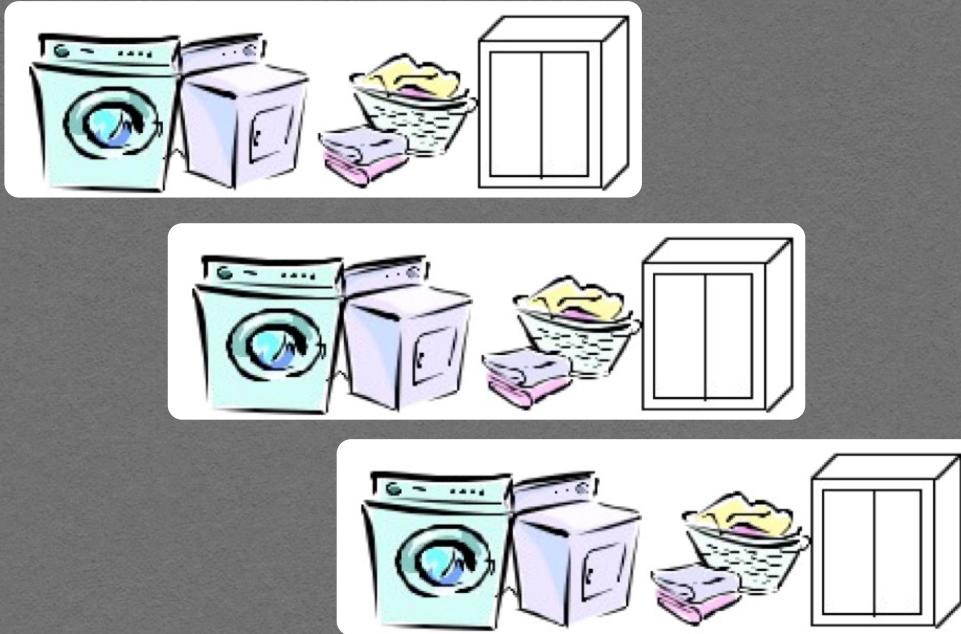
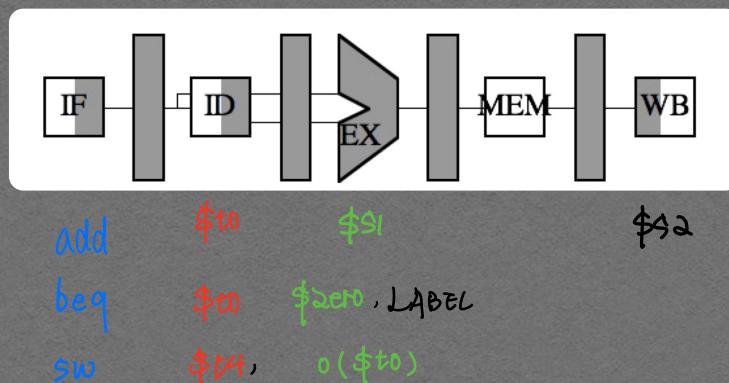
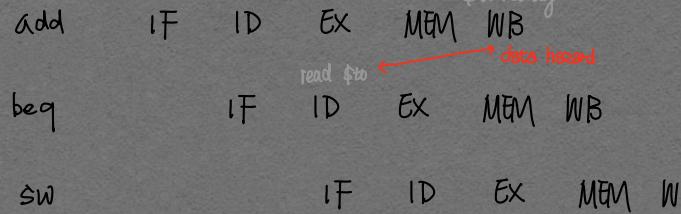


jr object.function() \$register



Instruction class	IF	ID	EX	MEM	WB
Get Inst	Get Inst	Read Regs	Do Calc	Load/ Store	Store Result
Load word (lw)	X	X	X	X	X
Store word (sw)	X	X	X	X	
R-format (add, sub, and, or, slt)	X	X	X		X
Branch (beq)	X	X	X		





software

NOP all 0s "sll \$zero, \$zero, 0"

