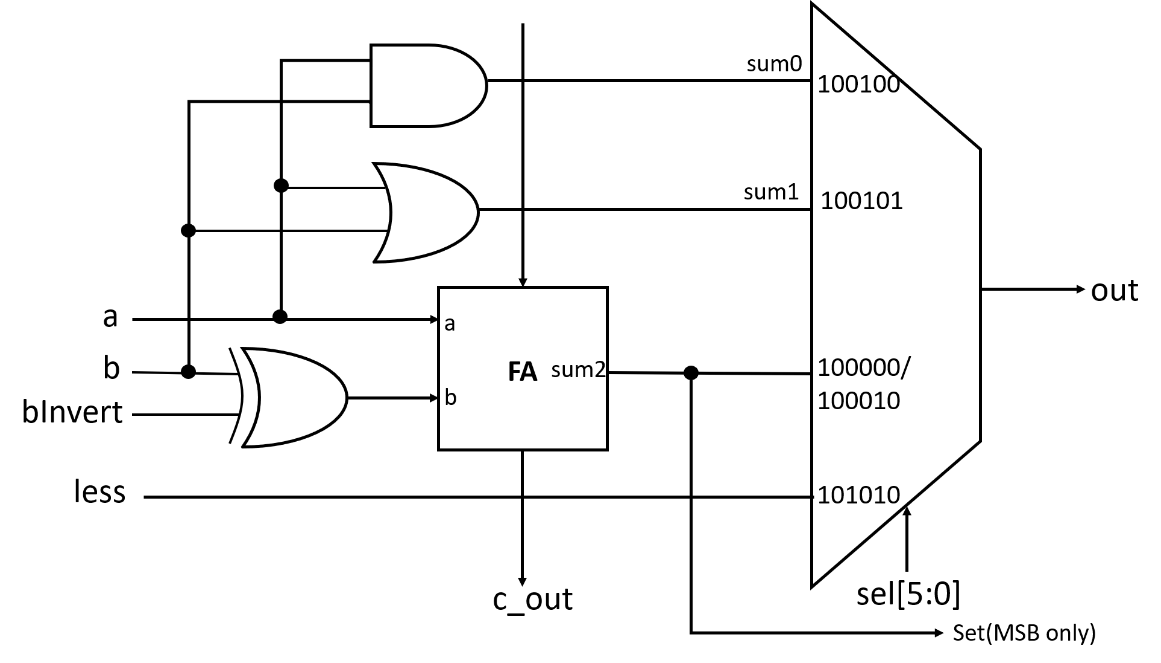
109學年度第2學期

1. Datapath與詳細架構圖

* 1位元全加器



* 1位元alu

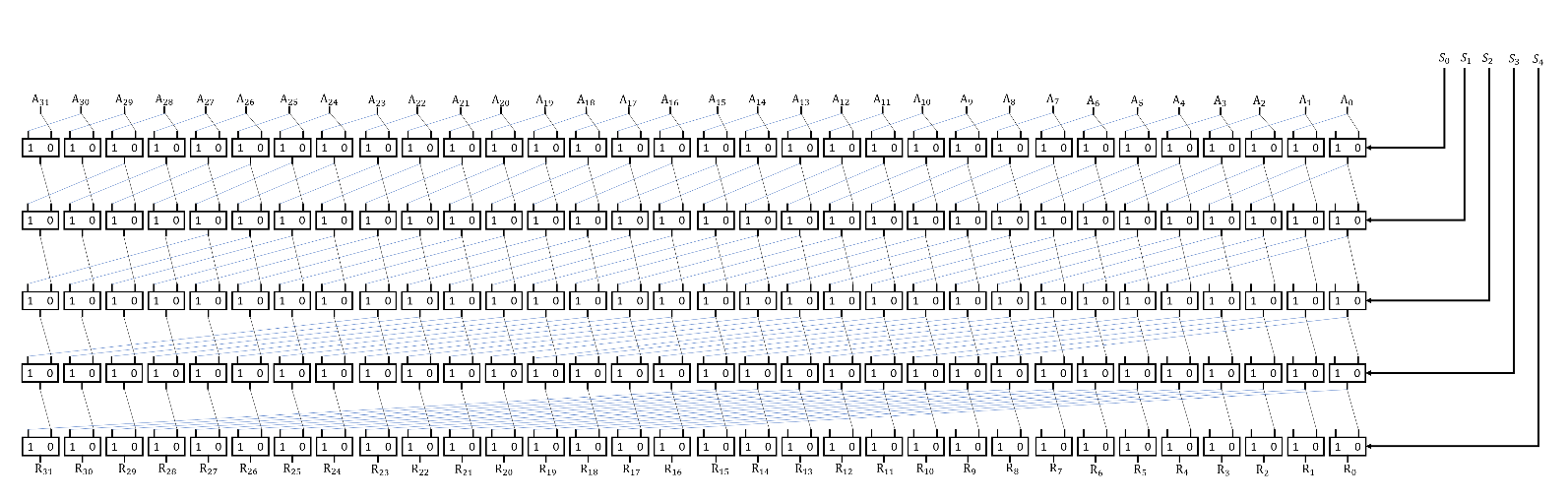


* 32位元alu

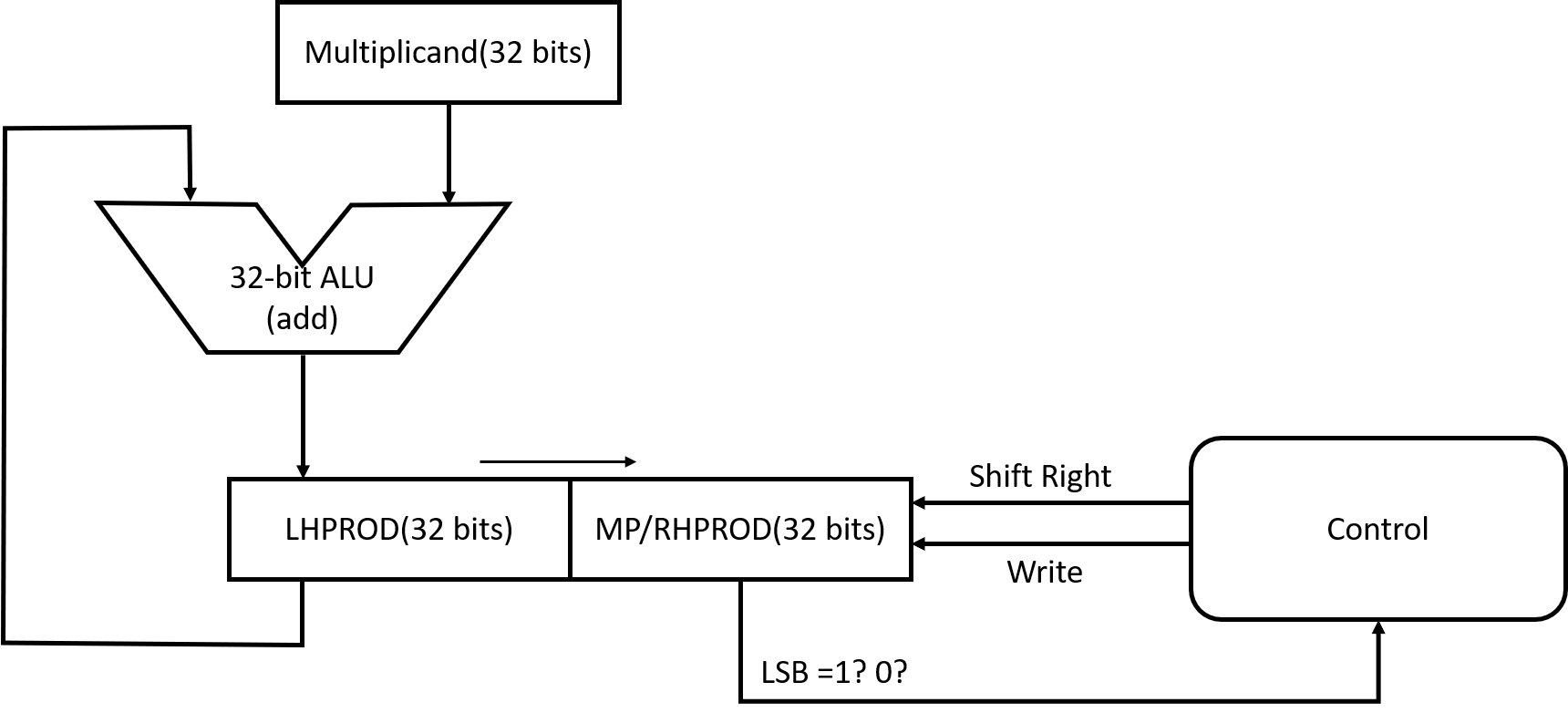
一張含有 文字 的圖片

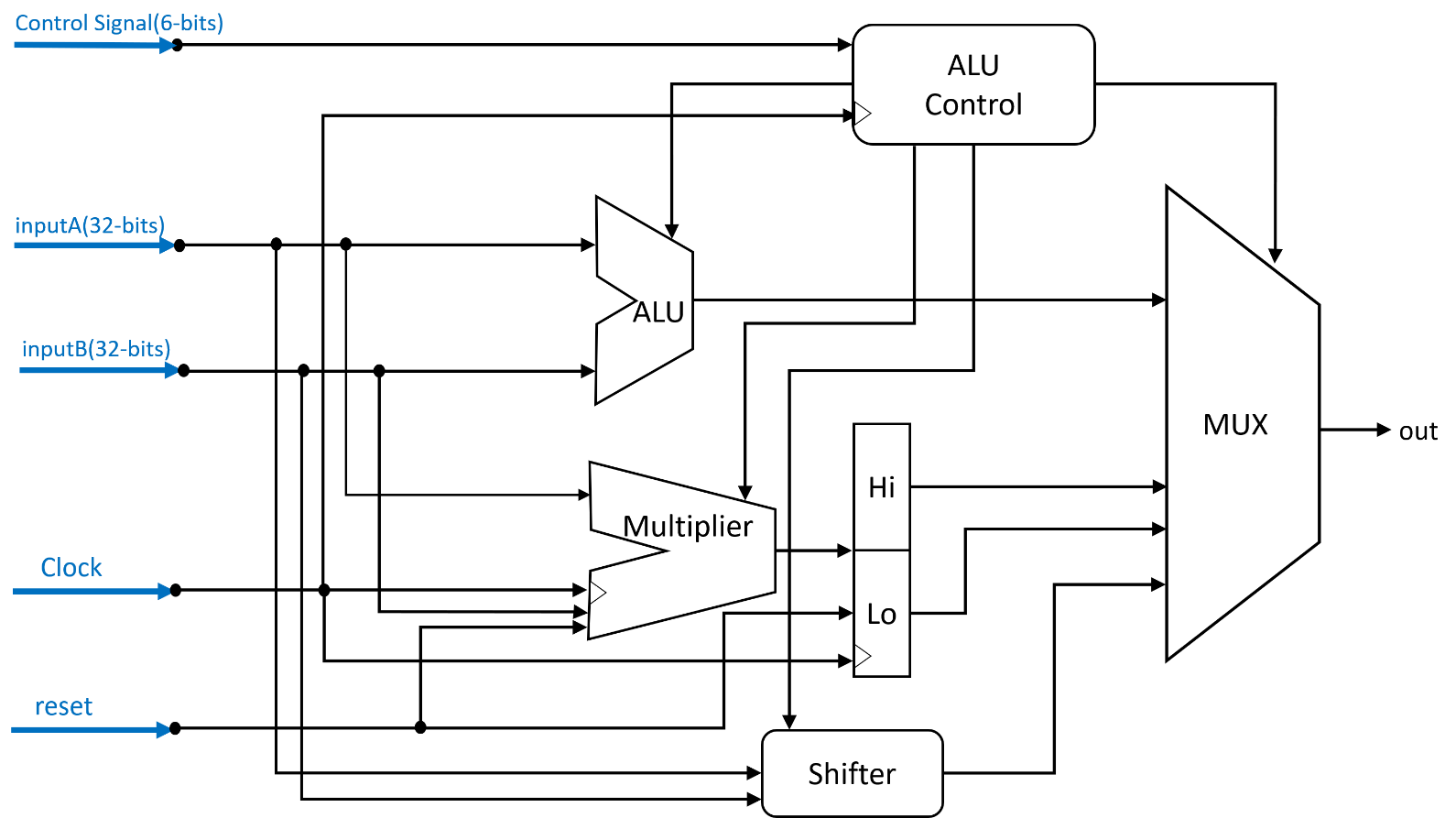
自動產生的描述

* 移位器



* 乘法器





1. 設計重點說明
2. ALU:

此為計算ADD、OR、ADD/SUB、SLT的結果。首先先設計一個1bit的alu，並串接(ripple carry)32個alu1bit，即為一個32bit的Alu。1bit的alu中有1個and的邏輯閘、1個or的邏輯閘、1個加上xor邏輯閘的全加器、以及一個less的輸入訊號(此為slt的計算結果)，以上全部的運算都會計算一次，再通過一個多工器去選擇傳入訊號所需要的答案。And與or的計算結果是直接通過邏輯閘做運算，而加上xor邏輯閘的全加器則為在b訊號的輸入端加上一個xor邏輯閘，此為判斷加減的依據，並將control傳入的訊號一個傳入xor邏輯閘一個傳入carry in的訊號端，用通過xor邏輯閘後的結果與a訊號做全加器的運算，計算出加減法的結果，最後slt的答案輸出則為第32個alu(1bit)的carry out輸出的訊號回接給第1個alu(1bit)的less輸入訊號，若減法溢位，則sign bit為1，所以slt的輸出結果為00000000000000000000000000000001，若否則全為0。

