VLSI System Design (Graduate Level)

Fall 2021

HOMEWORK I

REPORT

Must do self-checking before submission:

Compress all files described in the problem into one tar

All SystemVerilog files can be compiled under SoC Lab environment

All port declarations comply with I/O port specifications

Organize files according to File Hierarchy Requirement

No any waveform files in deliverables

Student name: \_楊承翰\_

Student ID: \_NE6091124\_

RTL & Gate-level simulation:

RTL: Gate-level:

Program0:

Text

Description automatically generated Text

Description automatically generated with medium confidence

Program1:

Text

Description automatically generated Text

Description automatically generated

Program2:

Text

Description automatically generated Text

Description automatically generated with low confidence

Program3:

Text

Description automatically generated Text

Description automatically generated

Area report:

Text

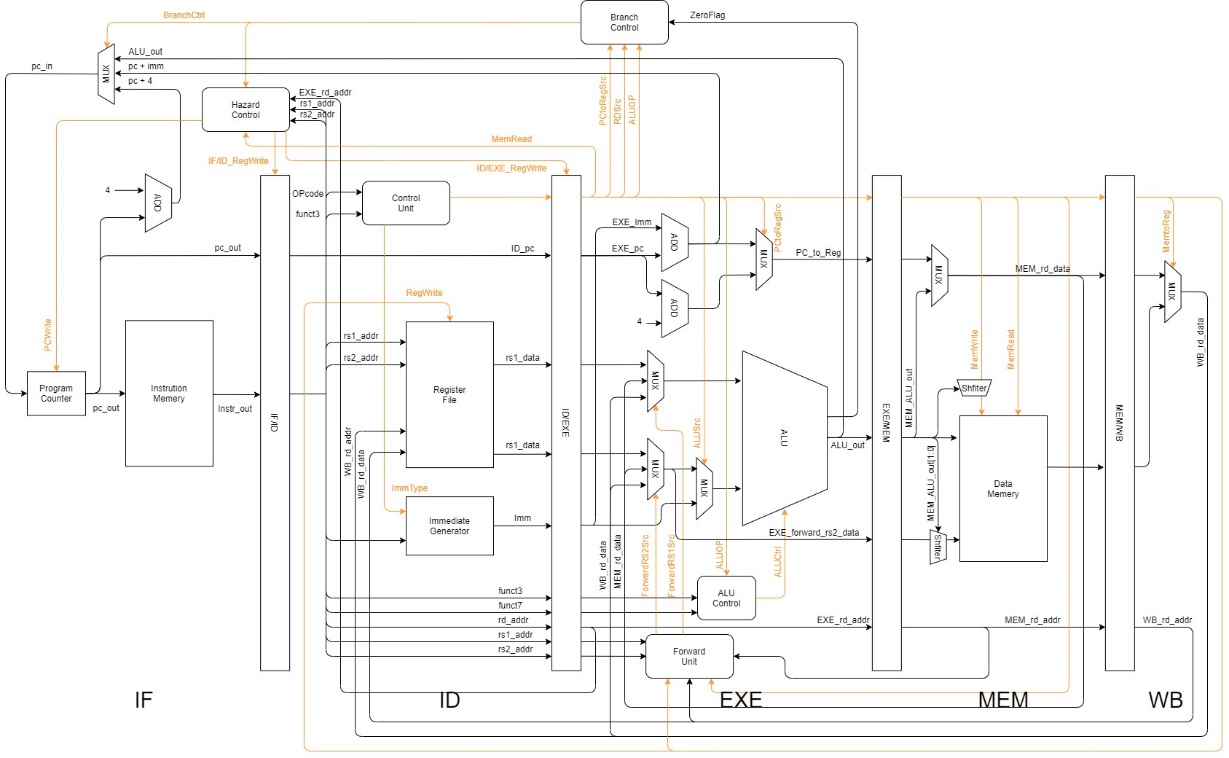
Description automatically generated

Timing report:

A picture containing text

Description automatically generated

Block Diagram:



Summary:

作業一是要完成RISC-V的5-stage pipeline CPU，上圖是我的block diagram，首先我按照以前課本所教的5-stage pipeline CPU將我的design切成IF、ID、EXE、MEM、WB五個階段，先規劃負責處理資料的module，如:Program Counter、Instruction Master、Register File、Immediate Generator、ALU & Data Master，再寫負責控制訊號的module，如:Control Unit、Branch Control、ALU Control、Forward Unit、Hazard Control，最後再用IF/ID、ID/EXE、EXE/MEM、MEM/WB四個Register切成五個stage。接下來我會一一介紹我每個module是怎麼完成的，以及每個指令的wave form。

Description of my design:

Program counter:

在rst = 1的時候會初始化pc\_out = 0，每個clk 正緣來的時候就會讓pc\_out = pc\_in，會根據Hazard Control傳過來的PCWrite訊號，正常情況下，PCWrite = 1，pc\_out會正常送出，並且發出InstrRead = 1訊號給Instruction Master，如果有load-use情況發生， PCWrite = 0，此時pc\_out會維持不變一個clk。因為向Instruction Memory及Data Memory讀取資料時，需要等一個clk的時間，所以在這段期間會需要暫停所有module一個clk，這時會收到一個stall的訊號，此時pc\_out會維持不變，而InstrRead = 0。

Instruction Master:

當收到InstrRead = 1的時候，會發給其他module一個stall的訊號，並將讀取資料的位址準備好傳給Instruction Memory，等到下一個clk，收到資料後，會將stall = 0，並將讀取到的Instruction傳給IF/ID Register。

Register file:

在rst = 1的時候會初始化x0 = 0，read data階段會根據收到的addr給出相對應的data，write data階段則是會在clk正緣時，根據RegWrite的值，將data用不同的方式(LW, LH, LB)存進Register File，另外如果有要改寫x0的話，會略過這個動作。

Immediate generator:

會根據Control Unit所傳過來的ImmType將Instruction轉換成對應的Immediate。

ALU:

會根據ALU Control傳來的ALUCtrl將兩個input作相對應的運算並輸出。

Data Master:

當收到MemRead的時候，會發給其他module一個stall的訊號，並將讀取資料的位址準備好傳給Data Memory，等到下一個clk，收到資料後，會將stall = 0，並將讀取到的Data傳給MEM/WB Register，而收到MemWrite時，一樣會發給其他module一個stall的訊號，並將資料的位址及資料傳給Data Memory，等到下個clk時，再將stall = 0。

Control unit:

會根據Instruction裡的opcode(Instruction[6:0])以及funct3(Instruction[14:12])，產生相對應的control signal。

Branch control:

當有branch發生時會改變BranchCtrl讓其他module做相對應的行為。

ALU control:

會根據ALUOP、funct3以及funct7來產生ALUCtrl，讓ALU做相對應的運算。

Forward unit:

會先判斷在MEM階段準備要在WB階段存回register file的資料是否剛好有要在EXE被使用，如果有則會發出訊號讓該值先拿回來做運算，接著判斷在WB階段存回register file的資料是否剛好有要在EXE被使用，如果有則會發出訊號讓該值先拿回來做運算。

Hazard control:

當load-use發生時，會讓program counter保持一個clk，而當branch發生時，則會使IF/ID register跟ID/EXE register裡面的值清成0。

Wave form:

R-type:

ADD:

Graphical user interface, text, application

Description automatically generated

在EXE階段PC = 0x130，instruction = 0x62823， funct3 = 0x0，funct7 = 0x0，ALU作用是ADD。

EXE\_Forward\_rs1\_data : 0xffffffff

EXE\_rs2\_data\_ALU : 0xffffffff

EXE\_ALU\_out : 0xfffffffe

SRA:

A screenshot of a computer

Description automatically generated

在EXE階段PC = 0x2ec，instruction = 0x4062d2b3， funct3 = 0x5，funct7 = 0x20，ALU作用是SRA。

EXE\_Forward\_rs1\_data : 0x87654321

EXE\_rs2\_data\_ALU : 0x4

EXE\_ALU\_out : 0xf8765432

AND:

Graphical user interface, text

Description automatically generated

在EXE階段PC = 0x378，instruction = 0x62f2b3， funct3 = 0x7，funct7 = 0x0，ALU作用是AND。

EXE\_Forward\_rs1\_data : 0x12345678

EXE\_rs2\_data\_ALU : 0xffffffff

EXE\_ALU\_out : 0x12345678

I-type:

LW:

Graphical user interface

Description automatically generated

在EXE階段PC = 0x400，Instruction = 0xffc12303，ALU作用是ADD，下一個clk時把ALU加起來的結果0x90f8送到Data Memory，並將MemRead訊號改成1，下一個clk時資料回傳，要存的值是0x90f4，下一個clk將data傳給Register File並將RegWrite改成1，下一個clk時Register File 裡面的值即更新完成。

XORI:

Graphical user interface

Description automatically generated

在EXE階段PC = 0x540，Instruction = 0x7b2c293，ALU作用是XOR。之後把ALU加起來的結果0xffffff84送到Register File並將RegWrite改成1，下一個clk時Register File 裡面的值即更新完成。

JALR:

Graphical user interface

Description automatically generated

在EXE階段PC = 0x72c，Instruction = 0x30367， Control Unit會先根據opcode使ALUOP = 0x5，ALU作用是ADD，Branch Control根據ALUOP判斷出有branch發生，將BranchCtrl訊號改成0x2，讓pc\_in透過Mux選到ALU加起來的值0x744，Hazard control也會將IF\_ID\_RegWrite及ID\_EXE\_RegWrite兩個控制訊號改成0，使IF/ID Register及ID/EXE Register裡面所有的值清成0，下一個clk，PC = 0x744。而要存進Register File的值，Control unit根據opcode讓PCtoRegSrc = 0，使得EXE\_PC\_to\_Reg = EXE\_PC\_plus\_4，讓RDSrc = 1，使得在MEM階段 MEM\_rd\_data = MEM\_PC\_to\_Reg，MemtoReg = 0，RegWrite = 1，在WB階段將值0x730存入Register File。

S-type:

SW:

Graphical user interface

Description automatically generated

在EXE階段PC = 0x7c4，Instruction = 0xfe542e23，funct3 = 0x2，下一個clk會把ALU算出的結果0x8074送到data memory確定位址，並將EXE階段得到的rs2\_data的值0xf傳到data\_in，還有將MemWrite訊號改成0x0。

SB:

Graphical user interface, text

Description automatically generated

在EXE階段PC = 0x7f0，Instruction = 0xffe409a3，funct3 = 0x0，下一個clk會把ALU算出的結果0x807f送到data memory確定位址，並將EXE階段得到的rs2\_data的值0x12345678根據EXE\_ALU\_out[1:0]做left-shift，而此時EXE\_ALU\_out[1:0] = 0x3，所以rs2\_data的值0x12345678會左移24bits再傳到data\_in，MemWrite訊號也會改成0x7。

SH:

Text

Description automatically generated

在EXE階段PC = 0x7f4，Instruction = 0xffe41723，funct3 = 0x1，下一個clk會把ALU算出的結果0x807a送到data memory確定位址，並將EXE階段得到的rs2\_data的值0x12345678根據EXE\_ALU\_out[1:0]做left-shift，而此時EXE\_ALU\_out[1:0] = 0x2，所以rs2\_data的值0x12345678會左移16bits再傳到data\_in，MemWrite訊號也會改成0x3

B-type:

BEQ:

Graphical user interface

Description automatically generated

在EXE階段PC = 0x85c，Instruction = 0x628463，funct3 = 0x0，Control unit根據opcode送出ALUOP = 0x3，ALU會去判斷EXE\_Forward\_rs1\_data是否等於EXE\_rs2\_data\_ALU，這邊兩個相等，所以zeroflag = 1，而Branch control收到zeroflag = 1以及ALUOP = 0x3會使BranchCtrl = 1，讓pc\_in透過Mux選EXE\_PC\_plus\_Imm的值0x864，Hazard control也會將IF\_ID\_RegWrite及ID\_EXE\_RegWrite兩個控制訊號改成0，使IF/ID register及ID/EXE register裡面所有的值清成0，再下一個clk時PC改成0x864。

BLT:

A screenshot of a computer

Description automatically generated

在EXE階段PC = 0x920，Instruction = 0x62c463，funct3 = 0x4，Control unit根據opcode送出ALUOP = 0x3，ALU會去判斷EXE\_Forward\_rs1\_data是否小於EXE\_rs2\_data\_ALU，這邊成立，所以zeroflag = 1，而Branch control收到zeroflag = 1以及ALUOP = 0x3會使BranchCtrl = 1，讓pc\_in透過Mux選EXE\_pc\_plus\_imm的值0x928，Hazard control也會將IF\_ID\_RegWrite及ID\_EXE\_RegWrite兩個控制訊號改成0，使IF/ID register及ID/EXE register裡面所有的值清成0，下一個clk時PC改成0x928。

BGEU:

A screenshot of a computer

Description automatically generated with medium confidence

在EXE階段PC = 0xaa8，Instruction = 0x62f463，funct3 = 0xf，Control unit根據opcode送出ALUOP = 0x3，ALU會去判斷EXE\_Forward\_rs1\_data是否大於等於EXE\_rs2\_data\_ALU，這邊成立，所以zeroflag = 1，而Branch control收到zeroflag = 1以及ALUOP = 0x3會使BranchCtrl = 1，讓pc\_in透過Mux選EXE\_pc\_plus\_imm的值0xab0，Hazard control也會將IF\_ID\_RegWrite及ID\_EXE\_RegWrite兩個控制訊號改成0，使IF/ID register及ID/EXE register裡面所有的值清成0，下一個clk時program counter改成0xab0。

U-type:

AUIPC:

Graphical user interface

Description automatically generated

在EXE階段PC = 0xac8，Instruction = 0x297， Control unit根據opcode讓PCtoRegSrc = 1，使得EXE\_PC\_to\_Reg = EXE\_PC\_plus\_Imm，讓RDSrc = 1，使得在MEM階段 MEM\_rd\_data = MEM\_PC\_to\_Reg，MemtoReg = 0，RegWrite = 1，在WB階段將值0xac8存入Register File。

LUI:

Graphical user interface

Description automatically generated

在EXE階段PC = 0xae8，Instruction = 0xfffff337， ALU會將EXE\_rs2\_data\_ALU的值0xfffff000直接assign給EXE\_ALU\_out，在WB階段將值存入Register File。

J-type:

JAL:

Graphical user interface

Description automatically generated

在EXE階段PC = 0xb04，Instruction = 0x80036f，Control unit根據opcode讓PCtoRegSrc = 0，使得EXE\_PC\_to\_Reg = EXE\_PC\_plus\_4，讓RDSrc = 1，使得在MEM階段 MEM\_rd\_data = MEM\_PC\_to\_Reg，MemtoReg = 0，RegWrite = 1，在WB階段將值0xb08存入Register File。

另外Branch Control會根據EXE\_PCtoRegSrc以及EXE\_RDSrc判斷出有branch發生，將BranchCtrl訊號改成0x1，讓pc\_in透過Mux選到EXE\_PC\_plus\_Imm的值0xb0c，下個clk，PC = 0xb0c。

Lessons learned:

之前都只在課本讀到相關知識，自己實際寫出來後，更清楚理解每個stage之間的資料交換，當有Forwarding發生時，要按照順序先判斷MEM階段的資料再判斷WB階段的資料。而Hazard發生時，則要先確認是branch還是load-use來決定是否要清除原本暫存的資料。