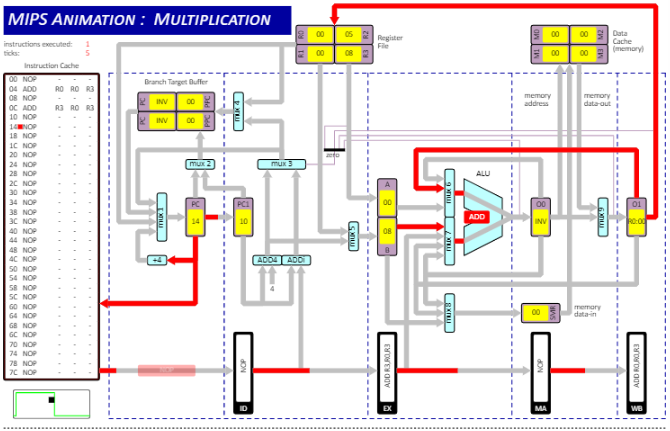
**Tutorial 4**

**Anna Maliakal -17324430**

**Q1.**

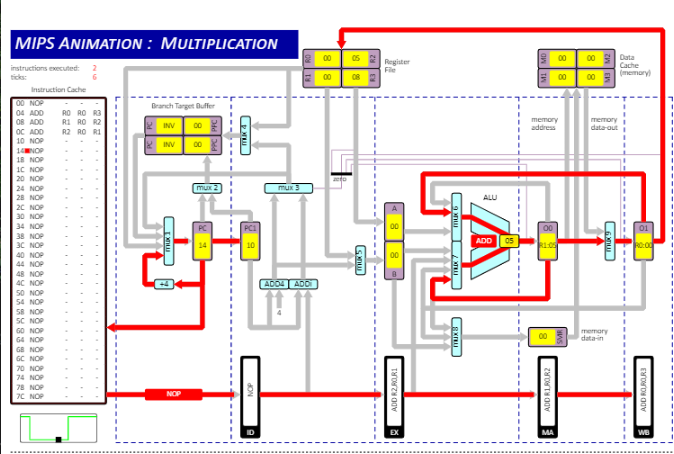
1: O1 to MUX6

* ADD R0, R0, R3
* NOP
* ADD R3, R0, R3



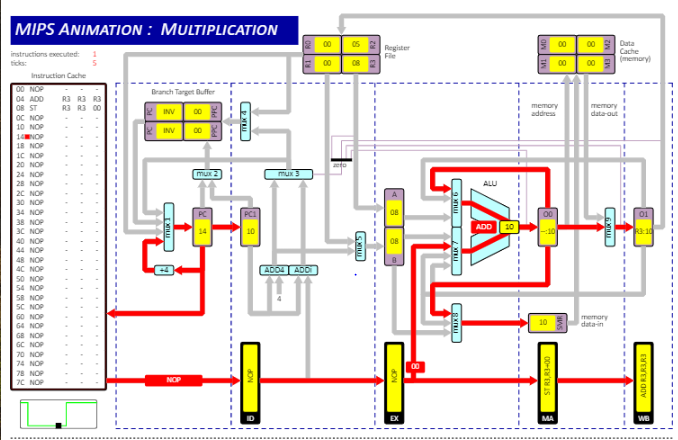
2: O0 to MUX7 and O1 to MUX6 (simultaneously)

