計算機組織 Midterm Project: ALU Design

112學年度第2學期

老師：朱守禮 老師

學生：11127132羅海綺、11127137黃乙家、11127138林雨臻

一、背景

為了使學生充分了解，該如何使用Verilog HDL與Modelsim模擬器，以計算機組織課程講義：

[1] Chapter 3:Arithmetic for Computers Part 1: ALU (Add/Subtraction)

[2] Chapter 3:Arithmetic For Computers Part 2: Multiplication and Division

[3] Chapter 3:Supplement: Verilog Concepts

為基礎，設計 ALU 與乘法器，以供 Final Project 之用。

二、方法

1.功能說明：本Project包含AND, OR, SUB, SLT, SLL, MULTU…等7項功能。

2.設計要求：Datapath 與詳細架構圖(並以 PowerPoint 或 Word 設計繪製)。

(1) ALU:

(a) 設計重點說明：

* + - 包含 32-bits AND, OR, ADD, SUB, SLT 等功能。
    - 使用Gate-Level Modeling與Data Flow Modeling(Continuous Assignments)，從Full Adder開始，以Ripple-Carry的進位方式，連接32個1-bit ALU Bit Slice。
    - 本模組為組合邏輯(Combinational Logic)。

(b) Datapath與詳細架構圖：

(2) Multiplier:

(a) 設計重點說明：

* 為32-bits無號數乘法Sequential Multiplier，須採用Second Version Sequential Multiplier來設計。
* 可使用Always Block或Procedure Assignment來設計，但【不接受迴圈形式的設計】。
* 本模組為循序邏輯(Sequential Logic)，因此須以 Clock 訊號同步。

(b) Datapath與詳細架構圖：

(3) Shifter:

(a) 設計重點說明：

* 32-bits Barrel Shifter，完成邏輯左移運算。
* 以 Data Flow Modeling(Continuous Assignments)完成，不
* 能直接用 '>>'或'<<' operator，亦不可使用 Always Block 或 Procedure Assignment 來設計。
* 以 160 個 Mux 實現 Shifter。
* 本模組為組合邏輯(Combinational Logic)。

(b) Datapath與詳細架構圖：

(5) HiLo暫存器:

(a) 設計重點說明：

* 乘法器計算完後，儲存計算結果之64-bit暫存器。
* 本模組為循序邏輯(Sequential Logic)，因此須以 Clock 訊號同步。

(b) Datapath與詳細架構圖：

(5) MUX:

(a) 設計重點說明：

* Data Flow Modeling 設計。
* 本模組為組合邏輯(Combinational Logic)。

(b) Datapath與詳細架構圖：

(6)ALU Control:

(a) 設計重點說明：

* Data Flow Modeling 設計。
* 本模組為組合邏輯(Combinational Logic)。

(b) Datapath與詳細架構圖：

* 以上皆不可包含延遲敘述。
* 不接受迴圈形式的設計。
* 不能使用always@(\*)敘述。

程式架構：

TotalALU:

ALUControl

ALU

ALUbit

FullAdder

MUX4\_1

Multiplier

Shifter

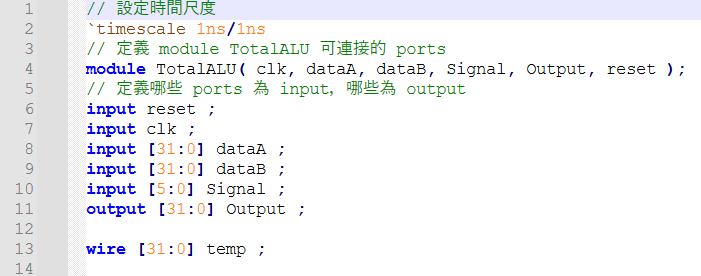
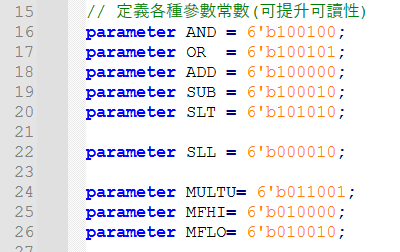
MUX2\_1

