

ICLAB Final Project Report

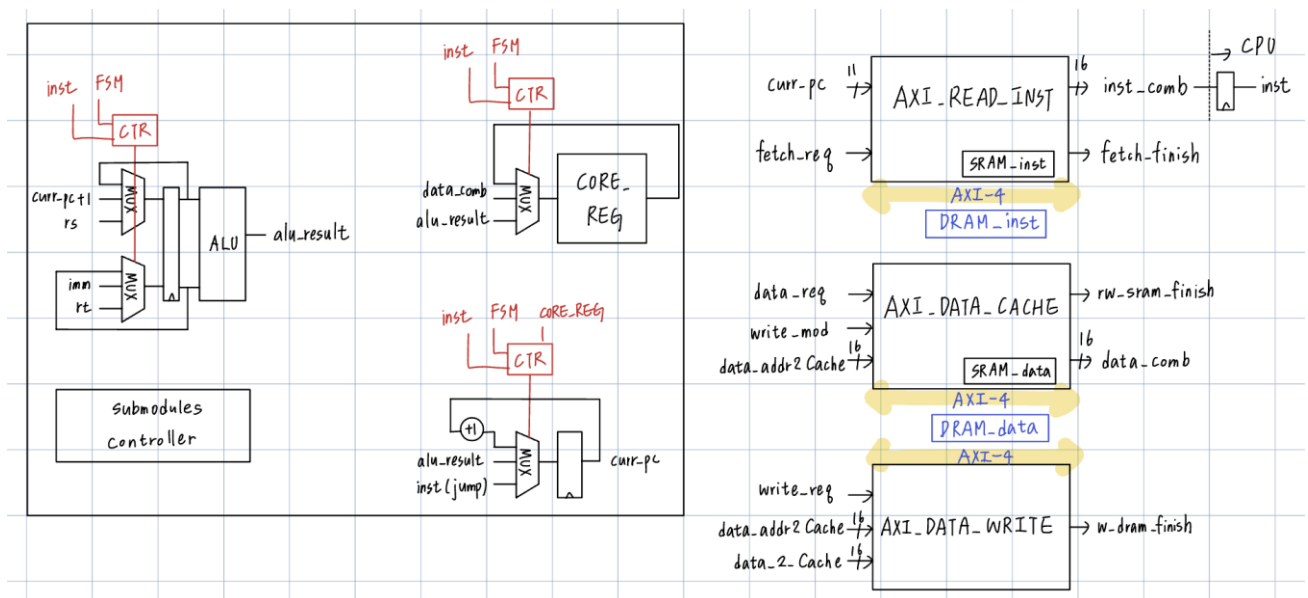
Single Core Central Processing Unit (CPU)

312591003 呂昀庭

一、 電路架構

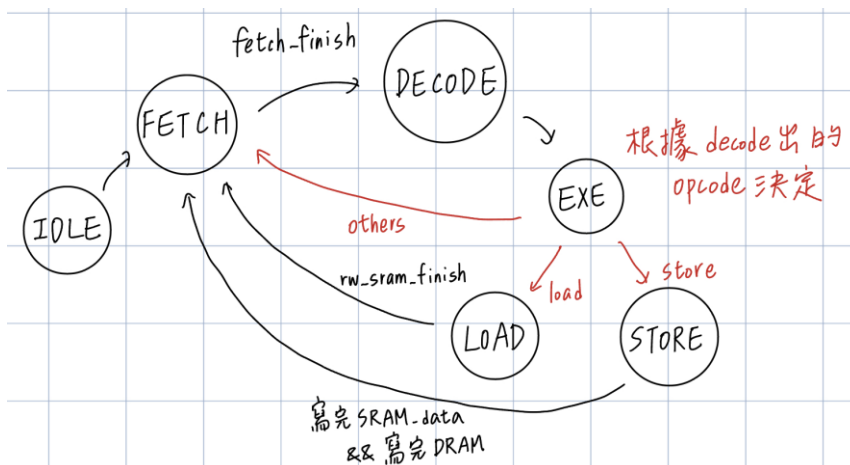
此次設計使用 multi-cycle 的做法，將 design 分為一個主要 module (CPU)及三個子 module (AXI_READ_INST, AXI_DATA_CACHE, AXI_DATA_WRITE)，使 code 更加清晰易懂。原始架構參考了 GitHub 上 mirkat1206 學姊的設計，並以此為基礎進行優化。