* Stall the Pipeline

* All control bits are 0

* PC won't change

hardware

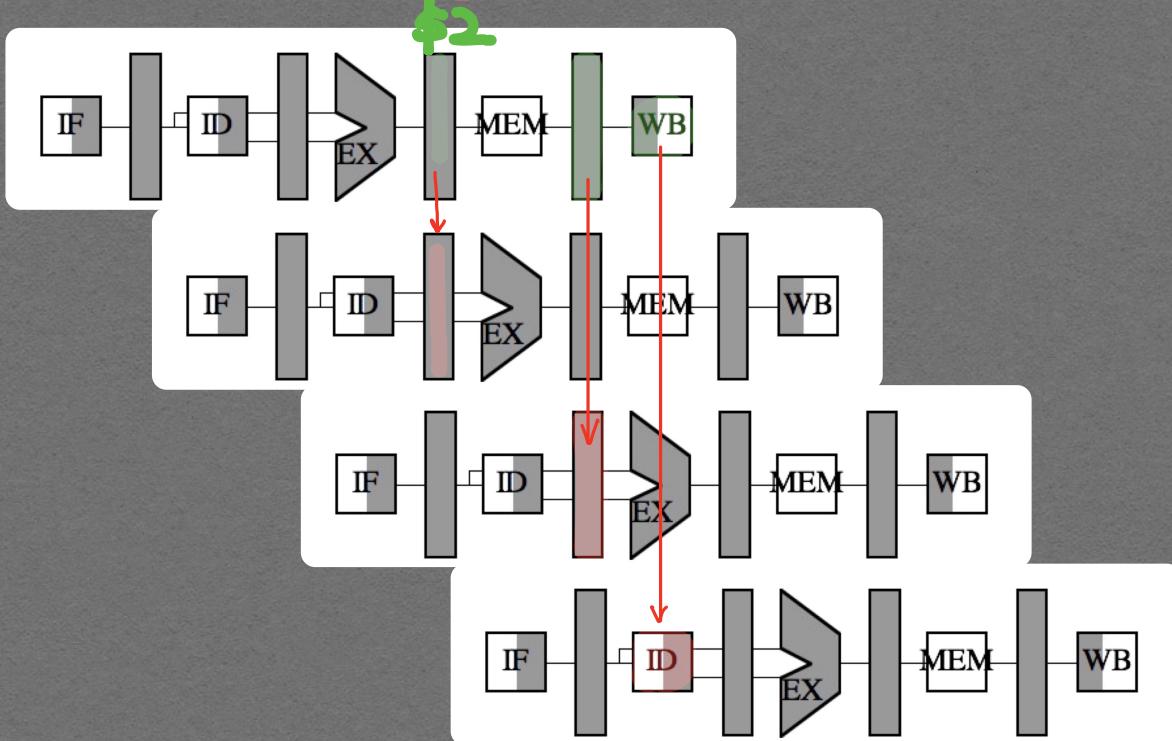
Stall

* Slow one pipe down

Data Forwarding

sub	\$2, \$1, \$3	IF	ID	EX	MEM	WB
and	\$12, \$2, \$5	IF	ID	EX	MEM	WB
add	\$13, \$2, \$2	IF	ID	EX	MEM	WB
or	\$14, \$5, \$2	IF	ID	EX	MEM	WB
sw	\$15, 100(\$2)	IF	ID	EX	MEM	WB

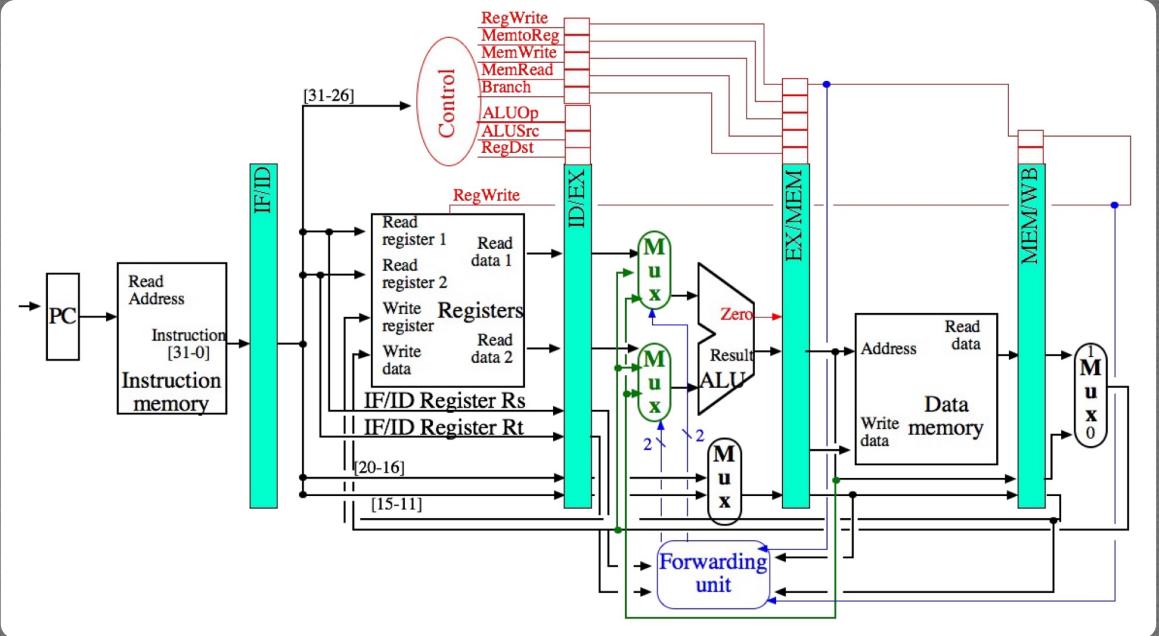
sub



and

add

sub



```

addi    $s0, $zero,10      WB
add    $s0, $s0,$t2      MEM
add    $t3, $s0,$t3      EX

if (ex_mem.regWrite == 1 && actually change reg
    em_mem.writeReg == id_ex.rs) the reg needs to be updated
    return ex_mem.aluResult;
if (inst in WB writes to rs)
    return value written by WB

```