

tags: 計算機結構

## HW3

### 3.1. 計算 Figure 3.47 的 clock cycle

```
fld f2, 0(Rx) +3  
execution +1  
l0: fmul.d f2, f0, f2 +4  
execution +1  
l1: fdiv.d f8, f2, f0 +10  
execution +1  
l2: fld f4, 0(Ry) +3  
execution +1  
l3: fadd.d f4, f0, f4 +2  
execution +1  
l4: fadd.d f10, f8, f2 +2  
execution +1  
l5: fsd f4, 0(Ry) +1  
execution +1  
l6: addi Rx, Rx, 8 +0  
execution +1  
l7: addi Ry, Ry, 8 +0  
execution +1  
l8: sub x20, x4, Rx +0  
execution +1  
l9: bnz x20, Loop +1  
execution +1
```

現在我們將每個指令的延遲加總起來：

$$4 + 5 + 11 + 4 + 3 + 3 + 2 + 1 + 1 + 1 + 2 = 37$$

但是，依照題目還要考慮分支被接受後的 delay slot，題目假設 delay slot 是一個週期，所以再添加一個週期。

$$37 + 1 = 38$$

因此，答案為 38 個 clock cycle

### 3.11

A. HOW MANY CLOCK CYCLES PER LOOP ITERATION ARE LOST TO BRANCH OVERHEAD?

4 個 clock cycles

- 在沒有 bypassing 的情況下，sub指令的結果要等到 Write-back 階段才可用
- 這在迴圈結束時增加了額外的 4 個 clock cycles，因為下一個迴圈的指令無法在分支完成之前開始執行。

**B. ASSUME A STATIC BRANCH PREDICTOR, CAPABLE OF RECOGNIZING A BACK- WARD BRANCH IN THE DECODE STAGE. NOW HOW MANY CLOCK CYCLES ARE WASTED ON BRANCH OVERHEAD?**

**2 個 clock cycles**

- Static branch predictor 通常使用一些簡單的啟發式方法來預測分支的執行結果
- 像是假設迴圈的分支通常是被採取的，因此對於具有負偏移量的分支，可以預測它們是被採取的
- 在上述例子中，仍然需要 fetch 和 decode 來檢查這一點，因此會損失 2 個 clock cycles。

**C. ASSUME A DYNAMIC BRANCH PREDICTOR. HOW MANY CYCLES ARE LOST ON A CORRECT PREDICTION?**

**0 個 clock cycle**

- Dynamic branch predictor 利用運行時的資訊和過去的分支行為來進行預測，使處理器能夠在不停滯的情況下取指和執行正確的指令
- 當遇到分支指令時，Dynamic branch predictor 會根據過去的分支行為和運行時資訊來預測分支的結果（是否採取）。如果預測正確，pipeline 可以繼續取指和執行指令，而不會有任何額外的時鐘週期損失。