1. Multiplier:

利用Sequential Logic，並且用clock進行訊號同步。首先會判斷reset訊號。若reset訊號為1的話，進行reset動作，將dataOut左32bit歸零後，右32bit放入乘數。若訊號為MULTU(25)的話進行乘法動作，判斷dataOut的第一個bit，也就是乘數第一個bit。若為1，則將被乘數右方補上32bit加入dataOut中，若為0，則不動作。最後將dataOut右移1bit。重複做32個循環便會得到完整乘法的答案。

1. Shifter:

先設計一個2對1的多工器，並串接32個MUX2\_1（共五次），因此全部會有32 \* 5 = 160個MUX2\_1。每32個為一組，每一組的位移量分別是20、21、22、23、24，因此每一組都分別會有1、2、4、8、16個in0被設為0，這是為了讓每個位元都能夠分別左移1、2、4、8、16個位元。若是多工器接收到的訊號是1就將in0傳出去，若為0則是將in1傳出去，透過這個方式就可以完成所需要左移的位元。

1. HiLo:判斷傳入的訊號是不是reset訊號，若是則reset答案為0，若否則將傳入的乘法答案分別存入Hi與Lo的暫存器中。
2. Mux:根據輸入的6bit控制訊號去選擇需要的計算結果。
3. Alu Control:

根據輸入的6bit控制訊號，決定該完成哪種運算。如果輸入的訊號為乘法計算，則開啟計步器，計算至32次時，輸出開啟HiLo暫存器的訊號，將乘法器的值放入HiLo暫存器中，並將計步器歸0。若非乘法運算，則將6bit的控制訊號分別assign給個別的運算訊號，並去執行個別所需的運算。

1. Test Bench:為所設計模組的測試平台，負責產生時脈週期，以讀檔的方式讀入控制訊號、第一個運算元、第二個運算元，將他們傳入各個元件中，以Alu Control決定要做哪一種運算，最後再將做完的運算結果輸出。
2. Icarus Verilog驗證結果與Waveform輸出圖形，並加以說明

