

# hw7

李晨昊 2017011466

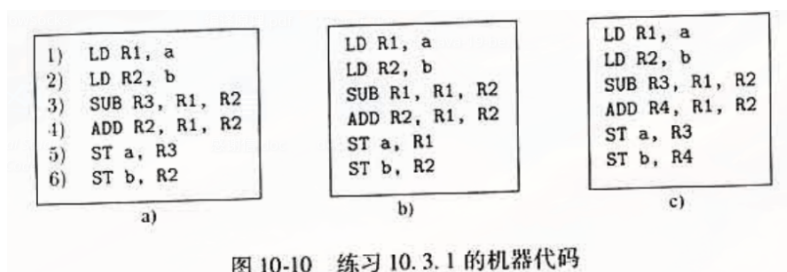
2019-11-6

## 目录

<b>1</b>	<b>Exercise 10.3.2</b>	<b>1</b>
1.1	.....	1
1.2	.....	2
1.3	.....	3

## 1 Exercise 10.3.2

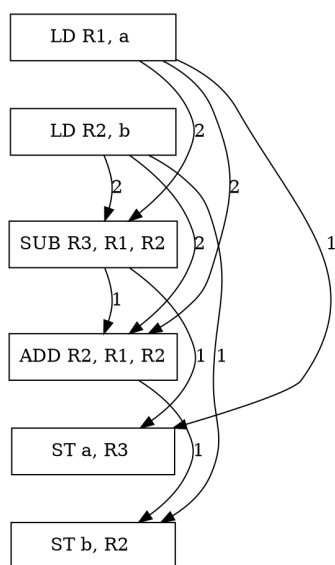
假设一个机器具有一个 ALU 资源 (用于 ADD 和 SUB 运算) 和一个 MEM 资源 (用于 LD 和 ST 运算)。假设除了 LD 运算需要两个时钟周期之外, 其余所有运算都只需要一个时钟周期。在对一个内存位置的 LD 运算执行一个时钟周期之后就可以执行对同一个位置的 ST 运算。为图 10-10 中的每个代码片段寻找一个最短调度方案。



以下假设可以确定 a 与 b 不指向同一个内存地址。

### 1.1

数据依赖图如下:



一个带优先级的拓扑排序为：{1, 2, 3, 4, 5, 6}。

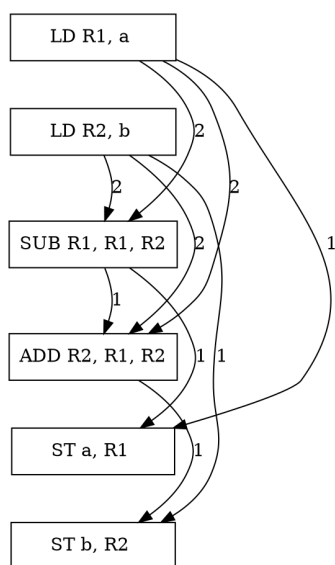
调度方案为：

```

t = 0: LD R1, a
t = 1: LD R2, b
t = 2: nop
t = 3: SUB R3, R1, R2
t = 4: ADD R2, R1, R2; ST a, R3
t = 5: ST b, R2
  
```

## 1.2

数据依赖图如下：



一个带优先级的拓扑排序为：{1, 2, 3, 4, 5, 6}。

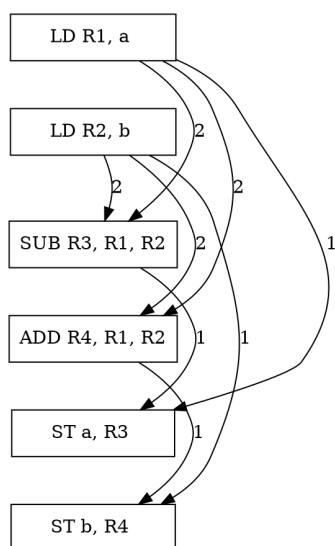
调度方案为：

```

t = 0: LD R1, a
t = 1: LD R2, b
t = 2: nop
t = 3: SUB R1, R1, R2
t = 4: ADD R2, R1, R2; ST a, R1
t = 5: ST b, R2
  
```

### 1.3

数据依赖图如下：



一个带优先级的拓扑排序为：{1, 2, 3, 4, 5, 6}。

调度方案为：

```

t = 0: LD R1, a
t = 1: LD R2, b
t = 2: nop
t = 3: SUB R3, R1, R2
t = 4: ADD R4, R1, R2; ST a, R3
t = 5: ST b, R4
  
```