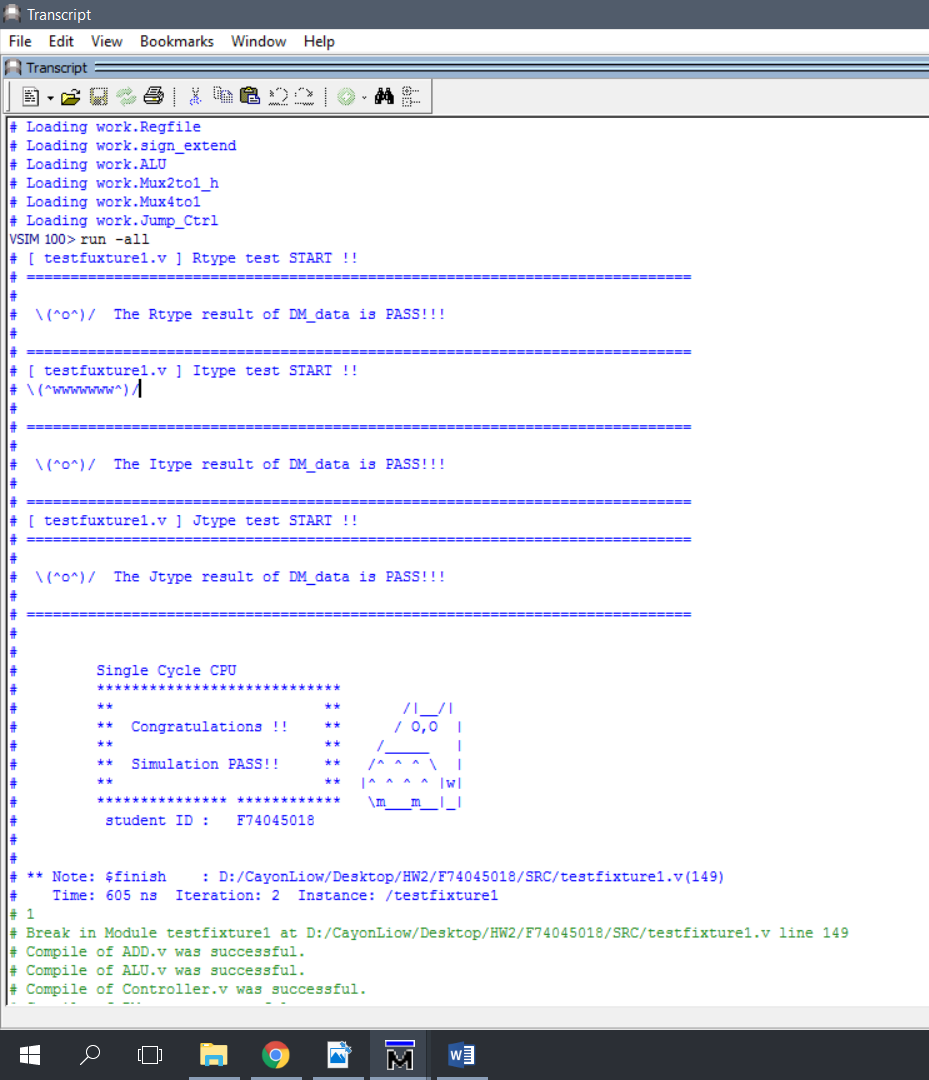
**Computer Organization 2017**

**HOMEWORK II**

系級: 資訊108 學號: F74045018 姓名: 廖其忻

**實驗結果圖(snapshot of result)**

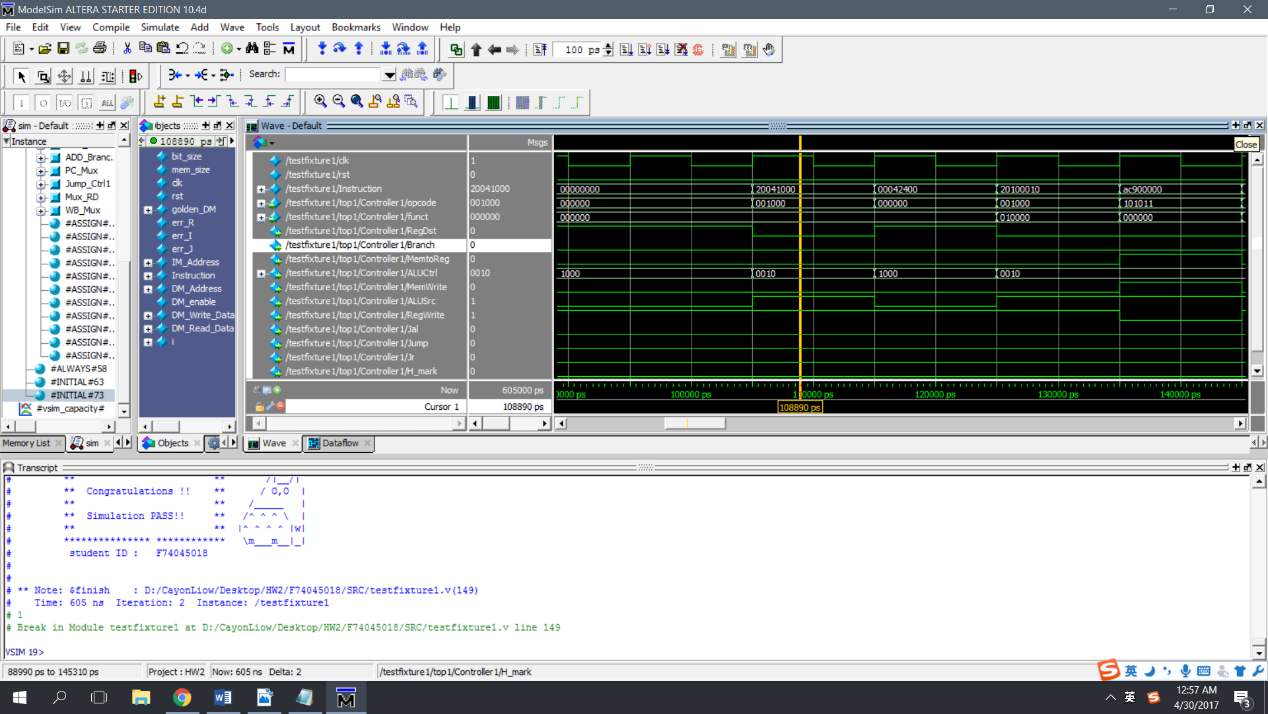


**指令波型圖( Instructions waveform )**

選擇的指令：

20041000 //addi $4, $0, 4096 main ; 7: addi $a0, $0, 4096

Controller:



Reason:

Opcode = 001000 in base 2 = 8 in base 10 = addi in I-type

Funct = 000000 in base 2, but not used in addi

RegDst = 0, Rt is selected instead of Rd

MemtoReg = 0, ALU\_result is selected instead of final\_RD\_data

ALUCtrl = 4’b0010, action of addition is activated in ALU

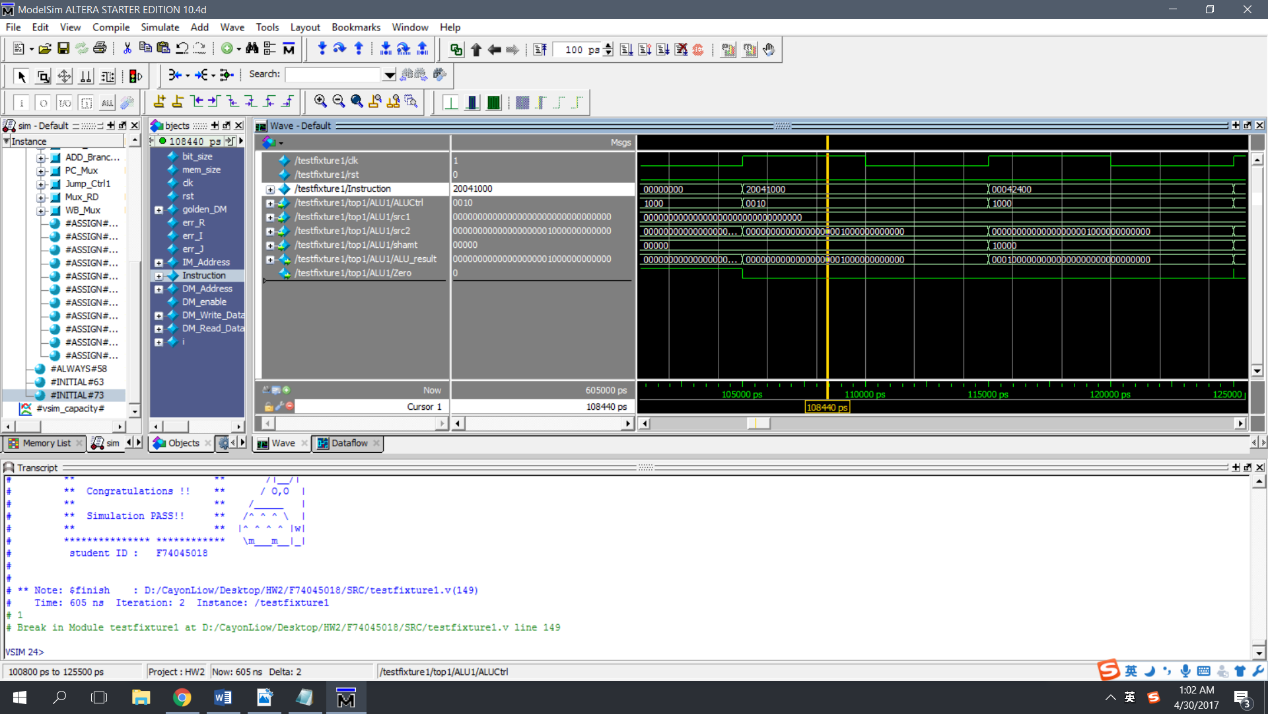
ALUSrc = 1, se\_imm ( result of sign\_extension) is selected instead of Rt\_data