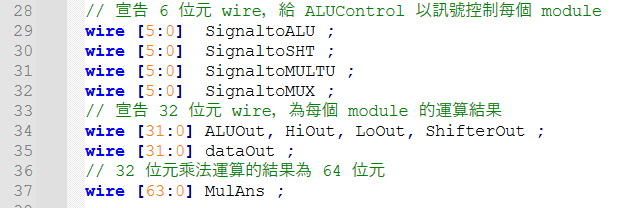
HiLo

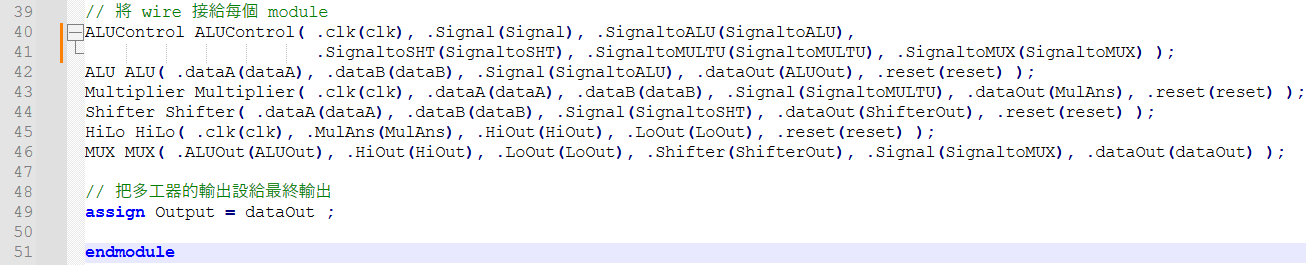
MUX

三、結果

(1) TotalALU:





(2) ALUControl:

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

(3) ALU:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

(4) ALUbit:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

(5) FullAdder:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

(6) MUX4\_1:

一張含有 文字, 螢幕擷取畫面, 軟體, 行 的圖片

自動產生的描述

(7) Multiplier:

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

一張含有 文字, 字型, 行, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 字型, 行, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

(8) Shifter:

一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 文件 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 文件, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 文件 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 文件 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 文件 的圖片

自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

(9)MUX2\_1:

一張含有 文字, 螢幕擷取畫面, 陳列, 字型 的圖片

自動產生的描述

(10) HiLo:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

(11) MUX:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

(12)tbALU:

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 字型, 數字, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面 的圖片

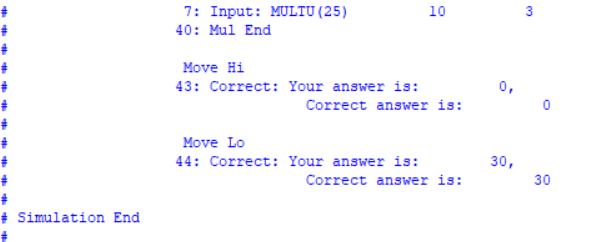
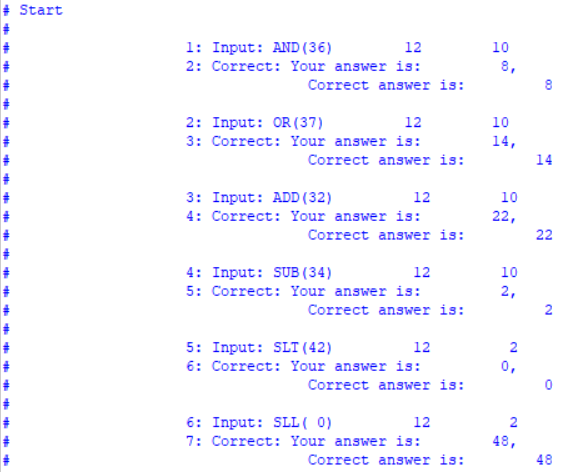
自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