* Icarus Verilog驗證結果

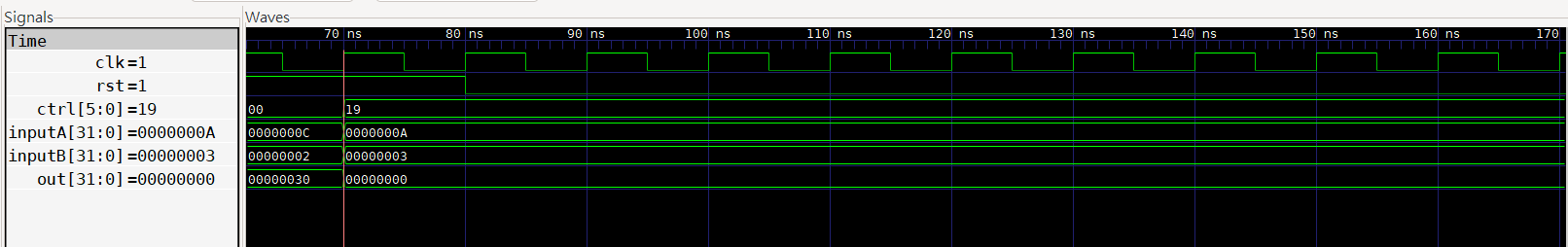
一張含有 文字 的圖片

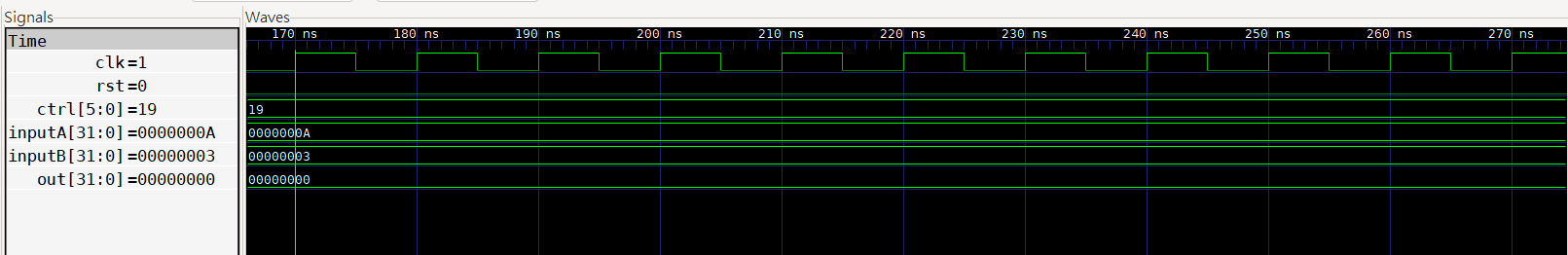
自動產生的描述

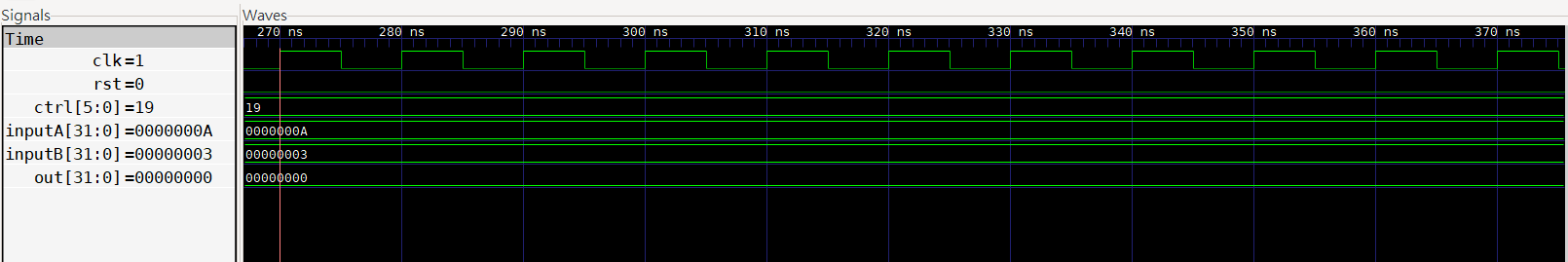
* Waveform輸出圖形

一張含有 文字, 監視器, 螢幕, 數個 的圖片

自動產生的描述





一張含有 文字, 室內, 螢幕擷取畫面 的圖片

自動產生的描述

* 說明

當讀入的控制訊號ctrl為，要選擇做AND運算，且讀入的inputA為，inputB為，則inputA & inputB的結果會是

⇒ ，與模擬結果相同

當讀入的控制訊號ctrl為，要選擇做OR運算，且讀入的inputA為，inputB為，則inputA | inputB的結果會是

⇒ ，與模擬結果相同

當讀入的控制訊號ctrl為，要選擇做ADD運算，且讀入的inputA為，inputB為，則inputA + inputB的結果會是

⇒ ，與模擬結果相同

當讀入的控制訊號ctrl為，要選擇做SUB運算，且讀入的inputA為，inputB為，則inputA - inputB的結果會是

⇒ ，與模擬結果相同

當讀入的控制訊號ctrl為，要選擇做SLT運算，且讀入的inputA為，inputB為，則inputA沒有小於inputB，因此會輸出0，與模擬結果相同

當讀入的控制訊號ctrl為，要選擇做SLL運算，且讀入的inputA為，inputB為，也就是12要左移2位元，相當於，與模擬結果相同

當讀入的控制訊號ctrl為，要選擇做MULTU運算，且讀入的inputA為，inputB為，則inputA inputB的結果會是，而因為做乘法運算需要32個週期，所以在前32個週期運算還沒完成時輸出都會是0，而完成運算後先後將控制訊號設為與，把乘法器算出來的結果存進Hi-Lo暫存器，因此算出來的結果為，沒有超過32位元，所以hi應為0 ，lo應為，與模擬結果相同。

1. 心得感想

透過這次的project讓我們更加的了解verilog的撰寫方式，過程中雖然遇到不少次的訊號全部為0或unkown、甚至在編譯時cmd出現I give up，但是在最後也都有找到問題點且順利解決。老師開出的一些條件，比如連接32個1bit alu bit slice、alu不可以使用always block或 procedure assignment、乘法器不可使用迴圈形式方式設計、shifter不可使用”>>”或”<<”等等，都可以說是構成讓我們學習硬體語言最扎實的基礎，藉由不停的翻閱講義、回想上課內容、與組員間的討論，讓我們更加的了解了整個Alu的架構。不過儘管在實作過程有許多挑戰，也花了非常多的時間去理解，但在實作後，也對於硬體語言與指令的操作都有更進一步的了解。