* ADD R0, R0, R3
* ADD R1, R0, R2
* ADD R2, R0, R1



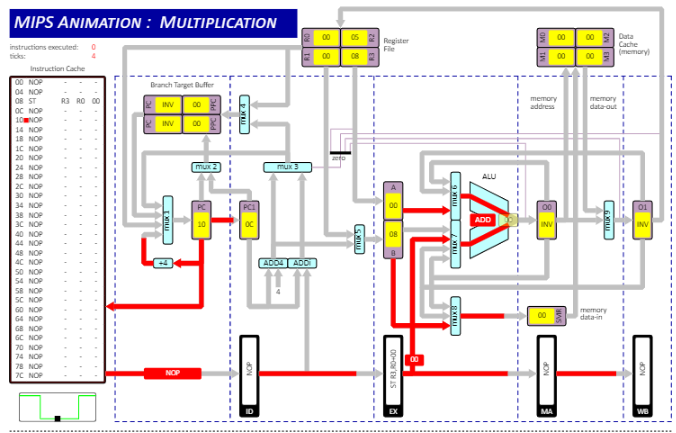
3: O0 to MUX8

* ADD R3, R3, R3
* ST R3, R3, 00



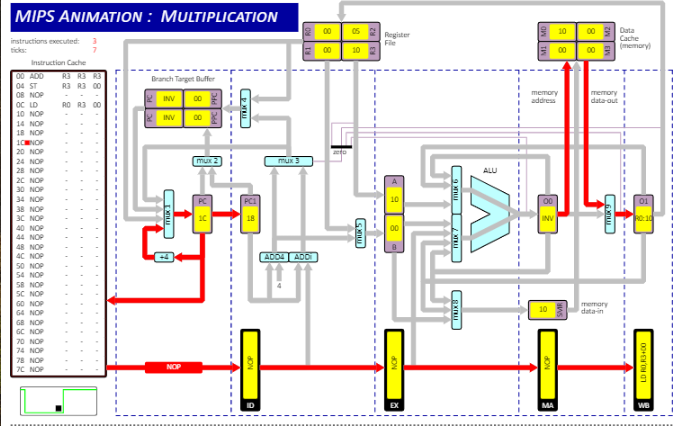
4: EX to MUX7

* ST R3, R0, 00



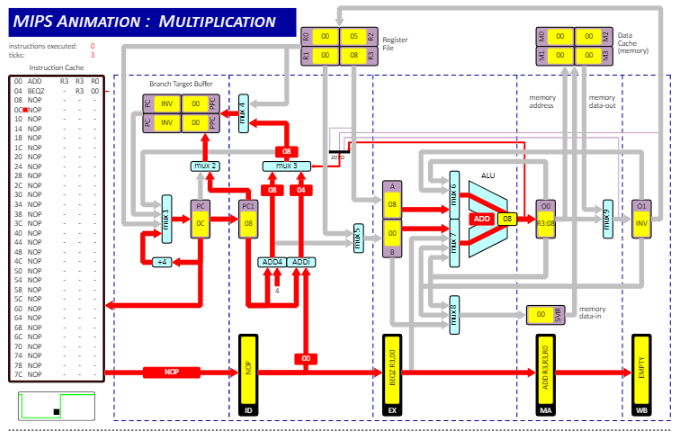
5: Data cache to MUX9 (memory data-out)

* ADD R3, R3, R3
* ST R3, R0, 00
* NOP
* LD R0, R3, 00



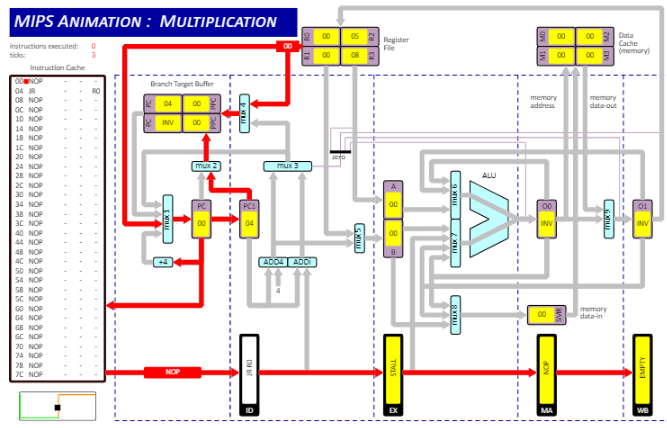
6: O0 to Zero detector

* ADD R3, R3, R0
* BEQZ R3, 00



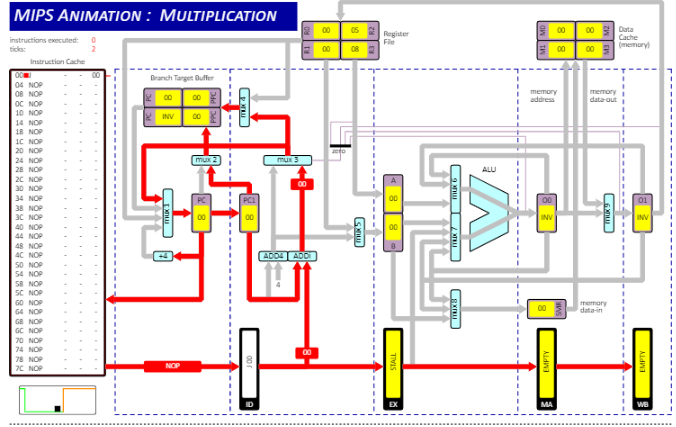
7: Register File to MUX1

* JR R0



8: Branch Target Buffer to MUX1

* J R0



**Q2.**

1. ALU Forwarding is enabled
   1. RESULTING VALUE OF R1 = 15
   2. CLOCK CYCLES = 10
   3. EXPLANATION = There are no stalls since ALU Forwarding is enabled and so there is less cycles required to reach result.
2. ALU Forwarding is disabled with CPU data dependency interlocks enabled
   1. RESULTING VALUE OF R1 = 15
   2. CLOCK CYCLES = 18
   3. EXPLANATION = Since ALU Forwarding is disabled, there are stalls because of that fact that O0 and O1 have to be stored in registers before being accessed by the ALU. This results in more cycles. The answer in R1 is correct due to CPU data dependency interlocks.
3. ALU Forwarding with CPU data dependency interlocks disabled
   1. RESULTING VALUE OF R1 = 6
   2. CLOCK CYCLES = 10
   3. EXPLANATION = As both ALU Forwarding and CPU data dependency is disabled stalls will not take place and this results in the same number of clock cycles as part (i). However the answer stored in r1 is incorrect as the program does not wait for the correct register values without the stalls.

**Q3.**

1. Default program:
2. INSTRUCTIONS EXECUTED = 38
3. CLOCK CYCLES = 50
4. EXPLANATION = The clock cycles and instructions executed do not equal because of the stalls which occur while branching initially or shifting. No instructions are executed during the stall cycles but the clock continues to cycle.
5. Branch Interlock:
   1. INSTRUCTIONS EXECUTED = 38
   2. CLOCK CYCLES = 53
   3. EXPLANATION = Branch prediction is disabled which causes the program to stall for every branch even after the initial branch for every loop.
6. Branch Interlock:
   1. INSTRUCTIONS EXECUTED = 30
   2. CLOCK CYCLES = 38
   3. EXPLANATION = The number of instructions has been decreased by 8 and the clock cycles have decreased to 38. The program no longer stalls before the first shift.