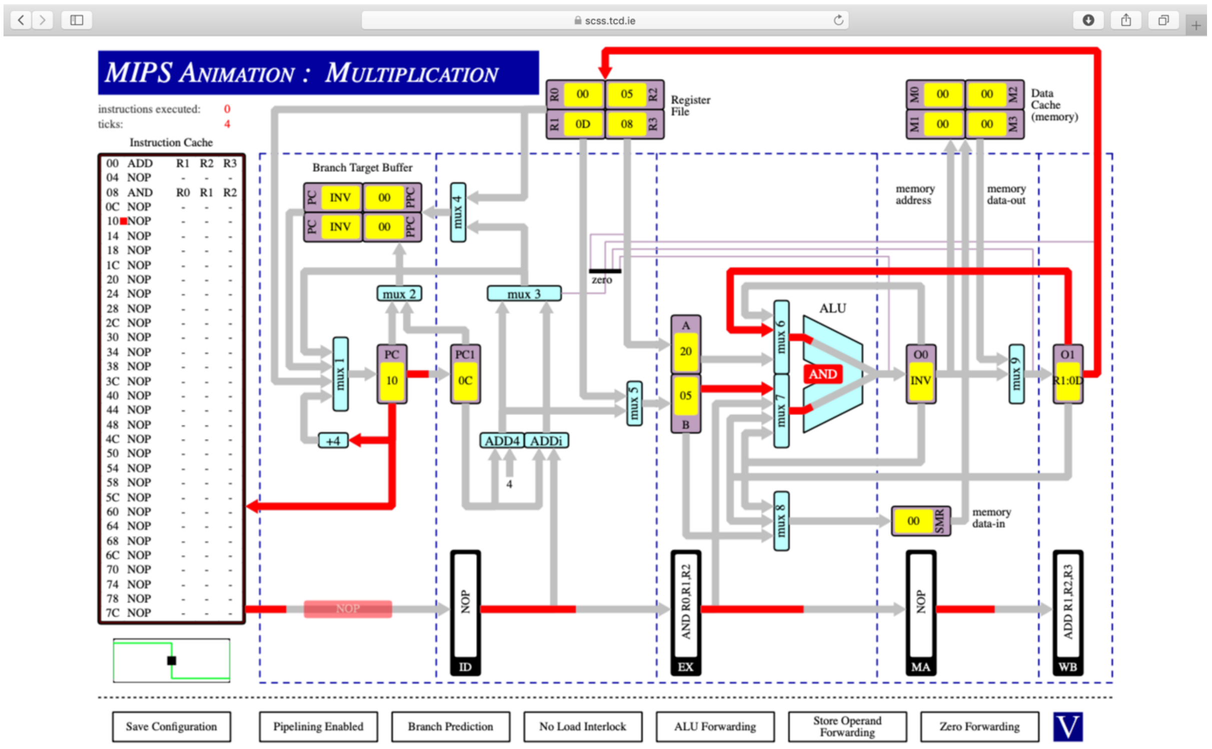
Q1

1



instructions executed: 0
ticks: 4

Instruction Cache

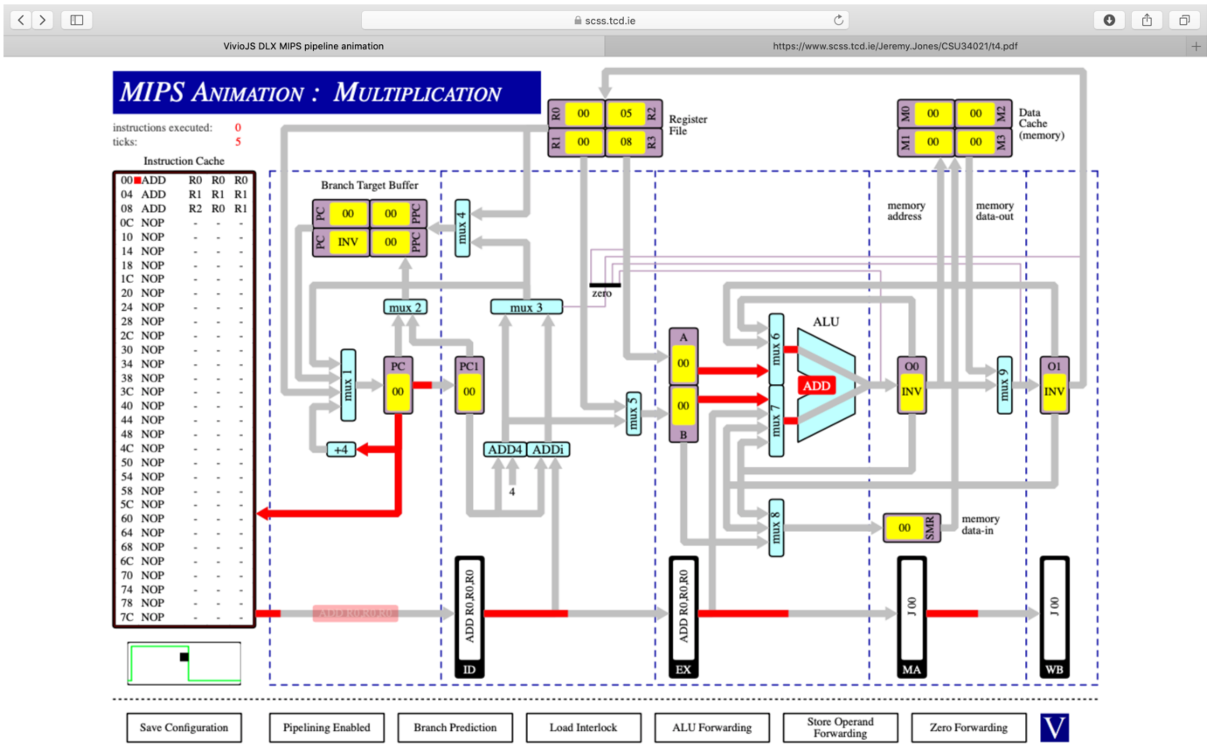
```
00  ADD   R1  R2  R3
04  NOP   -   -   -
08  AND   R0  R1  R2
0C  NOP   -   -   -
10  NOP   -   -   -
14  NOP   -   -   -
18  NOP   -   -   -
1C  NOP   -   -   -
20  NOP   -   -   -
24  NOP   -   -   -
28  NOP   -   -   -
2C  NOP   -   -   -
30  NOP   -   -   -
34  NOP   -   -   -
38  NOP   -   -   -
3C  NOP   -   -   -
40  NOP   -   -   -
44  NOP   -   -   -
48  NOP   -   -   -
4C  NOP   -   -   -
50  NOP   -   -   -
54  NOP   -   -   -
58  NOP   -   -   -
5C  NOP   -   -   -
60  NOP   -   -   -
64  NOP   -   -   -
68  NOP   -   -   -
6C  NOP   -   -   -
70  NOP   -   -   -
74  NOP   -   -   -
78  NOP   -   -   -
7C  NOP   -   -   -
```