AXI_READ_INST 負責 instruction fetch，有一個 SRAM_inst (128w, 17b)作為 cache 使用，當 miss 時則須與 DRAM_inst 溝通。AXI_DATA_CACHE 則是負責存取 data，同樣有一個 SRAM_data (128w, 17b)作為 cache，當 miss 時則與 DRAM_data 溝通(read channel)。AXI_DATA_WRITE 則是在 STORE 指令時將更新後的 data 寫回 DRAM_data 中(write channel)。

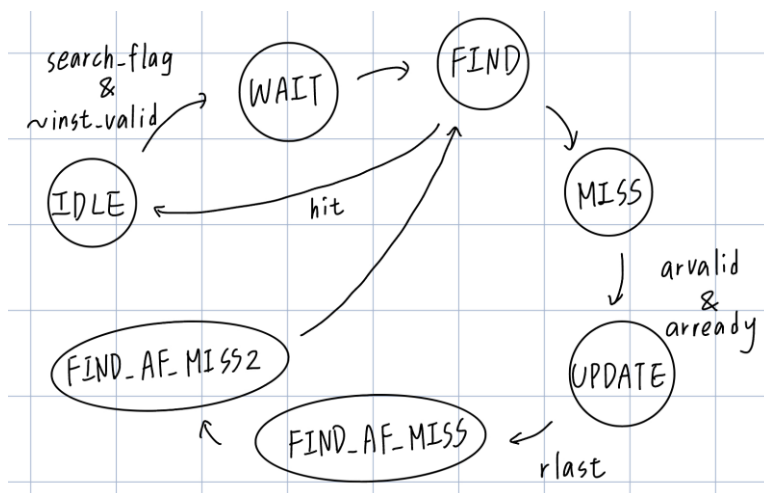


二、 FSM

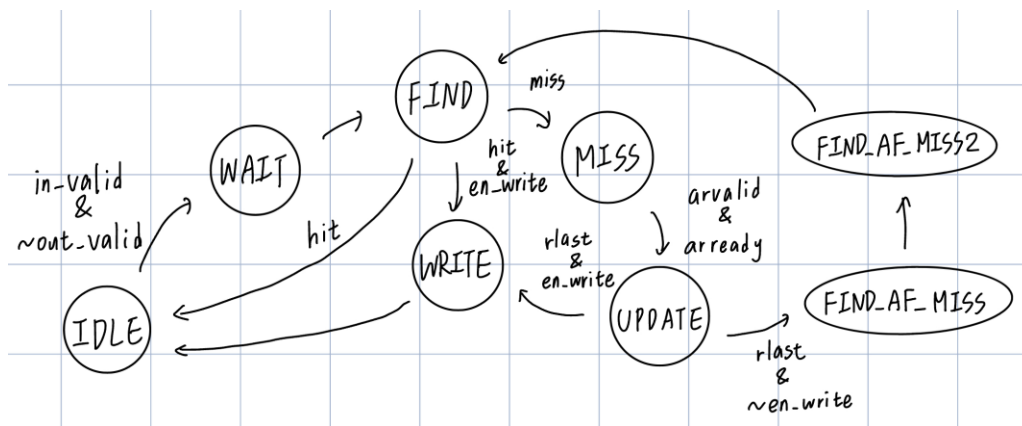
CPU:



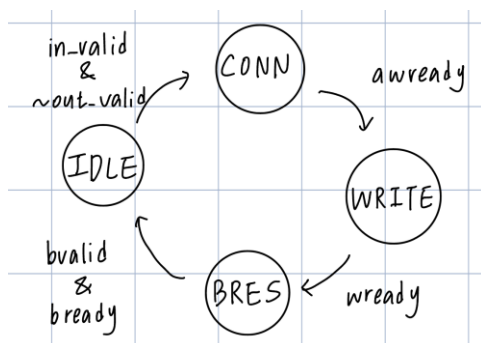
AXI_READ_INST:



AXI_DATA_CACHE:



AXI_DATA_WRITE:



三、 優化方法

實際上不論是 inst 還是 data 都只有 16 bits，但是我的 cache 多開了 1 bit 來當作 tag 使用。Mirkat 學姊的設計是不論 curr_pc 指向哪裡，都是將尾數為 0~127 筆的資料存進 cache 當中。但是由於 data dependence 是在 curr_pc 加減 64 的範圍，若剛好 curr_pc 指得很靠近邊界，下一次找的時候就很容易 cache miss。因此我最大化利用 dependency 的特性，希望增加 hit 的機率。不過這樣就需要注意 DRAM address 的有效範圍，須小心處理邊界問題。另外因為 tag 是存在 SRAM 中，要多花 1 cycle 將 tag 讀出才能知道 cache 中的資料是否為正確的。