Midterm Project

班別:資工二甲

組別:40

:

10727124 劉宇廷

10727131 陳彥綸

10727153 呂宜鴻

1. 設計重點說明

tb\_ALU

以讀檔方式讀入測試資料(input.txt)，利用資料依序執行TotalALU，並將答案比對ans.txt的答案，最後將結果輸出於螢幕上。

TotalALU

在TotalALU中建立各種module(ALU\_Control/ALU/Divider/HiLo/shifter)，並設定各種接線，使用於各module中。

ALU\_Control

分別設定兩暫存器(temp/counter)，當Signal訊號為AND/OR/ADD/SUB/SLT/SRL時，將Signal的值傳給temp；特別處理當Signal訊號為DIV時，首先將counter歸零，判斷當posedge時counter++，並處理當counter=33時，temp設為6'b111111，代表開啟HiLo暫存器，給除法器放值進去。

ALU

設定暫存器temp暫時儲存答案，首先判斷如果reset=1，dataOut設定為32'b0，代表歸零；其餘情況判斷如果Signal=ADD，設定ctr=0，否則ctr=1，以Ripple-Carry的進位方式，依序呼叫ALU\_bit連接31個bit，最後一個位元執行ALU\_Set，目的是增加SLT的判斷，最後將結果temp傳給dataOut。

ALU\_bit

設定3個wire(tempAND/tempOR/temp)，tempAND儲存a&b的結果，tempOR儲存a|b的結果，temp儲存FA處理的結果，最後判斷Signal=AND，tempAND傳給sum；Signal=OR，tempOR傳給sum；Signal=SLT，less傳給sum；Signal=ADD/SUB，temp傳給sum。

ALU\_Set

功能與ALU\_bit差不多，特別處理最後一個位元，當Signal=SLT，將temp值傳給set，在ALU中最後一個位元的set會更改第一個位元的set，完成dataA<dataB時，結果為{31'b0,1'b1}。

FA

利用gate-level的方式，將輸入的位元做full-adder的處理，減法也可利用相同概念，可利用A-B=A+(-B)做相同處理。

Divider

當Signal為DIVU，rem設定為32'b0與dataA的結合，divr為dataB與32'b0的結合，之後當posedge時，判斷如果reset=1，初始化暫存結果的temp和儲存商數的quot，其餘情況先將rem減去divr，並判斷rem的最高位元是否為0，若是，代表rem>=0，quot左移一位並設定LSB=1；若否，代表rem<0

，quot左移一位並設定LSB=0，將divr持續往右移一位，最後判斷Signal=OUT時，代表除法器執行結束，將temp值更改為quot與rem[31:0]的結合，代表前32位元儲存商數，後32位元儲存餘數，最後將temp值傳給dataOut。

HiLO

主要功能是將64位元的除法器結果DivAns，分成HiOut/LoOut，前者是

DivAns[31:0]代表餘數，後者是DivAns[63:32]代表商數 。

shifter

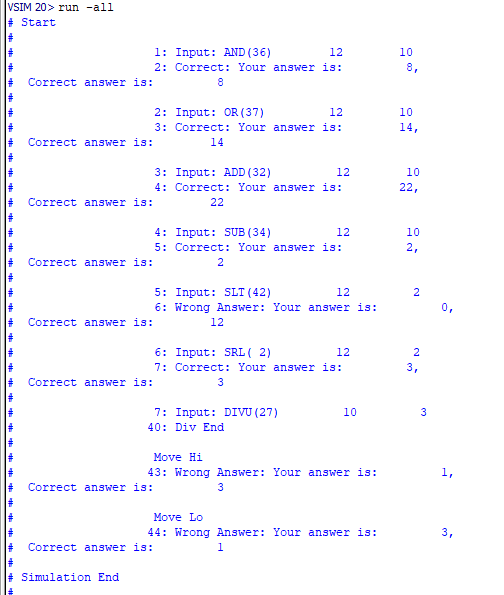
將輸入的dataA右移dataB字元，分別建立temp1~5表示每移一階所暫存的結果，首先判斷如果reset=1，dataOut歸零，其餘情況從第一階的dataA[0]開始，依序判斷如果dataB[0]為1，值為dataA[1]，否則維持原值，依序執行將結果存至temp1；第二階從temp1[0]開始，依序判斷如果dataB[1]為1，值為temp[2]，否則維持原值，依序執行並將結果存至temp2，以此類推，最後判斷如果dataB[4]為1，temp4[0]值為temp4[16]，否則維持原值，並將結果存至temp5，再將temp5傳給dataOut完成移位。