RegWrite = 1, final result is stored back to Rt

Jal = 0, PC+8 is not selected

Branch = 0

ALU



Reason:

ALUCtrl = 4’b0010, action of addition is activated

Src1 = 0(in 32bits), value of Rs ($zero), since there is first instruction, value of all registers is initialized to 0

Src2 = 32’b0000000000000000001000000000000(4096 in base 10)

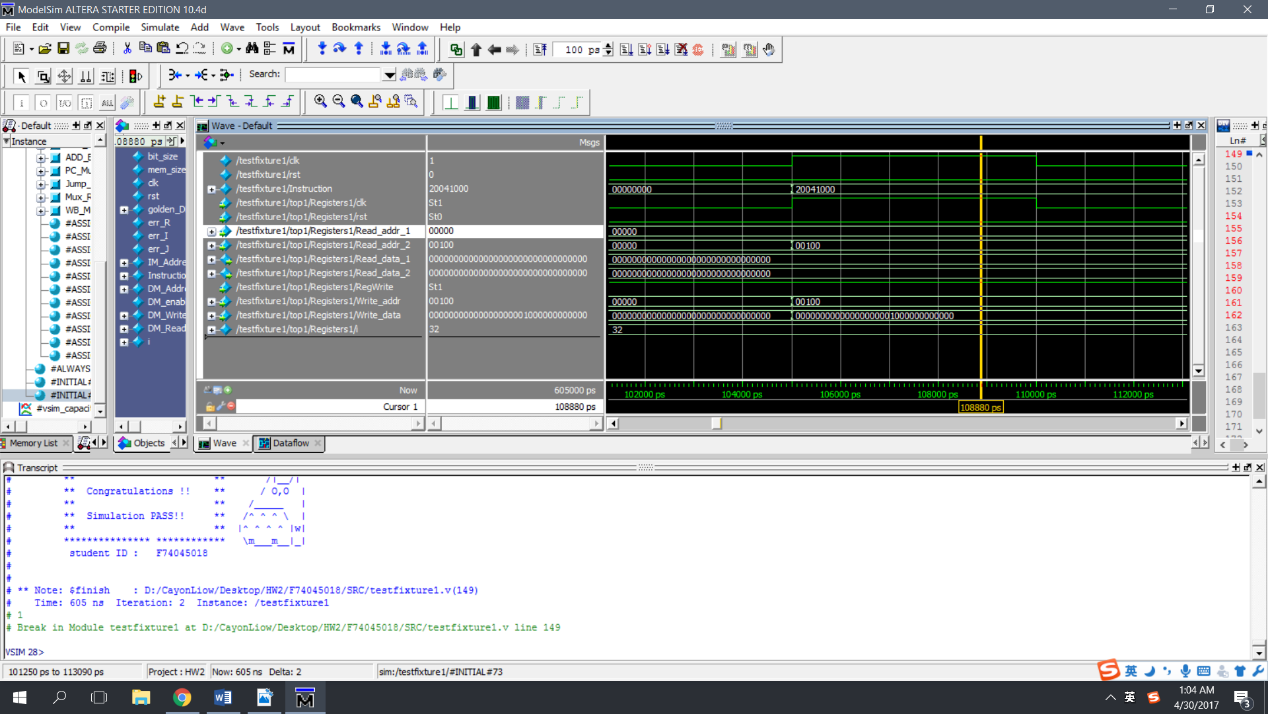
immediate after sign extension since ALUSrc = 1

