CS274 4/27/21

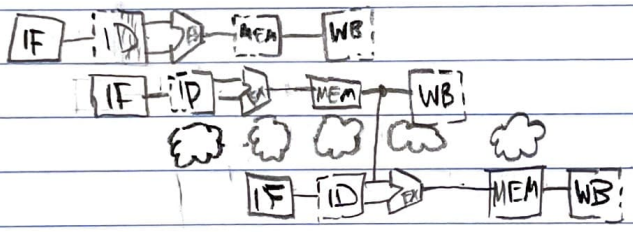
Lab 9

	1	2	3	4	5
Reg Dst	X	1	1	X	0
Reg Write	0	1	1	0	1
ALUSrc	0	0	0	1	1
PCSrc	1	0	0	0	0
MemRead	0	0	0	0	1
MemWrite	0	0	0	1	0
MemtoReg	X	0	0	X	1

1. Bne $t1, $t2, Else
2. add $a0, $t3, $t4
3. sub $s0, $t3, $t4
4. sw $ra, (0$sp)
5. lw $t0, 7($s0)

6. Well it utilizes the components at all time which makes the computer run faster, but since it needs to wait until the last instruction is done out of that section. It completes the components with using the clock.

7. lw $t1, 0($t0)
   lw $t2, 4($t0)
   add $t3, $t1, $t2



8. A data hazard can happen if your instructions are based on previous instructions that don't have enough time to get the output put on the right place. You can use "bubbles" or spaces so that your components can catch back up to continue the instructions.

9. Forwarding sends data to the next instruction without using the main registers. It just retrieves the missing data element from an internal buffer rather than waiting for it to arrive. It helps w/ data hazards since it does not have to wait for a missing element to continue.