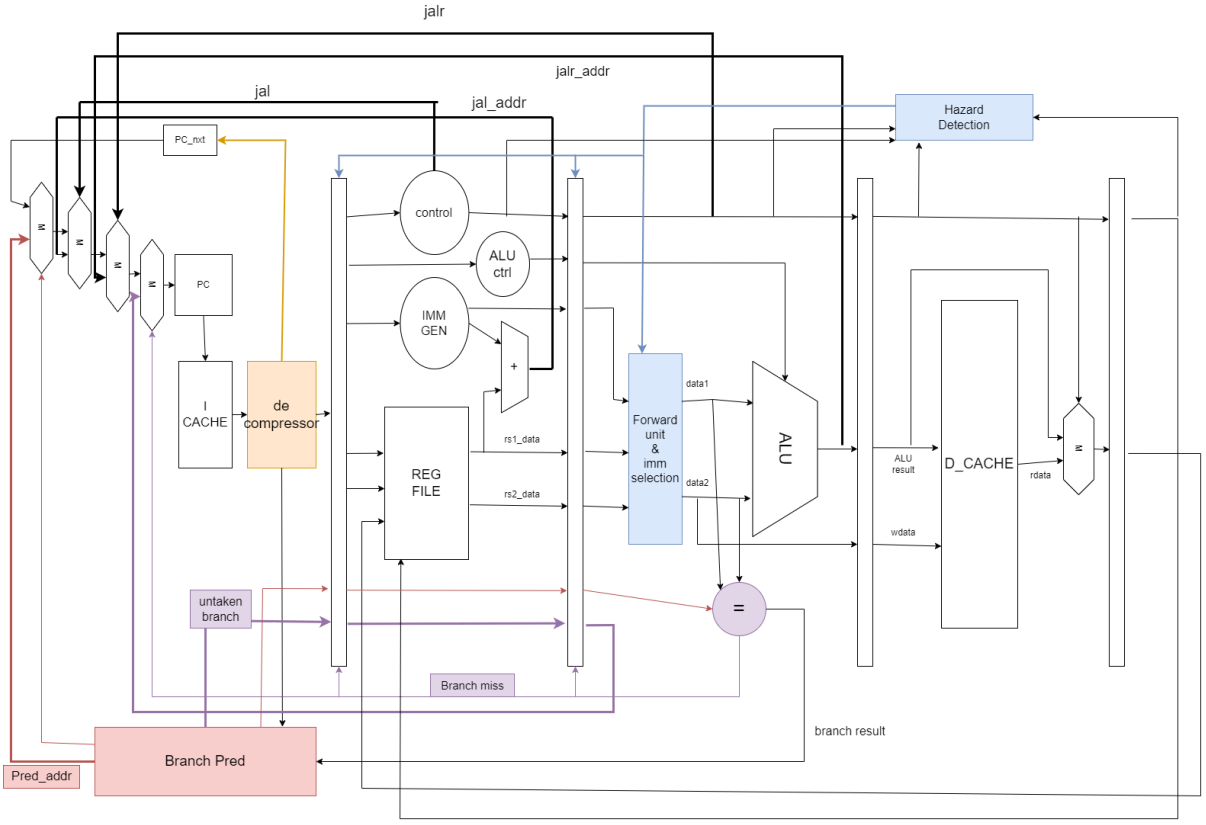
數位系統設計 final project report

第十四組 組員:顏柏聖、曾啟睿、莊昊晨

一、Verilog Program Architecture

1. Overall Architecture Diagram



a. IF satge:
用 I cache 把 PC 位置的 instruction 從 I memory 提取出來，放進 decompression 來解壓縮後給下一個 stage 與 Branch prediction block (後來已移除)。

b. ID stage:
把 instruction 處理成 imm,control signal…等訊號
把 reg file 中的資料提取出來
把 jal, branch 的位置計算完 (移除Branch prediction block後新增)。