ALU\_result = 32’b0000000000000000001000000000000,

Result of addition of src1 and src2

Zero = 0, since src1 != src2

Regfile



Reason:

Read\_addr\_1 = 00000, address of Rs, ($zero)

Read\_addr\_2= 00100, address of Rt, $4= ($a0)

Read\_data\_1 = 0 (in 32bits) value of $zero

Read\_data\_2 = 0 (in 32bits) value of $4 ($a0)

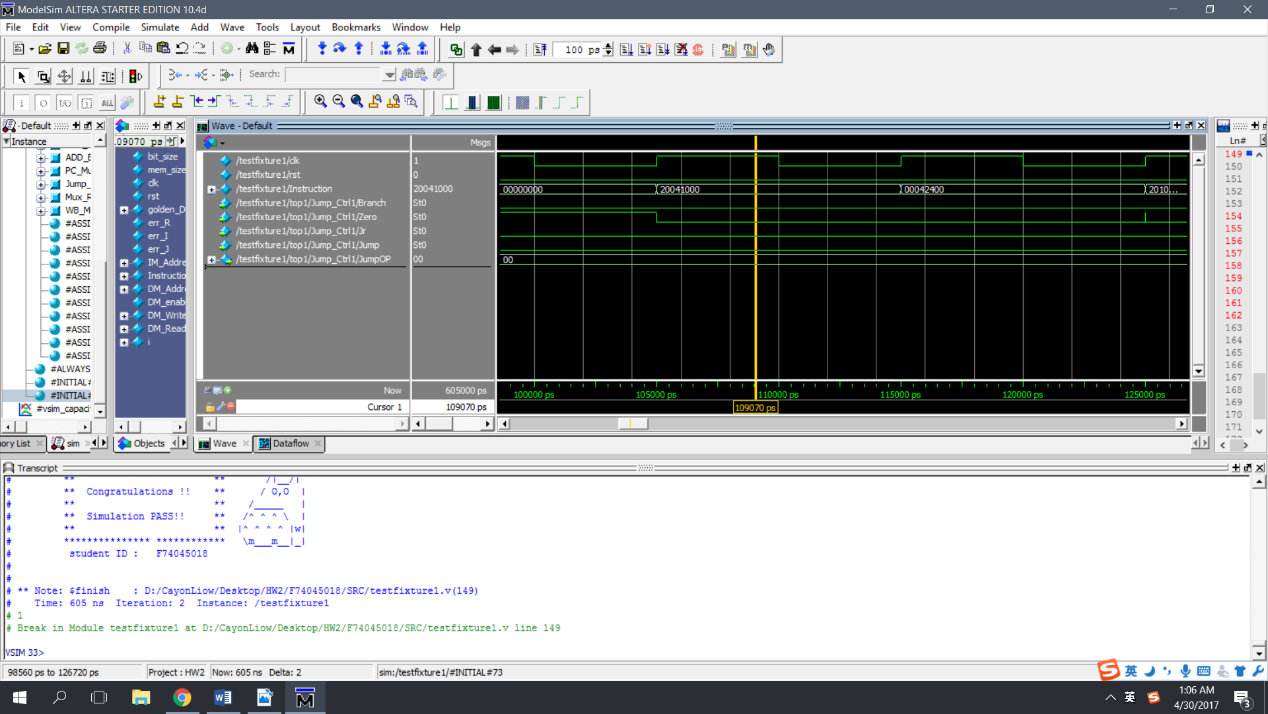
RegWrite = 1, final result is stored back to Rt

