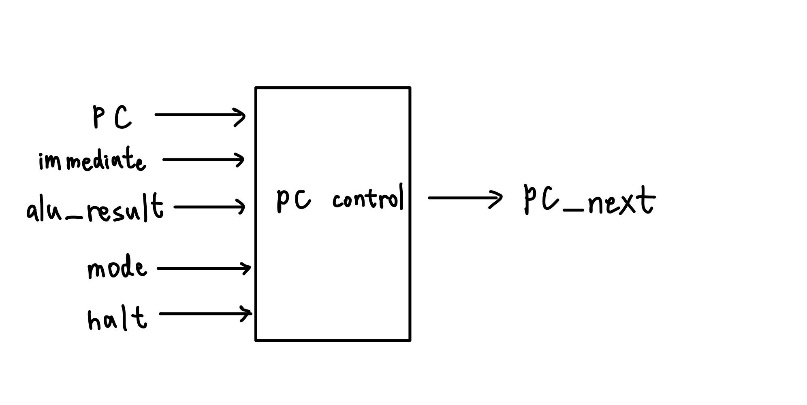
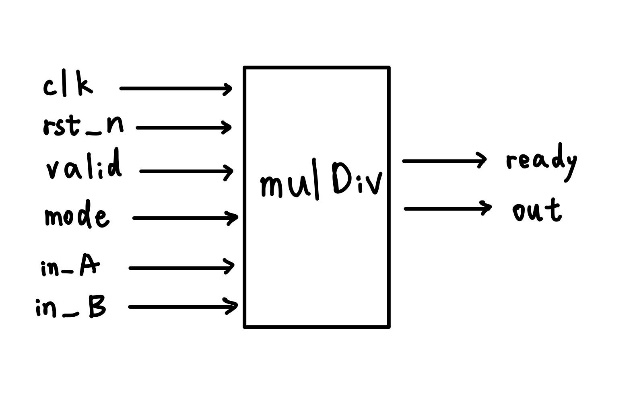
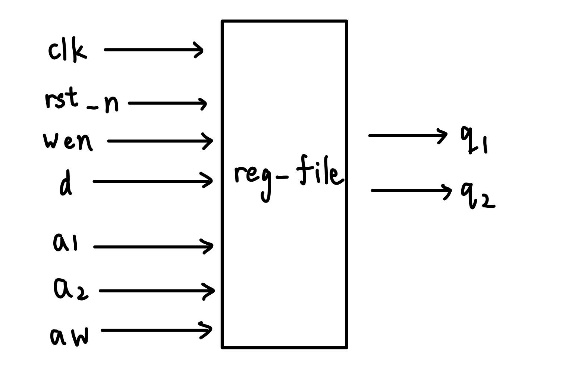