c. EX stage:
進行 Forwarding，然後進行 ALU 計算 (R type,I type, jalr)
進行branch miss的判斷。

d. MEM stage
   對 D cache 操作，同上課投影片。

e. WB stage
   將資料寫回 reg file，同上課投影片。

2. Block Detail

a. PC 運作：
   (1). stall / hazard: 將 PC 暫停
   (2). branch miss: 將 PC 更新成 branch 應該去的位置，由 EX stage 提供
   (3). EX stage jalr 發生: 將 PC 更新成 jalr 目標
   (4). ID stage jal 發生: 將 PC 洗成 jal 目標
   (5). 以上情況皆非: 由 compression 模組決定該前往 PC+0/PC+2/PC+4
   由多層MUX來控制以上情況的優先度: 1 -> 5 (多種情況同時發生的話越後面的 stage 發生的事越優先)

b. JAL/JALR/BRANCH 位置重分配：
   將以上分支指令從原本的 MEMstage 分配至 ID/EX stage，以減少 cycle 數。
   JAL: 若放在 IFstage 的話，再加上 decompression 會導致路徑太長，故目標地址的計算放在 ID
   JALR: 因為需要使用到 reg file 的資料來做運算，為減小路徑與面積，重複使用 EX 的 ALU 來運算
   BRANCH: 由於 branch prediction 後來被移除，為節省面積可重複使用 ID stage JAL 的加法器來計算目標地址

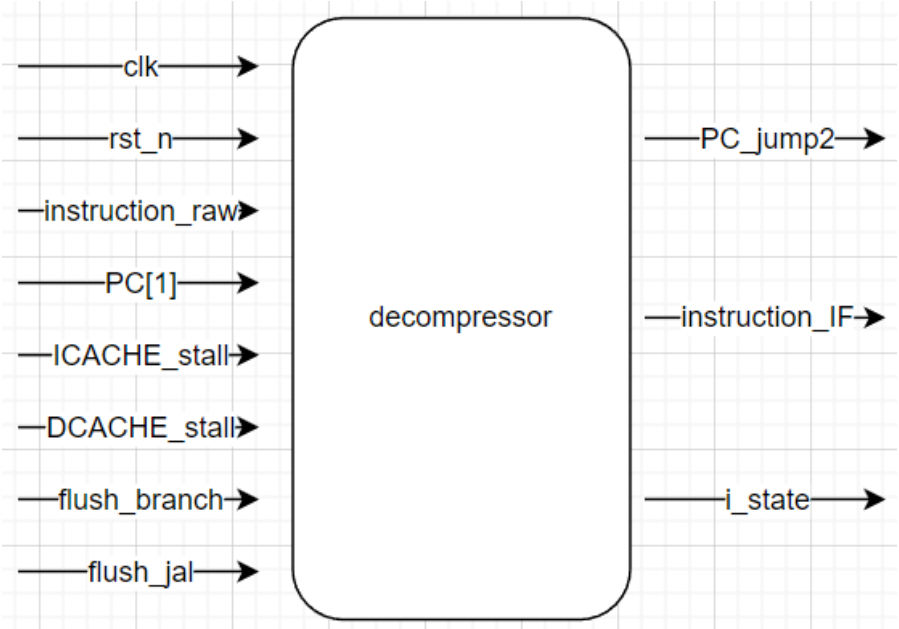
c. ALU
   可將 SLT 簡化成減法後檢查正負號。

d. branch miss
   在 EX stage 比較 branch 的預測結果是否正確，若正確則不做任何改變，若錯誤則發出branch miss訊號，洗掉前兩個 stage，改變 PC 值

e. hazard dectection
   同上課投影片的方式，偵測不同 stage 的訊號來判斷是否發生 hazard 來輸出 stall/forward訊號

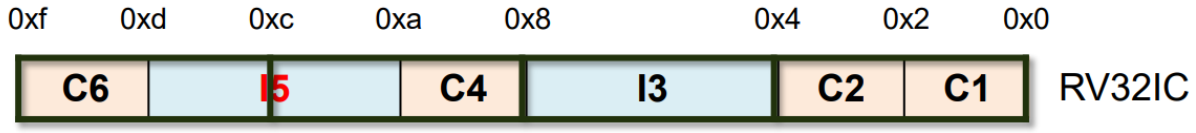
f. 其他 block 皆與上課投影片所描述的方法相同

3. Compression



compression 的好處在於可以在同樣的 cache 空間中存入較多的指令，可有效降低 miss rate 的同時不會增加 miss penalty，因為 memory acess 的次數減少故能提高 throughput，可以發現在跑 compress 指令需要花費的 cycle 數較原本的少

a. 透過 PC[1] 判斷此時 PC 的位置是在 block 中間 (ex:0x8) 還是邊界處 (ex:0xc)