四、討論

(1) Modelsim 驗證結果：



讀入 input.txt 後，根據每個 opcode 算出答案，

第一筆為 AND 運算，

第二筆為 OR 運算，

第三筆為 ADD 運算，

第四筆為 SUB 運算，

第五筆為 SLT 運算，，計算的

取 sum 的最高位 ，因此

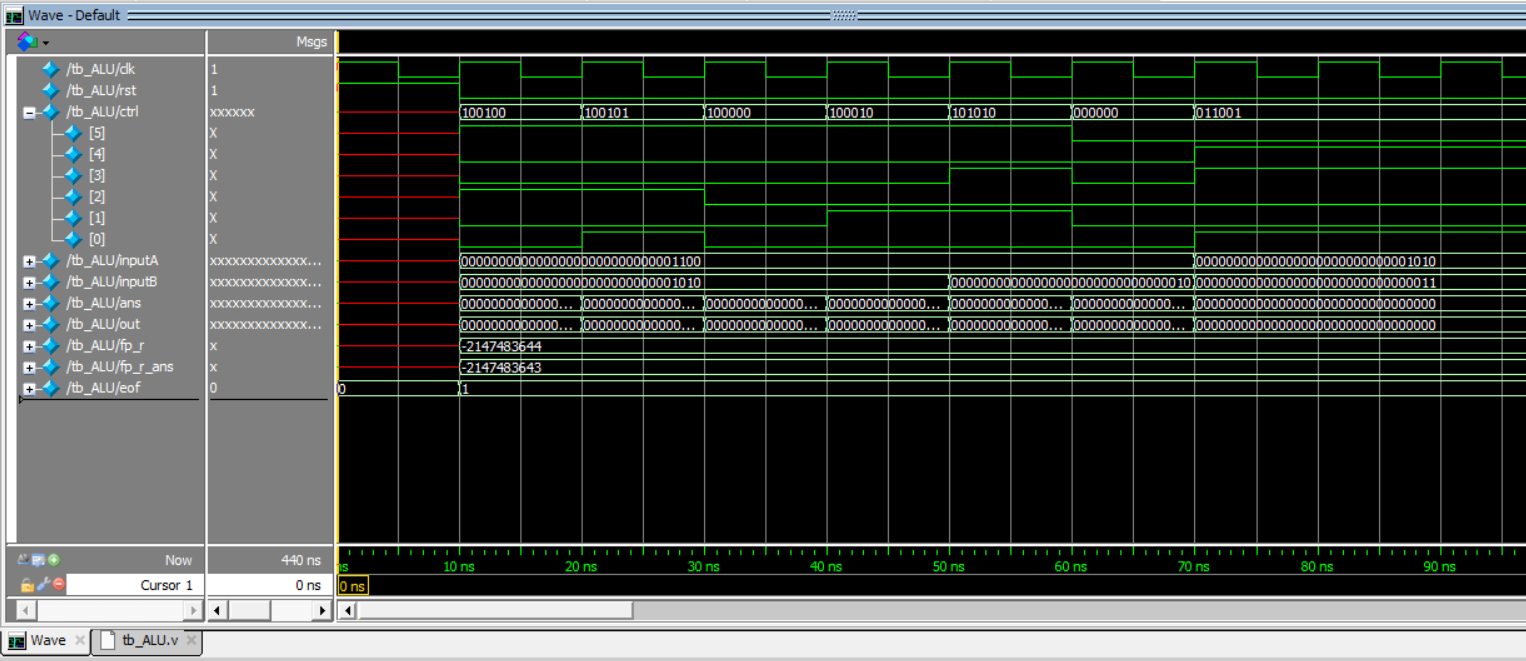
第六筆為 SLL 運算，

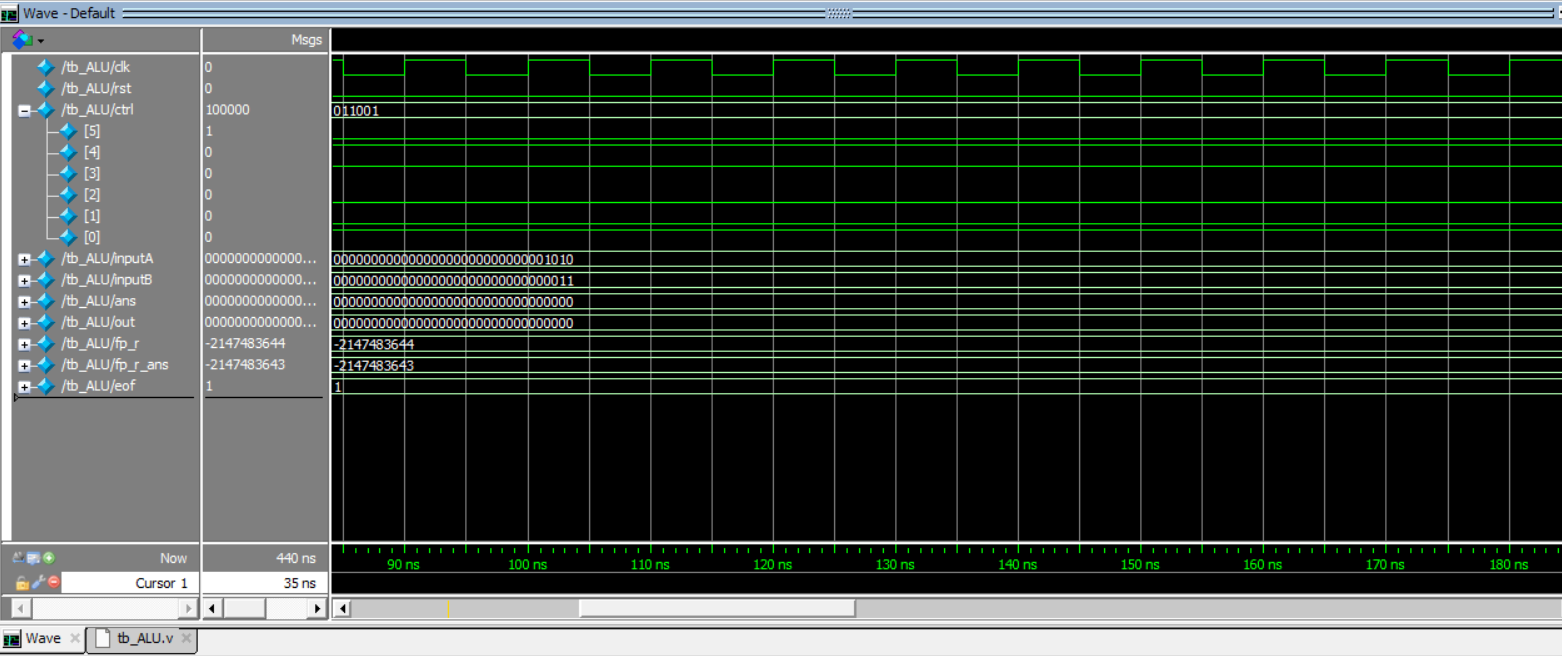