ADD R1, R2, R3

NOP

AND R0, R1, R2

2



instructions executed: 0
ticks: 5

Instruction Cache

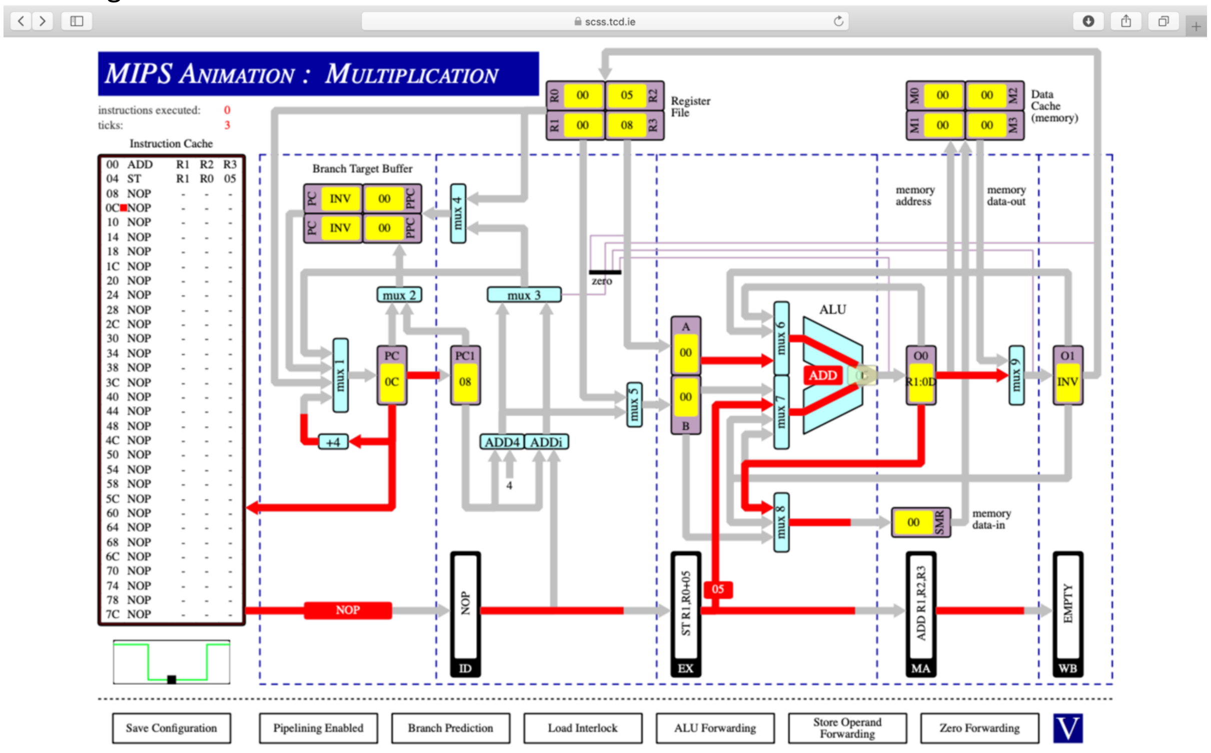
```
00  ADD   R0  R0  R0
04  ADD   R1  R1  R1
08  ADD   R2  R0  R1
0C  NOP   -   -   -
10  NOP   -   -   -
14  NOP   -   -   -
18  NOP   -   -   -
1C  NOP   -   -   -
20  NOP   -   -   -
24  NOP   -   -   -
28  NOP   -   -   -
2C  NOP   -   -   -
30  NOP   -   -   -
34  NOP   -   -   -
38  NOP   -   -   -
3C  NOP   -   -   -
40  NOP   -   -   -
44  NOP   -   -   -
48  NOP   -   -   -
4C  NOP   -   -   -
50  NOP   -   -   -
54  NOP   -   -   -
58  NOP   -   -   -
5C  NOP   -   -   -
60  NOP   -   -   -
64  NOP   -   -   -
68  NOP   -   -   -
6C  NOP   -   -   -
70  NOP   -   -   -
74  NOP   -   -   -
78  NOP   -   -   -
7C  NOP   -   -   -
```