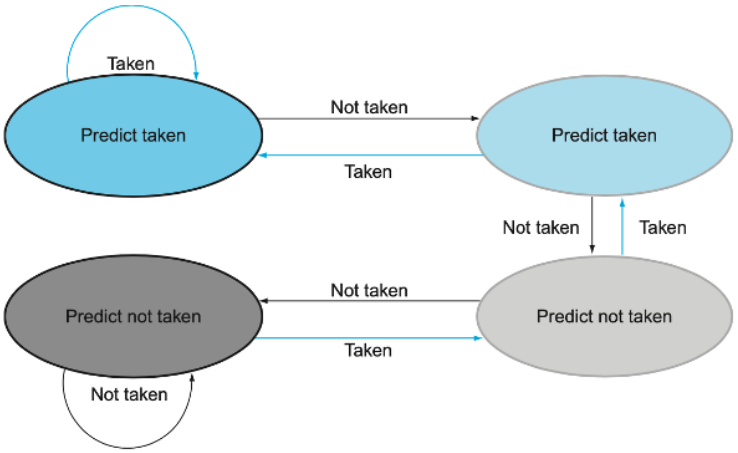
b. 依據情況不同 (如上圖C1(PC+2), C2(PC+2), I3(PC+4), I5(PC+0) 皆為不同情況) 判斷 PC 的新位置 (+0, +2, +4) 並輸出 decompress 完的 instruction 到 IF stage
c. 當出現如上I5的情況時如上圖(0xa~0xc)的內容存入buffer並將i_state輸出1，此時ICACHE會先在往後PC+4截取下一段指令再送入decompression中與buffer內存有的指令concatenate
d. 當遇到stall時buffer內的內容保持不變(不會被洗掉)，遇到flush時則清空buffer以及將i_state歸零，以免前後指令接錯的情況發生

4. Branch Prediction

branch prediction 的目標是當執行 beq 指令時，提前預測出是否會執行 branch，預期能藉由提升預測準確率來減少 branch miss 的發生，進一步減少整個程式碼模擬時需要執行的總 cycle 數同時減少 simulation time。

我們為此設計了 2-bit predictor，為一個由 4 個 state 組成的 finite state machine (如下圖)，每次在 IF stage 時根據其所處的 state 決定下一個指令是否

要跳到 branch 的地址 (假設為 beq 指令)，然後在 EX stage 時會先判斷是否執行為 beq 指令，若否，則 state 不改變；若是的話，則再依據實際是否需要 branch 來修改 state。而當發生 branch miss 時，先將目前執行的指令通通暫停並將傳遞中的資料歸零，回到上次的 beq 指令，並換成正確的下一個指令繼續運作。



在比較模擬結果時，我們發現 always not taken 反而會比使用 2-bit predictor，探討後發現原因出在於 2-bit predictor 所運作的那條線路剛好就是 critical path，而若只用 always not taken，由於不需要額外多跑兩個 mux，因此可以省下不少的 cycle time，因此即使 2-bit predictor 可以降低總 cycle 數量，仍然是不划算的。

此外，我們也在撰寫程式碼的過程中學習到了另外幾種 branch prediction 的運作原理，比如 2-level prediction 是會參考過去的幾筆 beq 指令 (n = 2 ~ 3) 而推測出接下來的beq 指令是否執行 branch，當執行的程式具有一定的規律時，這個方法能更有效的提升預測準確率，進一步降低總執行 cycle 的數量，然而從別的組別實作的經驗大致都顯示如此並不會比直接 always not taken 還要有更好的 AT performance，因此我們最後把優化的心力放在別的部分。