Write\_addr = 00100, address of Rt since RegDst= 0

Write\_data = 32’b0000000000000000001000000000000(4096 in base 10)

Since Jal=0, it is result of addition in ALU

Jump\_Ctrl

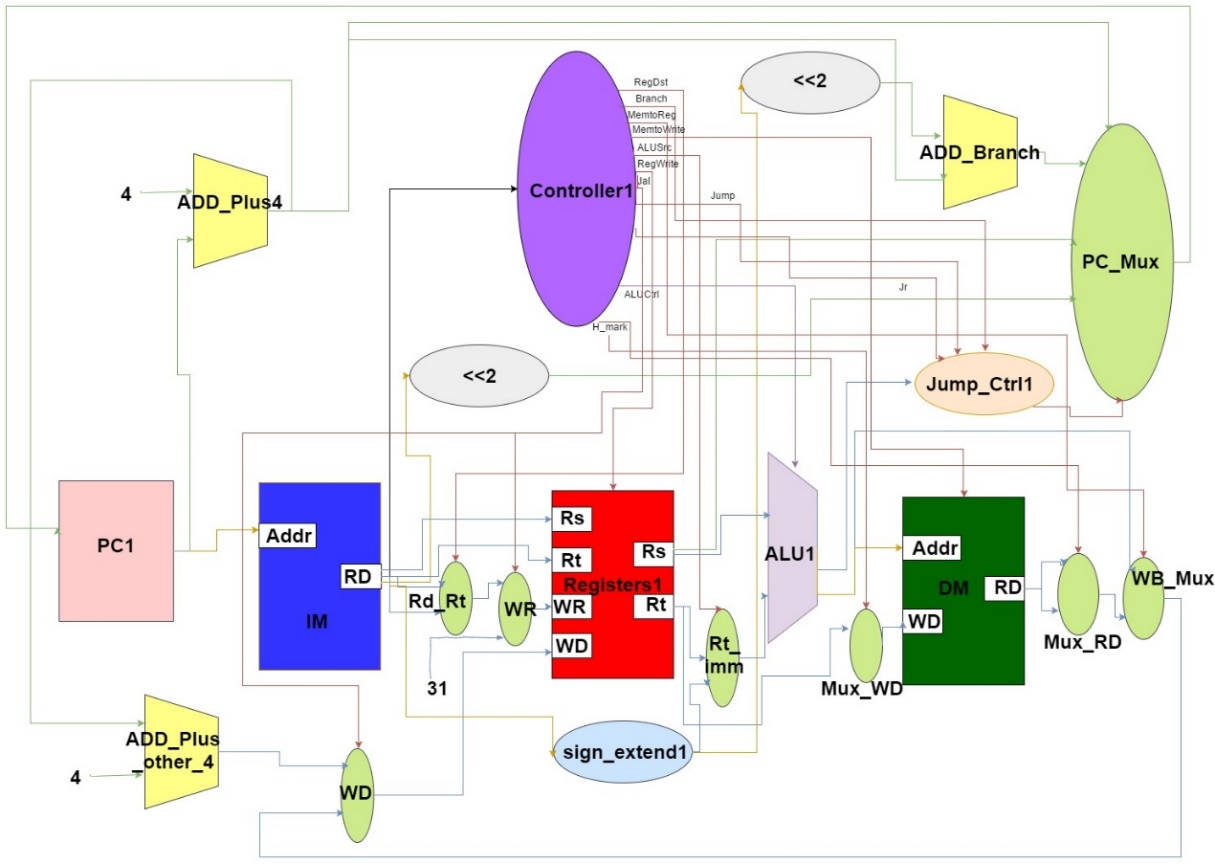


Reason:

All Inputs are 0, therefore JumpOP =2’00

And the input of PC+4 is selected in the multiplexer(PC\_Mux) to be the next address of instruction

