Final project report

工海三 B09505029 羅士朋

電機二 B10901008 張禾牧

Briefly describe your CPU architecture

CPU為non-pipline的架構，利用state(當前執行指令)與sub-state(執行階段)控制；在ID時，會將指令解析，並在下一個cycle進入EX sub-state。在EX時，會計算R-type等數值(類似ALU，但不包括branch)，除了MUL之外都會在下一個cycle進入WB sub-state。在WB時，會執行所有register的寫入，而最後則會更新PC，讀入指令進入下個迴圈。

Describe how you design the data path of instructions not referred in the lecture slides (jal, jalr, auipc, ...)

jal和jalr首先會在EX階段計算rd的值(PC+4)，與其他指令一樣一同在WB階段寫入，並在PC更新的階段，及時計算跳躍位置，並取代原本的PC更新。

auipc跟一般指令類似，同樣在EX計算rd的值(PC+{imm,12'd0})，並在WB寫入

Describe how you handle multi-cycle instructions (mul)

對於MUL，會利用mulWaiting，在初次進入EX時切為1，阻止substate的更新，直至ready為1時，再進入下個(WB)state。

Describe your observation

設計 data path 時，若要盡量節省時間資源(e.g. 讓 branch 指令執行較少 stages)，就會使整體架構變得複雜許多，要考慮的問題也會大幅增加。

如果用一般高階軟體語言的思維去寫 verilog 的話，很容易增加許多不必要的變數，且為了處理 latch 等問題，程式甚至會變得更加複雜。

將各功能分成 module 能增加 debug 的容易度，尤其是像 controller 等重要控制架構，避免與其他程式碼混合也能讓結構更加簡潔明瞭。

- Record total simulation time (CYCLE = 10 ns)

✓ leaf

```
Simulation complete via $finish(1) at time 775 NS + 0
```

✓ perm

```
Simulation complete via $finish(1) at time 4735 NS + 0
```

- Snapshot the “Register table” in Design Compiler (p. 22)

```
============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
============================================================================
|     alu_in_reg      | Flip-flop |   32  |  Y  |  N |  Y |  N |  N |  N |  N |
|     counter_reg     | Flip-flop |    5  |  Y  |  N |  Y |  N |  N |  N |  N |
|      state_reg      | Flip-flop |    3  |  Y  |  N |  Y |  N |  N |  N |  N |
|      shreg_reg      | Flip-flop |   64  |  Y  |  N |  Y |  N |  N |  N |  N |
============================================================================
```

```
============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
============================================================================
|       mem_reg       | Flip-flop |  994  |  Y  |  N |  Y |  N |  N |  N |  N |
|       mem_reg       | Flip-flop |   30  |  Y  |  N |  N |  Y |  N |  N |  N |
============================================================================
```

```
============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
============================================================================
|    sub_state_reg    | Flip-flop |    3  |  Y  |  N |  Y |  N |  N |  N |  N |
|    readInstr_reg    | Flip-flop |    1  |  N  |  N |  N |  Y |  N |  N |  N |
|       PC_reg        | Flip-flop |   31  |  Y  |  N |  Y |  N |  N |  N |  N |
|       PC_reg        | Flip-flop |    1  |  N  |  N |  N |  Y |  N |  N |  N |
|      state_reg      | Flip-flop |    5  |  Y  |  N |  Y |  N |  N |  N |  N |
============================================================================
```

- List a work distribution table

工作內容	B09505029 羅士朋	B10901008 張禾牧
verilog編寫與除錯	50%	50%
報告編寫	50%	50%