5. Optimization

a. remove branch prediction: 我們發現若加上 branch prediction 就必須要先計算出 branch 目標位置，多新增的加法器與控制訊號會使 IF stage 的 critical path 增加非常多，而在這個測資下 2-bit prediction 的 cycle 效益卻跟 always not taken 比並沒有非常多。故我們後來採取了 always not taken 也就是不做branch prediction，把 branch 位置計算後移到 ID，雖然使 cycle 數增加，但critical path 減少可達約 1ns 之多，面積也減少了，AT 值相比之下還是賺了不少。

b. read-only I cache: 由於 I cache 不需要 write 的操作，因此我們將這部分的功能移除以減少面積

c. D cache write buffer: 我們將 D-cache 要寫入 memory 的資訊存在 write buffer 內，D cache 在進行不需要 memory access 的操作時便可利用此段時間從 write buffer 寫入 memory，有效增加 throughput。

d. 由於 memory 是由負源讀取資料，為了避免 cache 在正源開始計算後來不及在負源讀取前穩定足夠長的時間而出現 setup time violation，因此在計算完成後推遲了一個 cycle 進行讀取，以確保不會出現上述的問題，雖然增加了 cycle 數量，但有效壓低了 tb 的 setup violation 出現

二、Comparison

1. I-cache read-only vs. I-cache write & read

想法：由於在這次的 pipeline CPU design 作業中，不需要寫入任何新的東西到 instruction memory 內，因此我們將原本能執行 read & write 的 I-cache 進行適當的刪減，使其變成僅能執行讀取的功能，如此的做法可以有效降低面積。

模擬結果：

a. read-only:
   .sdc cycle = 2.9 ns, tb cycle = 2.9 ns
   => Total cell area = 278124.078927 (um^2)
   => Simulation time = 431248.85 ns
   => A*T = 1.199E11

b. read & write:
   .sdc cycle = 2.9 ns, tb cycle = 2.9 ns
   => Total cell area = 291376.029387 (um^2)
   => Simulation time = 436475.5 ns
   => A*T = 1.272E11

2. branch prediction: always not taken vs. 2-bit prediction

想法：對於 beq 指令而言，初始時我們並不知道下一個指令會是需要 branch 的指令還是單純的 PC+4，因此在寫成 pipeline 形式時，我們的做法如下：

a. always not taken: 若執行 beq 指令，在 IF stage 時先假設 PC_nxt = PC + 4，接著在 ID stage 計算出 branch 的地址，並在 EX stage 檢驗實際是否應該執行 branch，若不需執行則程式繼續運作；若需執行則程式暫停並回到前一個 beq 指令重新執行 branch 到的指令。
b. 2-bit predition: 若執行 beq 指令，在 IF stage 時先把 branch 指到的地址算出來並利用 finite state machine 預測是否是否需要 branch，最後在 EX stage 實際判斷是否應該 branch 時，再傳回 IF 檢查。

模擬結果:

a. always not taken:
   .sdc cycle = 2.9 ns, tb cycle = 2.9 ns
   => Total cell area = 278124.078927 (um^2)
   => Simulation time = 431248.85 ns
   => A*T = 1.199E11

b. 2-bit predictor:
   .sdc cycle = 3.8 ns, tb cycle = 3.8 ns
   => Total cell area = 263180.169507 (um^2)
   => Simulation time = 509179.1 ns
   => A*T = 1.340E11

三、Simulation Result

1. AT performance from Q_sort

(no time violation & tb_v2)
RTL:

```
--------------------------------------------------

START!!! Simulation Start .....

--------------------------------------------------

---------------------------- Simulation FINISH !!----------------------------
================================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

================================================================================
$finish called from file "Final_tb.v", line 143.
$finish at simulation time             48269850
           V C S   S i m u l a t i o n   R e p o r t
Time: 482698500 ps
CPU Time:     42.880 seconds;       Data structure size:   0.3Mb
Sat Jun 17 14:17:41 2023
CPU time: .934 seconds to compile + .661 seconds to elab + .576 seconds to link + 42.940 seconds in simulation
```

