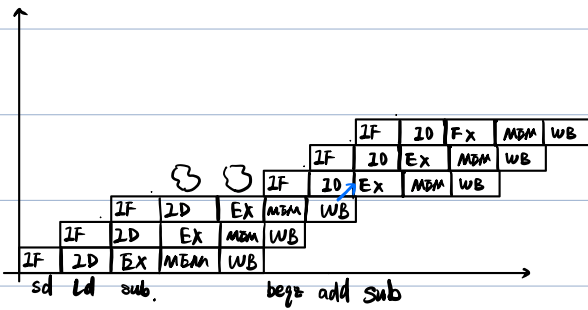


Hw 5. 4.16.4 由于只有 Load 与 Store 会使用数据寄存器, 所以利用率仅为 33%

4.16.5. 由于 Jump 都是类似 Branch 的指令, 所以, Jump 与 Store 用不上写回寄存器, 寄存器堆写回利用率 65%.

4.22.1



哈佛结构中:

假设 sub 与 beqz 没有段间寄存器 forwarding, beqz 需要等待 sub 写回才能取出结果

则 stall 2 个周期 (因为寄存器堆同步异步读)。

4.22.2 在哈佛结构下可以, 如下

```

sub x17, x15, x14
sd ...
ld ...
beqz x17, label
add ...
sub x15, x30, x14
    
```

4.22.3, 对于冯诺依曼结构, 指令与数据存在一起, 即使是插入 NOP 指令, 本质为 addi, x0, 0,

也同时需从存储器中取指令, 无法避免结构相关

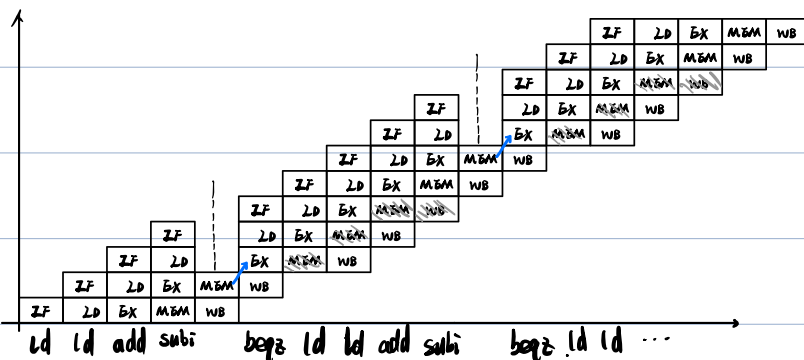
对于哈佛结构, 则可以通入插入 NOP, 解决结构相关 (如寄存器堆同步读写)

4.22.4. 对于冯诺依曼结构主要的结构相关主要是由于 ld, sd 指令导致的冲突

所以有 36% 的停顿

而对哈佛结构, 寄存器堆的结构相关与指令顺序, 种类相关, 无法具体讨论

4.25



在全负载运行中可以看出 不存在某一阶段, 五个流水级都在工作