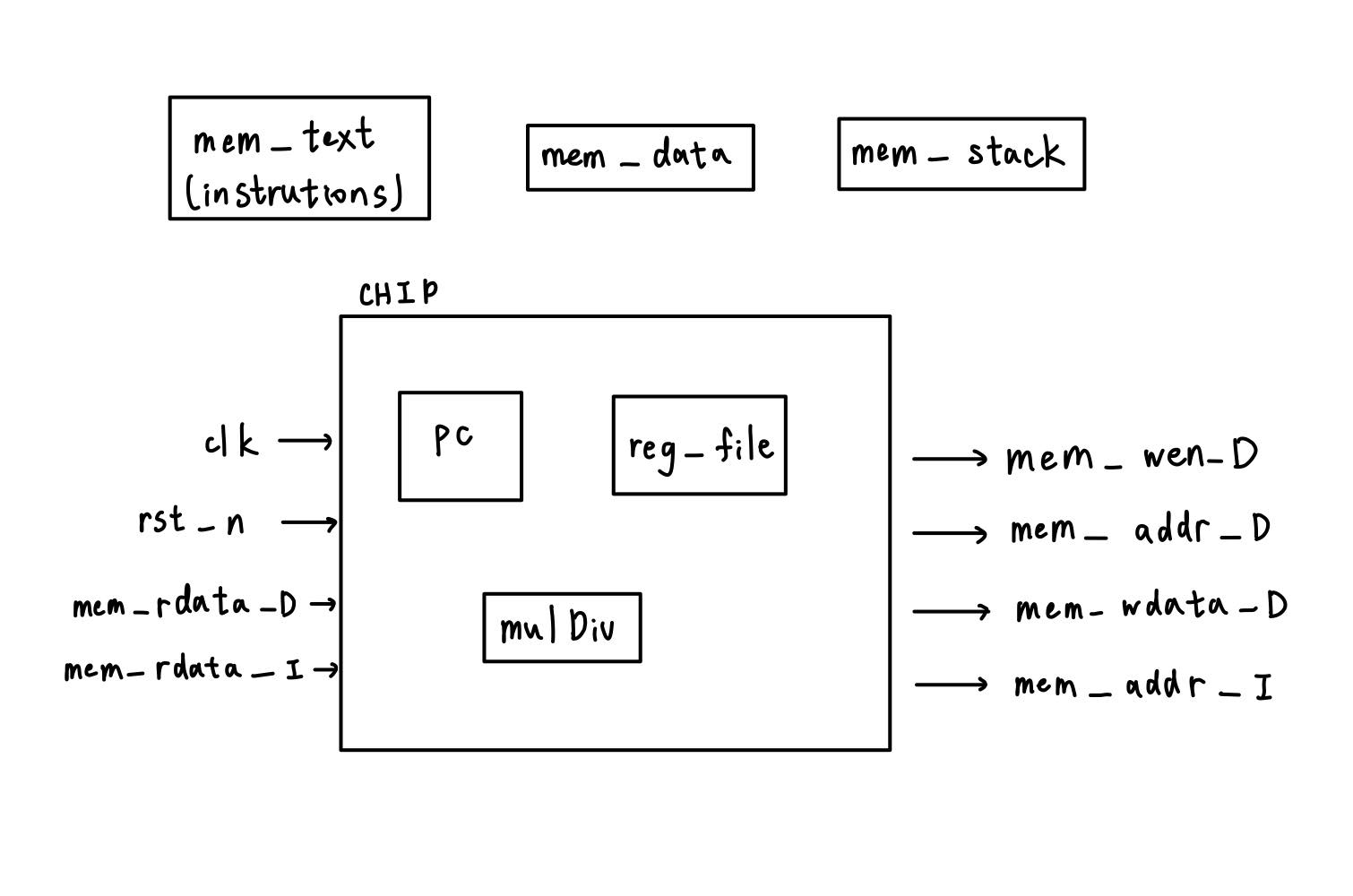
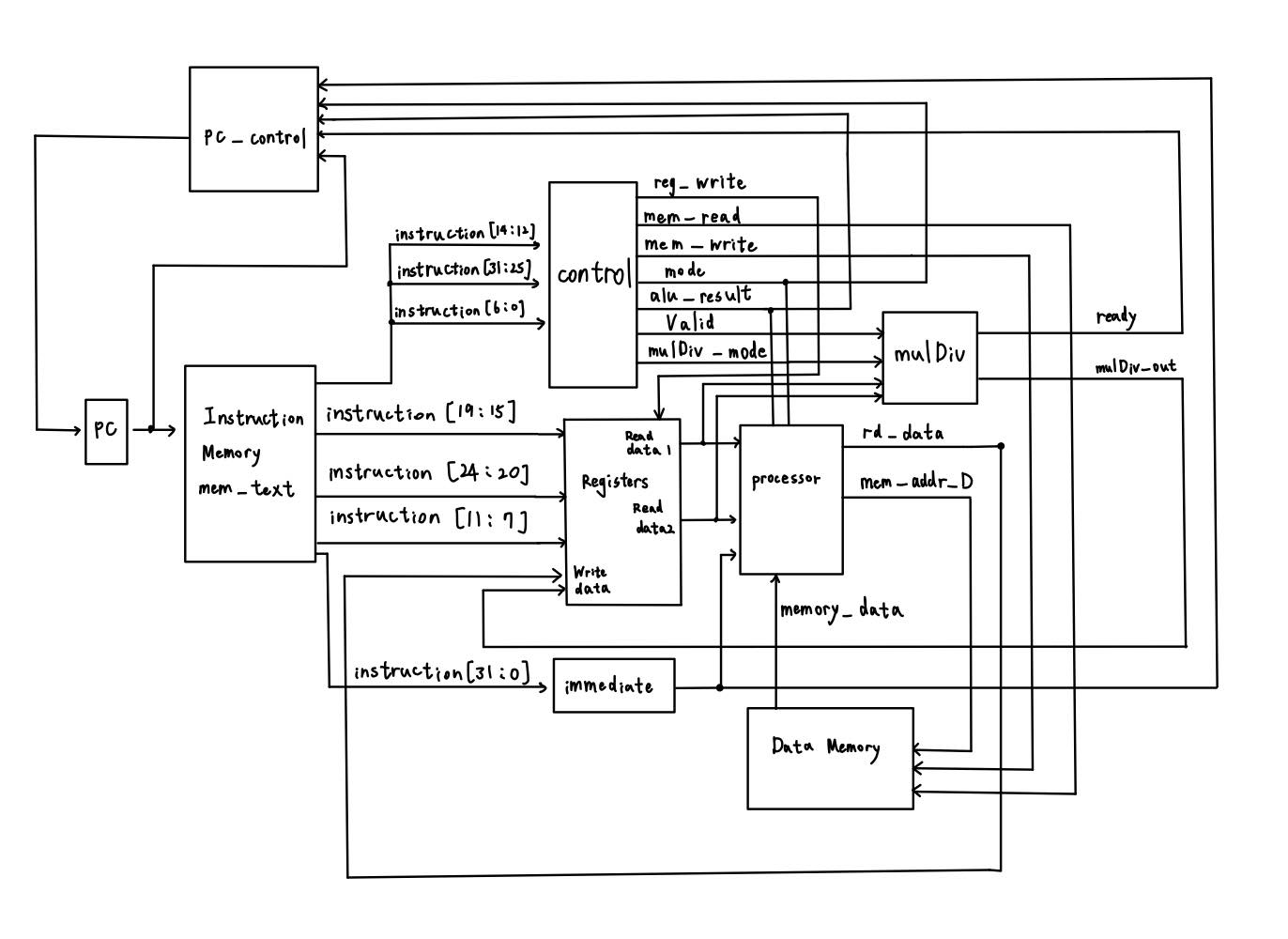
Report