SYN: (.sdc cycle = 2.9 ns, tb cycle = 2.9 ns)

```
Number of ports:                          1557
Number of nets:                          21014
Number of cells:                         18942
Number of combinational cells:           14537
Number of sequential cells:               4268
Number of macros/black boxes:                0
Number of buf/inv:                        3583
Number of references:                      151

Combinational area:             147234.172982
Buf/Inv area:                    25264.101541
Noncombinational area:          130889.905945
Macro/Black Box area:                0.000000
Net Interconnect area:         2333251.917267

Total cell area:                278124.078927
Total area:                    2611375.996193
```

```
--------------------------------------------------

START!!! Simulation Start .....

--------------------------------------------------

--------------------------- Simulation FINISH !!---------------------------
==============================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

==============================================================================
$finish called from file "Final_tb_v2.v", line 144.
$finish at simulation time            431248850
           V C S   S i m u l a t i o n   R e p o r t
Time: 431248850 ps
CPU Time:     50.200 seconds;       Data structure size:   5.1Mb
Sat Jun 17 22:18:52 2023
CPU time: 3.974 seconds to compile + 1.400 seconds to elab + 1.441 seconds to link + 50.249 seconds in simulation
```

=> Total cell area = 278124.078927 um^2

=> Simulation time = 431248.85 ns

=> **A*T = 1.199E11**

(no time violation & tb_v1)

RTL:

```
--------------------------------------------------

START!!! Simulation Start .....

--------------------------------------------------

--------------------------- Simulation FINISH !!---------------------------
==============================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

==============================================================================
$finish called from file "Final_tb.v", line 143.
$finish at simulation time             55595710
           V C S   S i m u l a t i o n   R e p o r t
Time: 555957100 ps
CPU Time:     42.430 seconds;       Data structure size:   0.3Mb
Fri Jun 16 13:17:14 2023
CPU time: .821 seconds to compile + .603 seconds to elab + .604 seconds to link + 42.472 seconds in simulation
```

SYN: (.sdc cycle = 3.1 ns, tb cycle = 3.1 ns)

```
Number of ports:                          1557
Number of nets:                          19261
Number of cells:                         17135
Number of combinational cells:           12730
Number of sequential cells:               4268
Number of macros/black boxes:                0
Number of buf/inv:                        1871
Number of references:                      149

Combinational area:             134242.272849
Buf/Inv area:                    16780.496433
Noncombinational area:          124892.992018
Macro/Black Box area:                0.000000
Net Interconnect area:         2268478.259064

Total cell area:                259135.264867
Total area:                    2527613.523931
```

```
*Verdi* : Enable +mda dumping.
*Verdi* : End of traversing.
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.
--------------------------------------------------

START!!! Simulation Start .....

--------------------------------------------------

------------------------- Simulation FINISH !!-------------------------
==============================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

==============================================================================
$finish called from file "Final_tb.v", line 143.
$finish at simulation time           450983350
          V C S   S i m u l a t i o n   R e p o r t
Time: 450983350 ps
```

=> Total cell area = 259135.264867 um^2
=> Simulation time = 450983.35 ns
=> A*T = 1.169E11

2. Compression

(no setup time violation)
tb cycle = 3.7 ns

```
--------------------------- Simulation FINISH !!---------------------------
==============================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

==============================================================================
$finish called from file "Final_tb.v", line 159.
$finish at simulation time              1962850
          V C S   S i m u l a t i o n   R e p o r t
Time: 1962850 ps
```

(with setup time violation)
tb cycle = 3.1 ns

```
--------------------------- Simulation FINISH !!---------------------------
================================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

================================================================================
$finish called from file "Final_tb.v", line 159.
$finish at simulation time              1645500
          V C S   S i m u l a t i o n   R e p o r t
Time: 1645500 ps
```

3. BrPred

(no setup time violation)

tb cycle = 3.7 ns

```
Branch Part A is complete.

Branch Part B is complete.

Branch Part C is complete.

--------------------------- Simulation FINISH !!---------------------------
================================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

================================================================================
$finish called from file "Final_tb.v", line 159.
$finish at simulation time              1666850
          V C S   S i m u l a t i o n   R e p o r t
Time: 1666850 ps
```

(with setup time violation)

tb cycle = 3 ns

```
Branch Part A is complete.

Branch Part B is complete.

Branch Part C is complete.

--------------------------- Simulation FINISH !!---------------------------
================================================================================

 \(^o^)/ CONGRATULATIONS!!  The simulation result is PASS!!!

================================================================================
$finish called from file "Final_tb.v", line 159.
$finish at simulation time              1399500
          V C S   S i m u l a t i o n   R e p o r t
Time: 1399500 ps
```