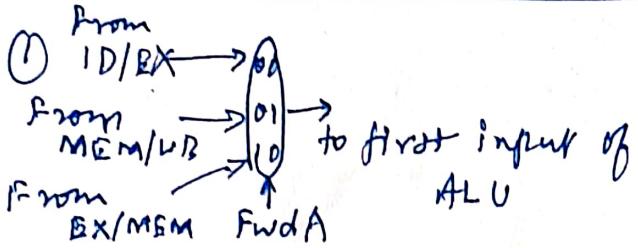
CS230 Tut 7 Solutions

(1) 

Start with same code as in slides (slide 7) - fwd from reg-reg to reg-reg

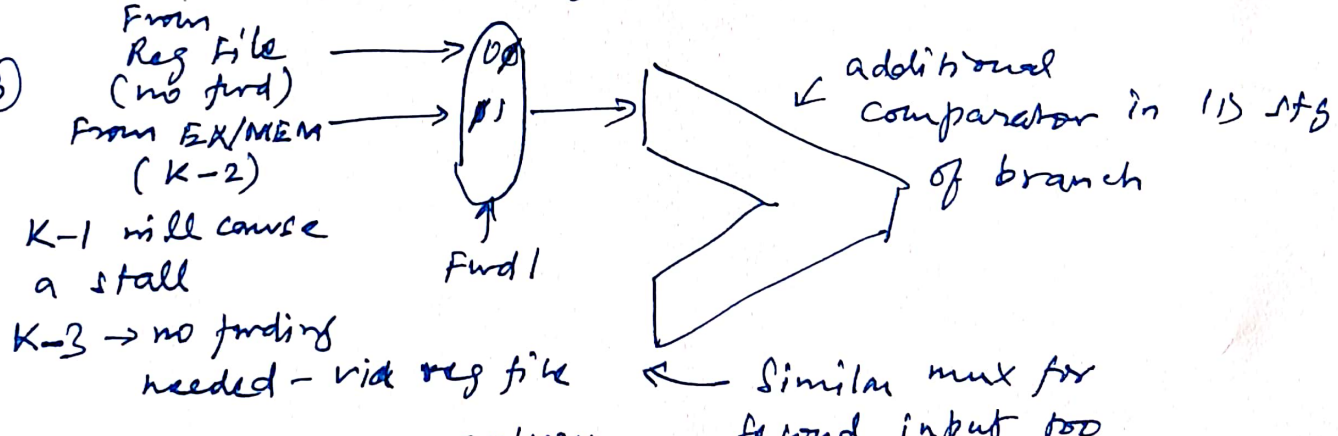
Add this condition at the end:

```
if((IF/ID.Rs == EX/MEM.Rt) &&
   (EX/MEM.MemRd == 1))
    FwdA = 01
```

Similar for FwdB - replace Rs with Rt

(2)
```
if ((IF/ID.Rs == ID/EX.Rt) &&
   (ID/EX.MemRd == 1))
    STALL
```

- can be combined with previous if conditions
- the STALL if case will be satisfied during the first time the dependent reg-reg instruction is in ID
- the forwarding if case will be satisfied during the 2nd time.

(3) 

(4)
```
if ((IF/ID.Rs == EX/MEM.Rd) &&
   (EX/MEM.RegWr == 1) &&
   (EX/MEM.MemRd == 0))
    Fwd1 = 1
else Fwd1 = 0
```

Similar for Fwd2, replace Rs with Rt

(5)
```
if ((IF/ID.Rs == ID/EX.Rd) &&
   (ID/EX.RegWr == 1) &&
   (ID/EX.MemRd == 0))
    STALL →
    IF/ID latch = 0
```

	CC1	CC2	CC3	CC4	CC5
lw	IF	ID	EX	MEM	WB
invalid opcode		IF	ID		

Invalid opcode triggers in CC3
In CC4, first instruction of invalid opcode exception handler is fetched. CC4 also detects misaligned mem exception which takes precedence