MUX

完成上述功能後的結果，利用MUX做最後的輸出判斷，如果Signal=AND/OR/ADD/SUB/SLT，dataOut為ALUOut；Signal=SRL，dataOut為Shifter；Signal=MFHI，dataOut為HiOut；Signal=MFLO，dataOut為LoOut。

1. Modelsim 驗證結果與 Waveform 輸出圖形

Modelsim:



由此圖得知:AND --- 12&10 = 8

OR --- 12 | 10 = 14

ADD --- 12 + 10 = 22

SUB --- 12 – 10 = 2

SLT --- 12 > 2 = 0 (由於ans.txt有誤，正確答案為0)

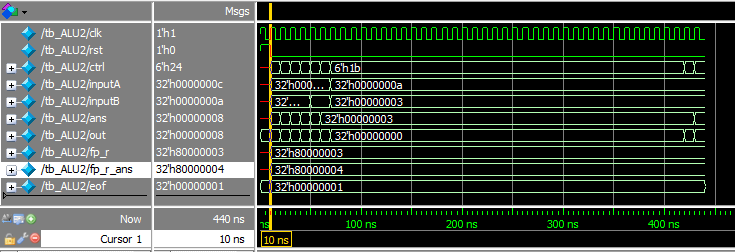
SRL --- 12>>2 = 3

DIVU --- 10 / 3 = 3....1

HiOUT--- 1 (由於ans.txt有誤，正確答案為1)

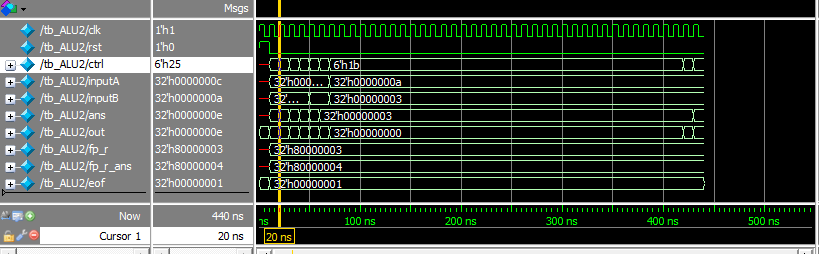
LoOUT---3(由於ans.txt有誤，正確答案為3)

Waveform:

And:

當control的值等於6’h24(也就是36)時執行AND運算，

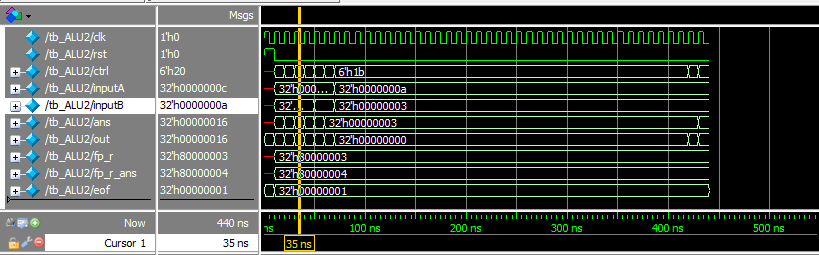
使out = inputA & inputB = 12 & 10 = 8。

OR:

當control的值等於6’h25(也就是37)時執行OR運算，

使out = inputA | inputB = 12 | 10 = 14。

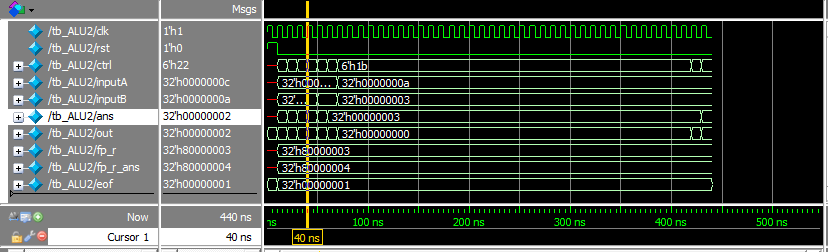
ADD:



當control的值等於6’h20(也就是32)時執行AND運算，

使out = inputA + inputB = 12 + 10 = 22。

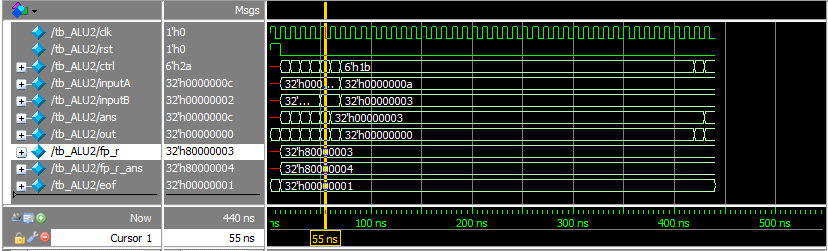
SUB:



當control的值等於6’h22(也就是34)時執行SUB運算，

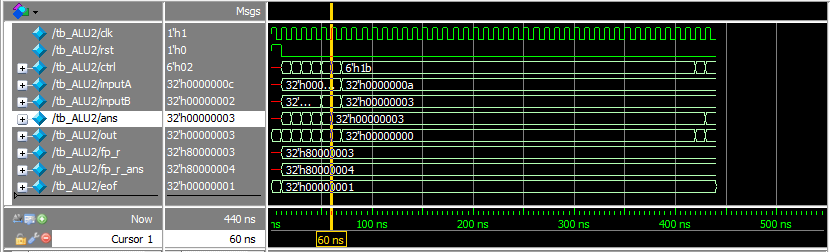
使out = inputA - inputB = 12 - 10 = 2。

SLT:



當control的值等於6’h2a(也就是42)時執行SLT運算，如果inputA < inputB 的話，out = 1，而inputA >=inputB，out = 0，此圖中inputA = 12，inputB = 2，因此out = 0。

SRL:

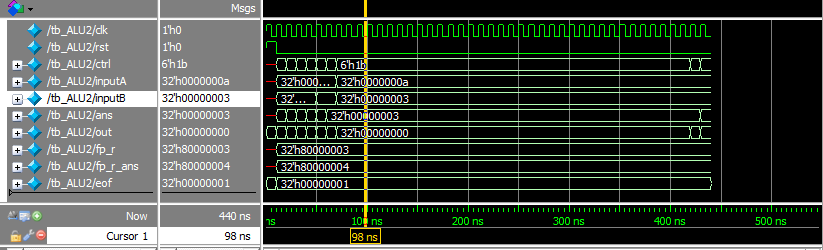


當control的值等於6’h02(也就是2)時執行SRL運算，

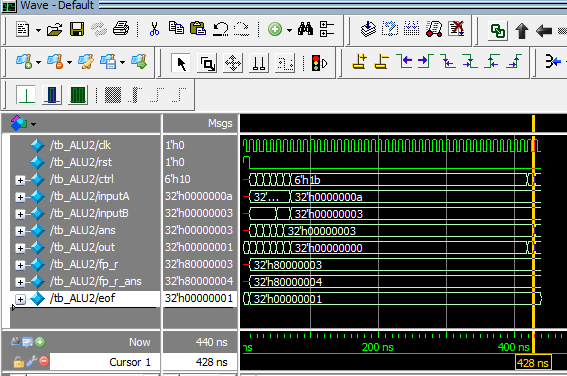
使out = inputA >> inputB = 12 >> 2 = 3。

DIVIDER:

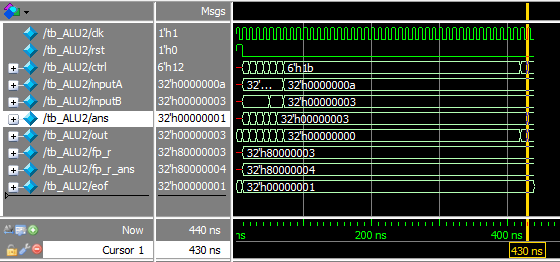
DIVU



HI



LO



當control的值等於6’h1b(也就是27)時執行DIVU運算，運算完成後將商傳至LO暫存器，餘數傳至HI暫存器，從這些圖可知10 / 3 = 3 ….1。

1. Datapath 與詳細架構圖

Datapath:



ALU架構圖:



Divider架構圖:



Shifter架構圖:



1. 心得感想:

這一次的project的內容主要並不難，主要在於不熟悉verilog，所以寫起來會覺得比較不順，或是不知道要怎麼開頭，而覺得最難的部分就是在寫除法器時，因為一開始不太知道要怎麼使用clk來計算次數和用64-bits做運算和移位，不過透過這次的project也學到了怎麼用testbanch和看waveform來看自己的程式是哪裡有問題，或是觀察每進行完一次計算後每個bit分會有什麼改變，相信這對之後寫verilog的project會有很大的幫助。

1. 各組員分工方式與負責項目

verilog程式 --- 陳彥綸、劉宇廷、呂宜鴻

設計重點說明 --- 劉宇廷

結果截圖及說明 --- 陳彥綸

心得感想 --- 呂宜鴻

Datapath與架構圖 --- 陳彥綸、劉宇廷、呂宜鴻