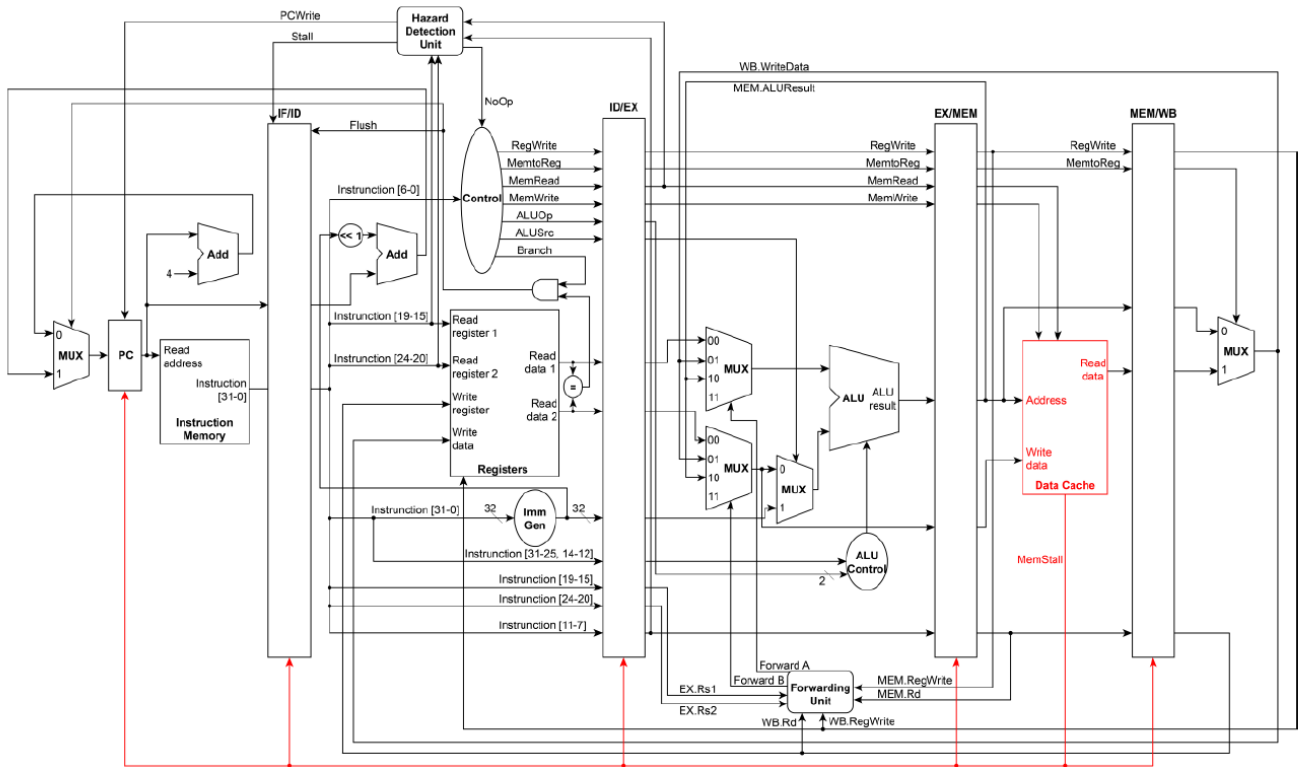


# Computer Architecture-Project 2

b06209027 李冠勳

## Structure



## Development environment

- linux
- iverilog
- gtkwave

## Implement each module

### dcache controller

- 首先要先理解助教的code
- 理解 state 的變化
  - IDLE
  - IDLE ⇒ MISS ⇒ READMISS ⇒ READMISSOK ⇒ IDLE
  - IDLE ⇒ MISS ⇒ WRITEBACK ⇒ READMISS ⇒ READMISSOK ⇒ IDLE
- 以上三種狀況分別代表1. hit, 2. not hit, old block not dirty 3. not hit, old block dirty
  - 範例測資中只給了前兩種情況，第三種情況要自行驗證

- write hit 是先read hit然後再將要write的data寫入cache
- 使用write back policy

	hit	miss
read	Fetch data from cache	Evict a block by LRU policy. Then bring the data from data memory into the cache.
write	Write only to the cache, and set up the dirty bit.	Evict a block by LRU policy. Then bring the data from data memory into the cache. Write to the cache, and set up the dirty bit.

### dcache sram

- 同步和非同步的 data 要分開來考慮
- output 的 data 是非同步的，但 cache 和 cache tag 是同步的
- 有一些 controller 需要用到 tag, hit 的 dependency 要加以判斷

## Difficulties & Problems

- 許多project 1沒有處理的問題在project 2中大量浮現導致要debug很久
- 範例測資並沒有測到 write back 的情況
- 許多 array 的資料不易debug
- 沒有使用波形模擬會陷入同步非同步的陷阱
- testbench 中的 initial memory data 有缺漏
- testbench flush cache 是固定的導致 2-way cache 的寫法被限制，應該要說明用何種方式實作