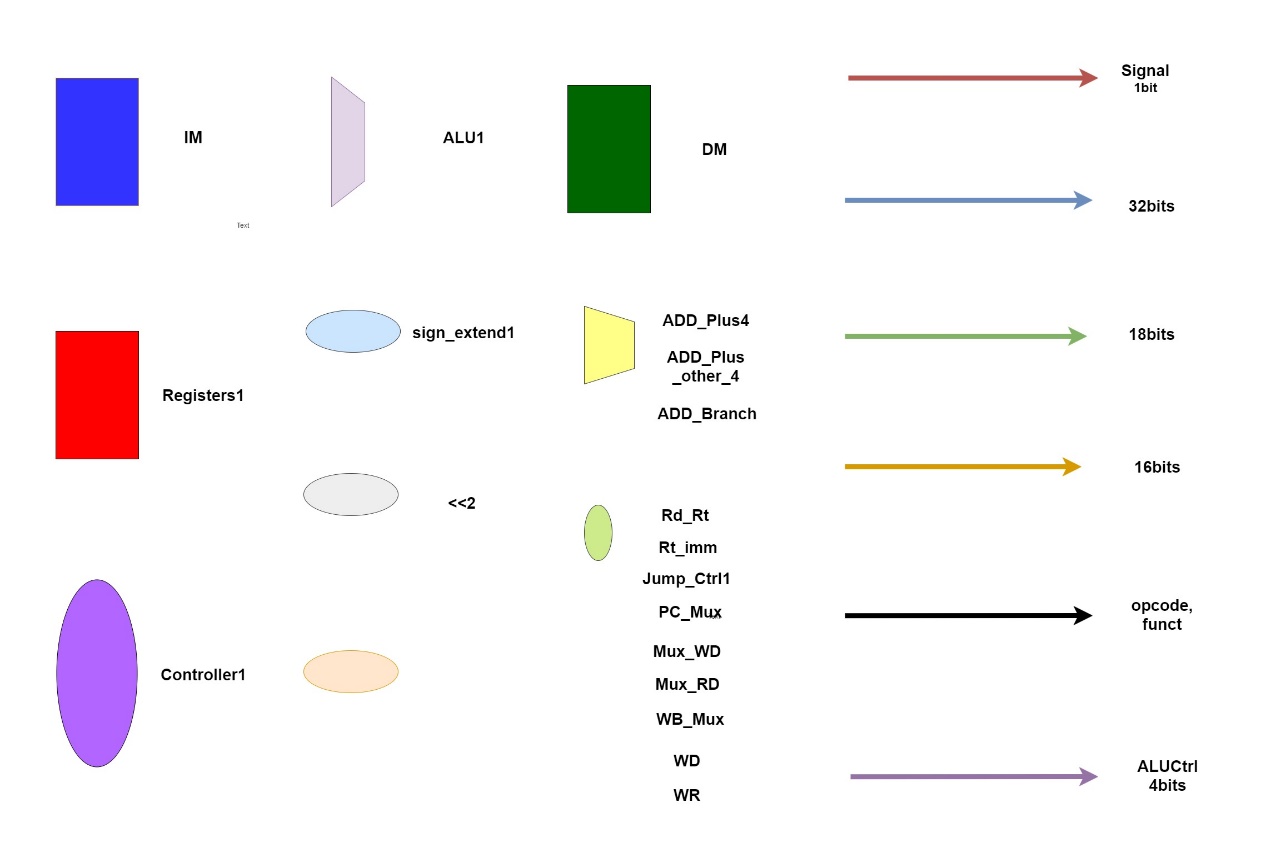
CPU datapath



（原圖尺寸A3版本鏈接：<https://drive.google.com/open?id=0B2qLIO0KTSixbG9HVW90SmVacjg>）

Reason:

在Controller 里多加了用來控制 lh與sh 的 signal.架構與課本, 作業說明所給的圖大概相同



（原圖尺寸A4版本鏈接：https://drive.google.com/open?id=0B2qLIO0KTSixYkJreTBFTjl0dGM）

**心得(Report)**

在這次的作業中，因為上課還沒有上到這一章，所以我們需要自學來了解這次的作業。這是在這次作業中遇到最大的困難，因為我的自我學習能力很差，所以讀了很久才弄懂了整個 Single Cycle Processor 的架構跟原理。

再來的是，在寫 Verilog 的時候，也寫的很卡。而且在助教所給的範例里，與我在課本所讀到的架構長得不一樣，也因此寫錯了兩天，浪費了兩天的時間，然後透過寄信及在課堂上問助教，問題才解決。

最後的是IM\_data的路徑問題，因為檔案沒有放到project 里所以無法讀取檔案。