1. Architecture and Data path





PC control:

1. Input: current PC, immediate, alu\_result, mode, halt
2. Output: PC\_next
3. Function: get the next PC. When dealing with multicycles, the halt stops the PC until ready set up. The mode will change when dealing with a branch and getting the correct next PC. Otherwise, PC\_next is PC + 4.

mulDiv:

1. Input: clk, rst\_n, valid, mode, in\_A, in\_B
2. Output: ready, out
3. Function: get the correct result of multiplication or division after 32 cycles.

Reg\_file: (designed by TA)

1. Input: clk, rst\_n, wen, d, a1, a2, aw
2. Output: q1, q2
3. Function: handle register affairs.

Data Flow:

When the process starts, PC control decides which instruction PC has to fetch. Then PC fetches the instruction from the instruction memory (memory text). After that, get the rs1, rs2, rd, immediate from the instruction. Pass instructions into the “control” unit, decides which operation to do. If the operation is “multiply” then deal it with “mulDiv”, else deal with the “processor” unit. The two hardware either fetch the data needed from Data Memory or calculate the correct answer. Then if the operation is not “bge”, “beq” or “sw”, write the result to the register rd. If the operation is “bge”, “beq”, “jalr” or “jal”, pass the mode to the PC\_control and PC\_control will decide the next instruction. If the operation is “sw”, the processor will write the result to Data Memory.

1. Descirbe how you handle multi-cycle instructions(mul)

When the ready signal in the module mulDiv is not rising up, the halt signal to PC\_contol is 1 and stops the PC from fetching the instructions.

1. Simulation time
2. Leaf:



1. Perm:



1. Bonus:

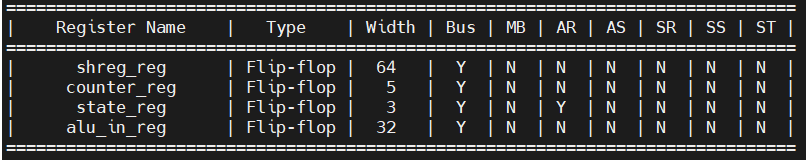


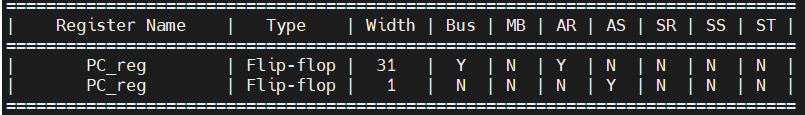
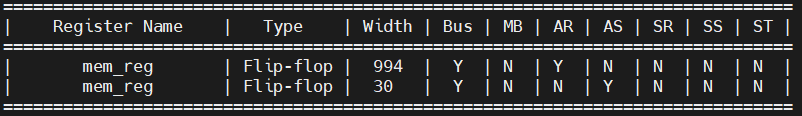
1. Our observation:

When we encounter a negative immediate, we should first convert it to its 2’s complement positive value and subtract it.

When writing the assembly code, we found that it’s important to use the stack pointer. Since we need to do recursive function calls, using the stack to store the value needed is quite convenient.

1. Register table





1. B09901019 劉瑄穎 : Assembly code of bonus and R-format and auipc and mulDiv module.

B09901047 陳以哲 : I-format and lw and sw and SB-format and JALR and JAL and PC-control module.