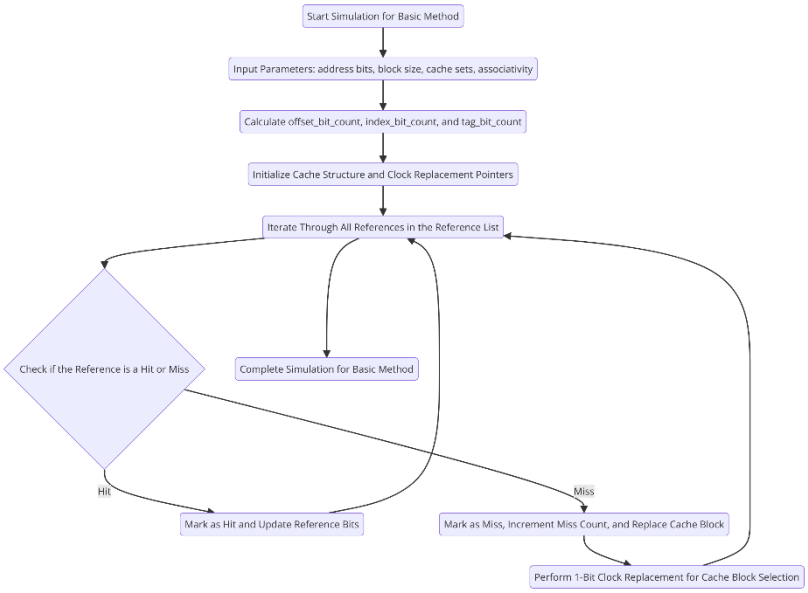
計算機結構 Final Project Report

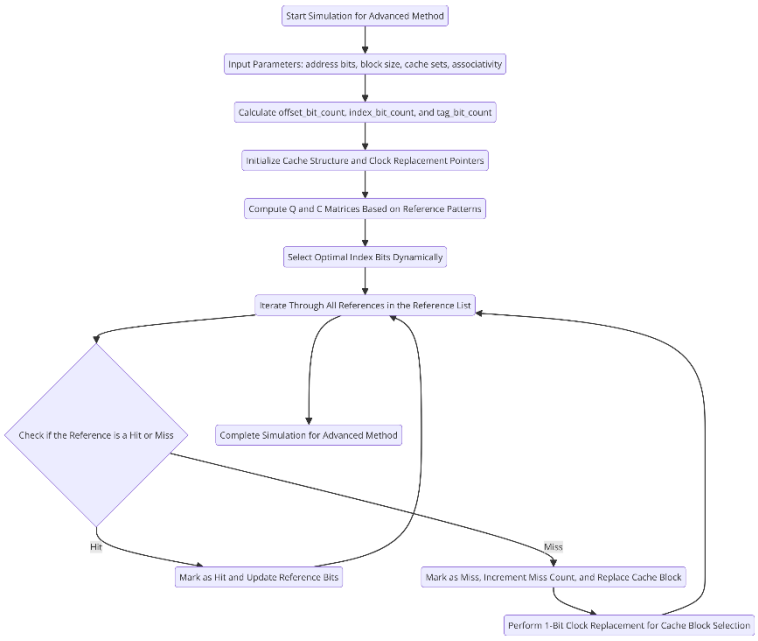
資應所二 吳佩蓮 112065538

一、 Algorithm Flow Chart

1. Basic Part



2. Advanced Part



二、 Data structure

1. 通用結構

用於儲存結果的資料結構。

成員名稱	類型	功能描述
reference_list	vector<string>	存儲每一筆記憶體參考的位元字串（例如：地址字串）。
hit_or_miss	vector<string>	每一筆記憶體參考對應的結果（hit 或 miss）。
miss_count	int	紀錄模擬結束後的總快取未命中次數。
indexing_bits	vector<int>	儲存最終輸出的索引位元（從大到小排序）。

2. 模擬方法一: Basic

成員名稱	類型	功能描述
offset_bit_count	int	計算偏移位元的長度，公式：log2(block_size)。
index_bit_count	int	計算索引位元的長度，公式：log2(cache_sets)。
index_bits_basic	vector<int>	儲存索引位元的編號（由低到高排序），在結果輸出階段排序為從大到小。
cache	vector<vector<string>>	快取結構，每個集合包含多個關聯表目錄（根據關聯度設定）。
reference_bits	vector<vector<bool>>	每個關聯項的引用位，用於 1-bit Clock Replacement。
clock_pointer	vector<int>	每個集合的時鐘指針，用於追蹤下一個可替換的快取位置。

3. 模擬方法二: Advanced

成員名稱	類型	功能描述
offset_bit_count	int	計算偏移位元的長度，公式：log2(block_size)。
index_bit_count	int	計算索引位元的長度，公式：log2(cache_sets)。
Z, O	vector<int>	紀錄每個位元出現 0 或 1 的次數。
E_count	vector<vector<int>>	統計任意兩個位元相等的次數，用於計算相關性（C 矩陣）。
D_count	vector<vector<int>>	統計任意兩個位元不相等的次數，用於計算相關性。
Q	vector<double>	每個位元的資訊增益比，用於貪婪選擇最佳位元作為索引位元。
C_val	vector<vector<double>>	兩位元之間的相關性值矩陣，公式：C(i, j) = min(e, d) / max(e, d)。
selected_order	vector<int>	儲存動態選出的索引位元的順序。
chosen_index_bits	vector<int>	最終選出的索引位元（大小排序）。
cache	vector<vector<string>>	快取結構，每個集合包含多個關聯表目錄。
reference_bit	vector<vector<bool>>	每個關聯項的引用位，用於 1-bit Clock Replacement。
clock_pointer	vector<int>	每個集合的時鐘指針，用於追蹤下一個可替換的快取位置。

4. 比較

項目	模擬方法一: Basic	模擬方法二: Advanced
索引位元選擇	直接選擇連續位元，基於地址中的位元區段。	動態選擇，基於 Q 矩陣和 C 矩陣計算出資訊增益最大化的位元。
結構複雜度	簡單，固定索引位元。	複雜，需統計每位元的 0/1 出現次數及相關性矩陣。
實現靈活性	靜態設計，適用於簡單快取模擬。	動態調整，適用於快取命中率優化的情境。
1-bit Clock Replacement	同樣採用，處理引用位更新與替換策略一致。	同樣採用，處理引用位更新與替換策略一致。