第七筆為 MUL 運算，使用第二代乘法器，

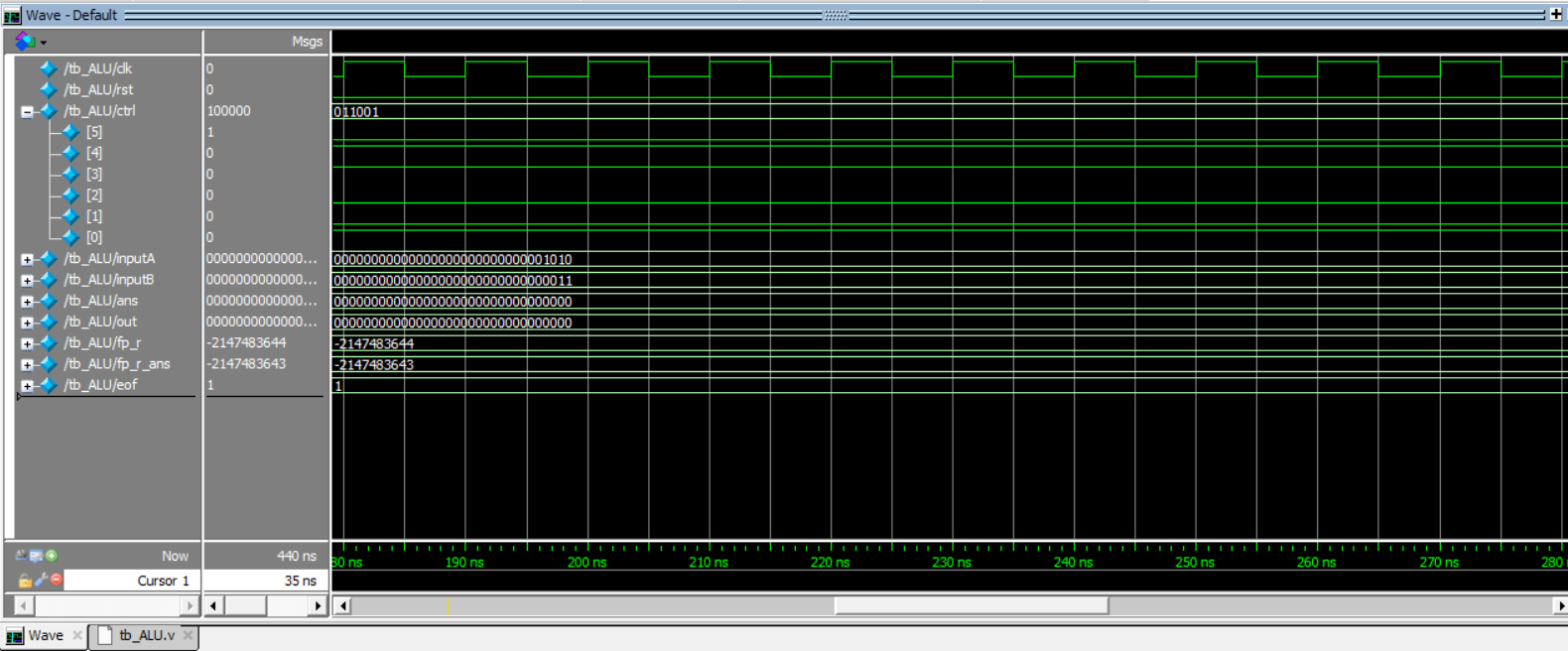
第一輪：確認乘數第 0 位的值，目前為 1，因此將乘數放進乘積的左半邊，並將乘數與 product 右移一位，

