Лаб. Упражнение No: 5

Дисциплина: Компютърни Архитектури

Асистент: Иван Янчев

Тема: Branch Predictors

========================================================================

**Теория:   
 Branch Predictor** - Това е цифрова електроника целяща да предвиди дали един условен преход (скок) ще бъде взет или не. Съществуват различни видове Branch predictor-и, като едни от най-простите предвиждат че скок винаги ще бъде взет (**Always taken**), или обратното, никога няма да бъде взет (**Always not taken**). Друг метод е да се пази история на предишни скокове. Примерно в 1 бит да се пази информация предишния преход дали е бил взет (1) или не (0). Съществуват и много по-сложни и изтънчени решения.

В на настоящото упражнение ще използваме т.н 2-bit Predictor, който може да се

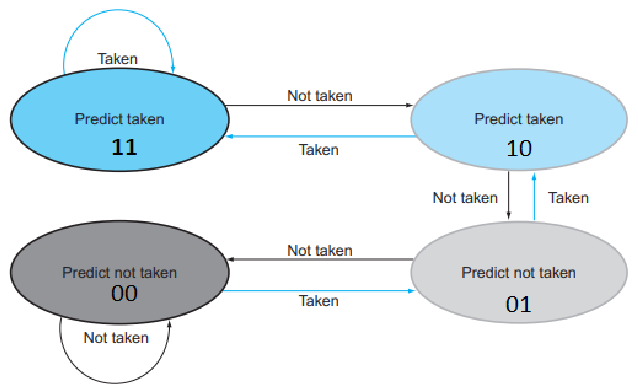
представи като finite-state машина. Като в зависимост от състоянието в която тя се

намира и от новопостъпилият сигнал (**brach taken** или **not taken**), се преминава в

последващо състояние. Като Predict **strongly taken** (11) & Predict **weakly taken** (10)

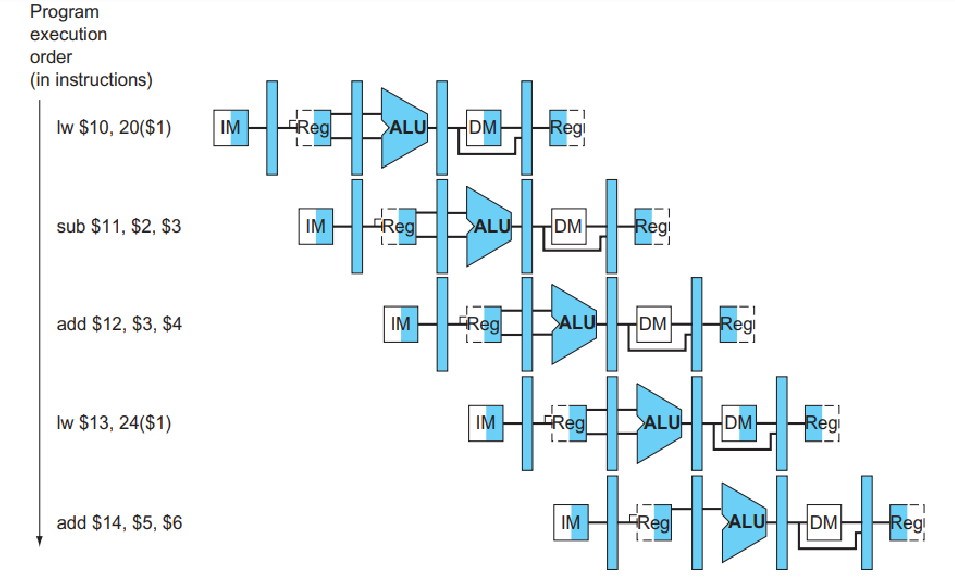
Предвиждат че преходът е взет, а s**trongly not taken** (00) & **weakly not taken** (01)

предвиждат че преходът не е взет. (фиг 1)



Фиг 1.

**Branch predictor**-а е важен защото вместо да чакаме условието да бъде изчислено за да разберем от кой адрес ще извличаме последващи инструкции (което отнема множество цикли при **pipeline**), можем веднага да започнем да изпълняваме инструкции от адрес от който сме предвидили че ще е правилен. Ако този адрес се окаже неправилен. Трябва да изхвърлим лошите инструкции от нашия pipeline и да продължим изпълнението от правилния адрес.



**Бонус:** Exception handling стр. 333

1. **За протокола**

Учебник стр.374 - 4.28, 4.29

Бонус: 4.30

1. **Линкове**

**Упражнения**: [https://github.com/tu-iyan/Computer-Architecture-2024](./упражнения)

**Учебник:** [http://library.lol/main/373C67B0C5E22C9B92B1D8FACDC47E68](./учебник)