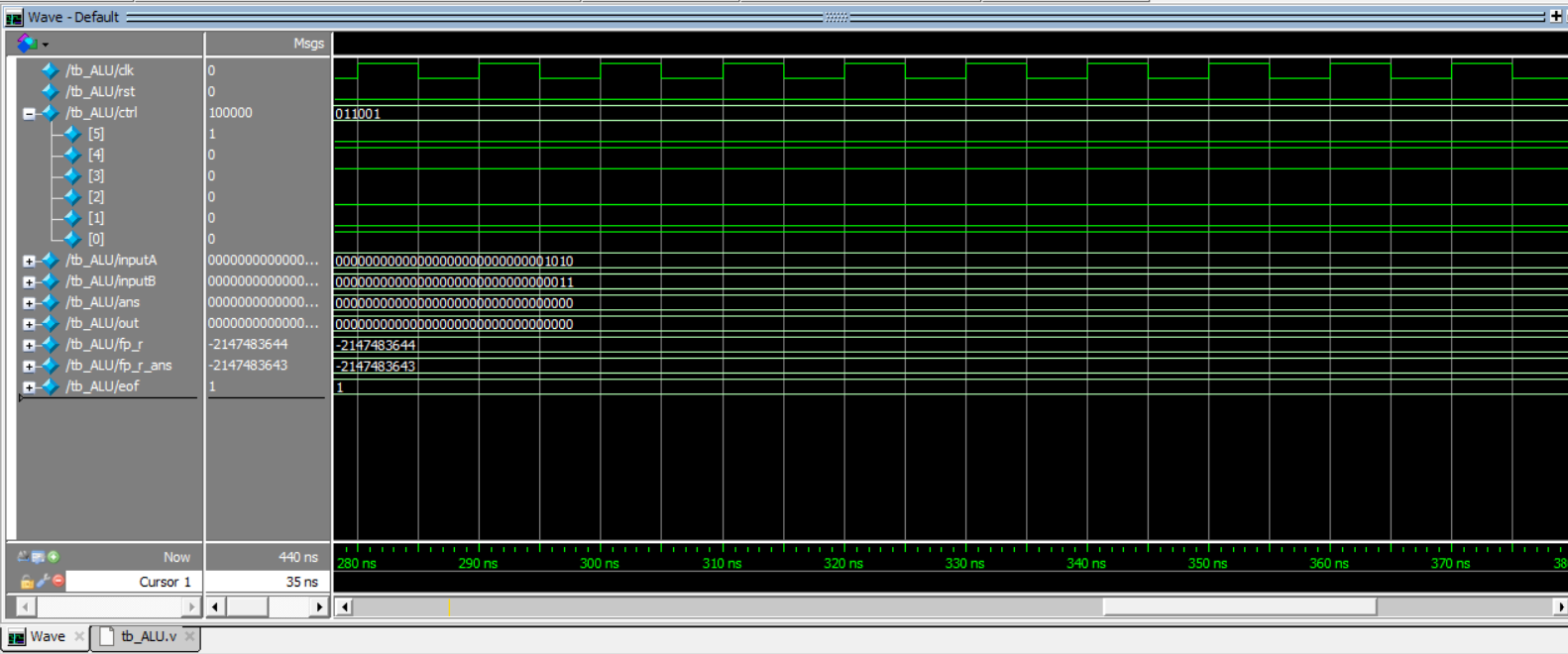
第二輪與第一輪相同，第三輪發現乘數第 0 位是 0，因此乘數與乘積直接右移即可，以此類推直到第 32 輪，最後 ，Hi 暫存器為 ，Lo 暫存器為

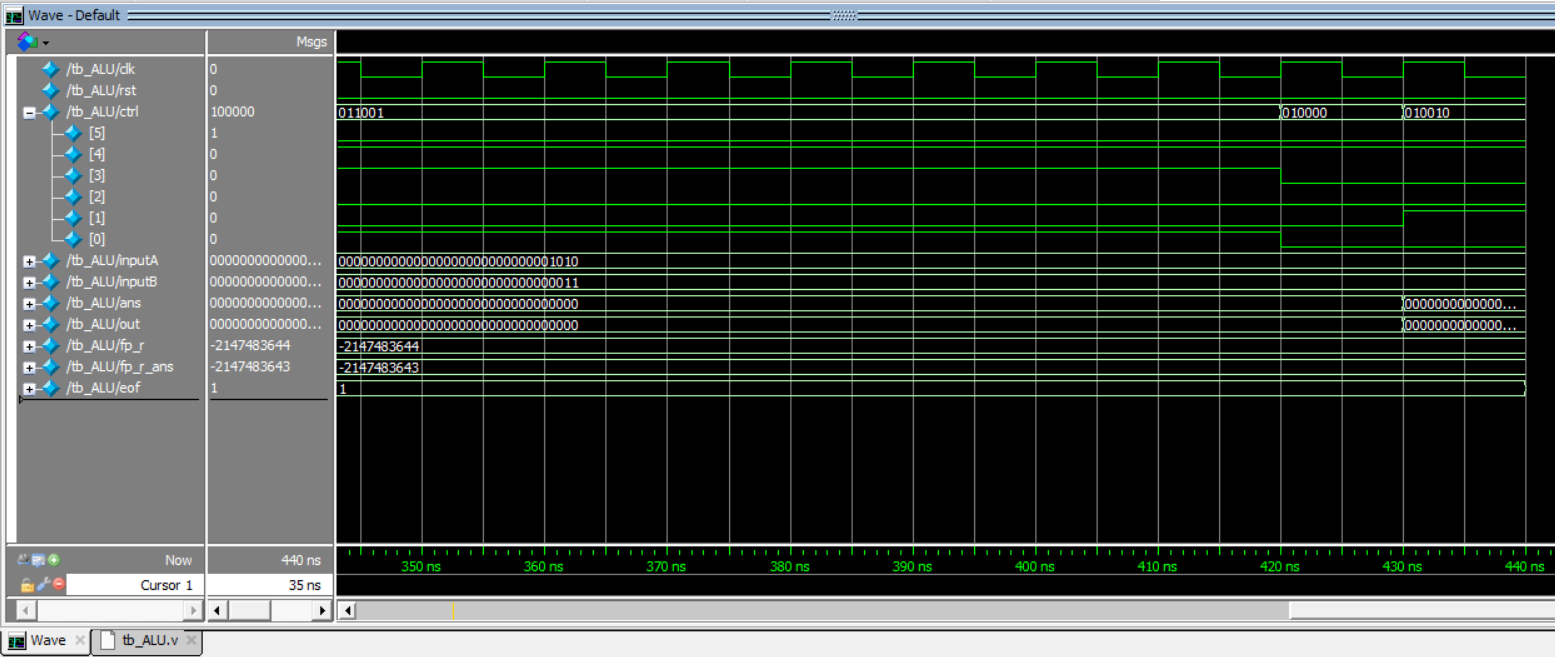
(2) Waveform 輸出圖形：



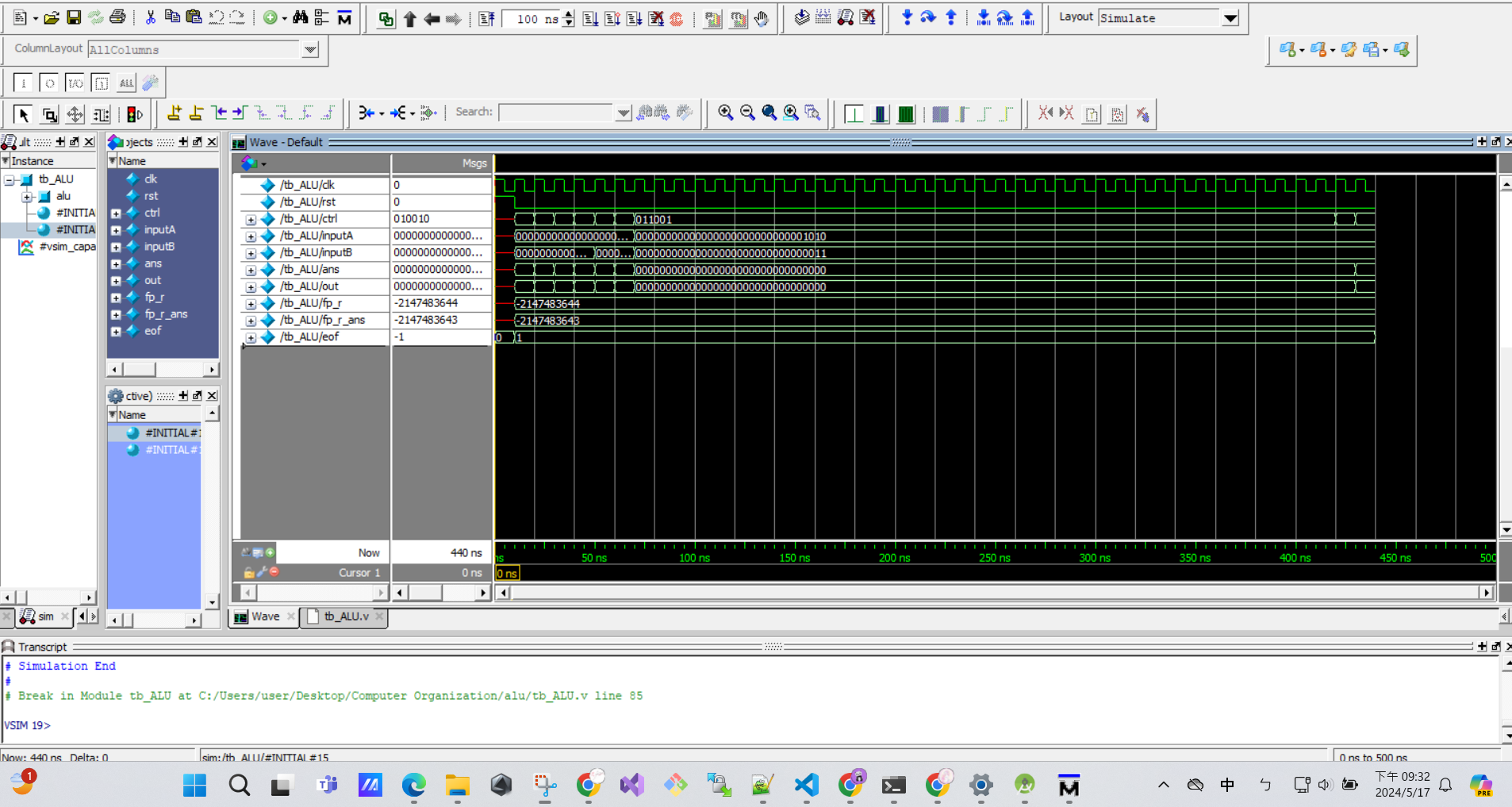








1. 結論

完整波型圖：

各組員分工方式與負責項目：

1. 分工方式：

組員討論寫法後，一人寫一支程式，其他人透過加註解及撰寫書面報告

的方式理解程式，為求所有人都理解程式內容。

1. 負責項目：
2. 程式碼：黃乙家、林雨臻、羅海綺
3. 書面報告：黃乙家、林雨臻、羅海綺

心得感想：

我們認為本次的計組期中專案太晚公告題目內容，因為儘管期中考週結束了，卻仍然有許多科目的project也才剛要開始，同時間我們組員中還有人必須準備多益檢定考試、程式考試，以及資結作業等等。

因為還不是非常熟悉Verilog的語法，因此在將程式轉換成電路的過程中，我們理解了一段時間。最後我們是由一個對於 Verilog 最熟悉的組員寫最複雜的module，其他人在那個module完成時，負責加上註解與理解 module 內容，最後再整理到書面報告上，也因為這種分工模式，負責製作報告與加註解的組員們可以在一有疑惑時就立馬發問，直到理解。

此次專案中特別的一點是，除了完成程式需要達成的事情之外還加入了一些限制規範，希望我們能去思考如何善用語法，以另一種 model 撰寫而不影響到電路本身的運作，這使我們可以更加理解那些語法的實際作用。雖然專案題目說的這些需求有可能增加程式的複雜度，但這同時可以幫助我們以後在編寫硬體描述語言，想設計自己的電路的時候熟悉基本的概念。

對於 Debugging，我們認為應該要先讓大家學會如何以波形圖來抓出錯誤，可以的話再介紹 modelSim 更多使用方式，因為在業界上有許多 IC 設計公司還是以 ModelSim 作為 Verilog 模擬平台，這樣才能夠幫助大家儘早熟悉編寫 Verilog。

六、未來展望

(1) 熟悉Verilog語法。

(2)