ADD R0, R0, R0

ADD R1, R1, R1

ADD R2, R0, R1

3 & 4 together

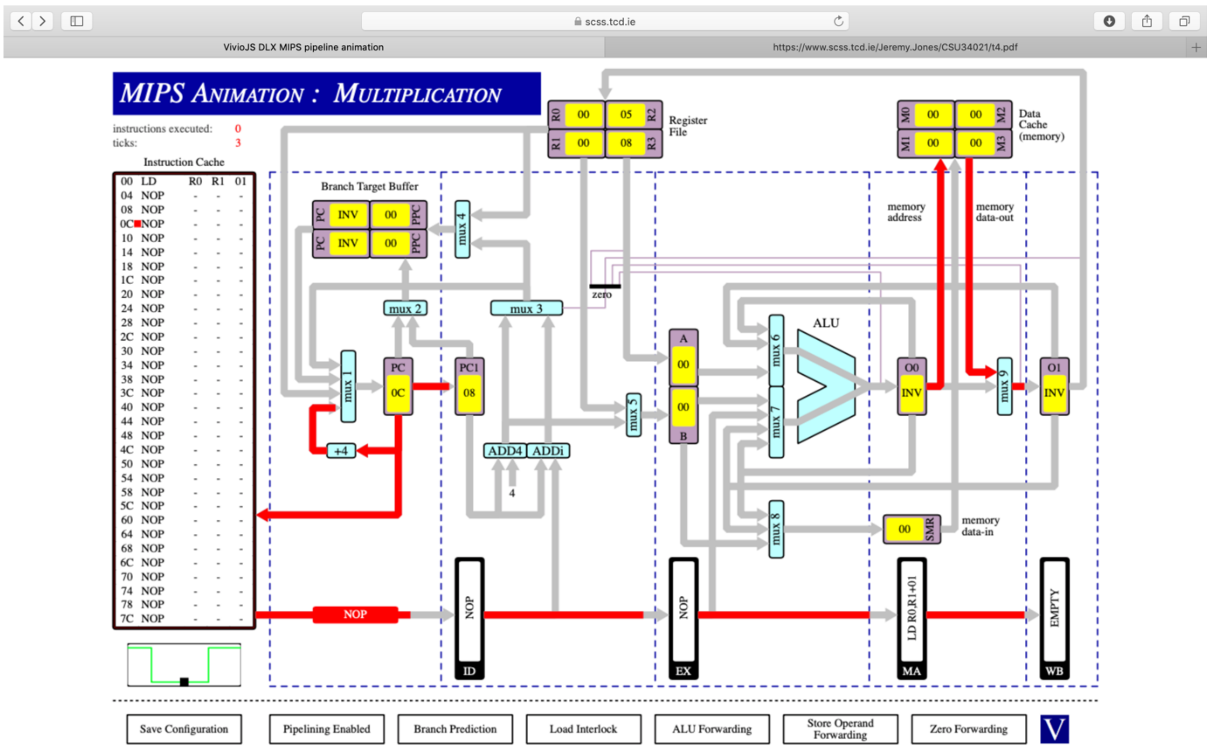


Instruction Cache

```
00  ADD   R1  R2  R3
04  ST    R1  R0  05
08  NOP   -   -   -
0C  NOP   -   -   -
10  NOP   -   -   -
14  NOP   -   -   -
18  NOP   -   -   -
1C  NOP   -   -   -
20  NOP   -   -   -
24  NOP   -   -   -
28  NOP   -   -   -
2C  NOP   -   -   -
30  NOP   -   -   -
34  NOP   -   -   -
38  NOP   -   -   -
3C  NOP   -   -   -
40  NOP   -   -   -
44  NOP   -   -   -
48  NOP   -   -   -
4C  NOP   -   -   -
50  NOP   -   -   -
54  NOP   -   -   -
58  NOP   -   -   -
5C  NOP   -   -   -
60  NOP   -   -   -
64  NOP   -   -   -
68  NOP   -   -   -
6C  NOP   -   -   -
70  NOP   -   -   -
74  NOP   -   -   -
78  NOP   -   -   -
7C  NOP   -   -   -
```

ADD R1, R2, R3
ST R1, R0, 05

5

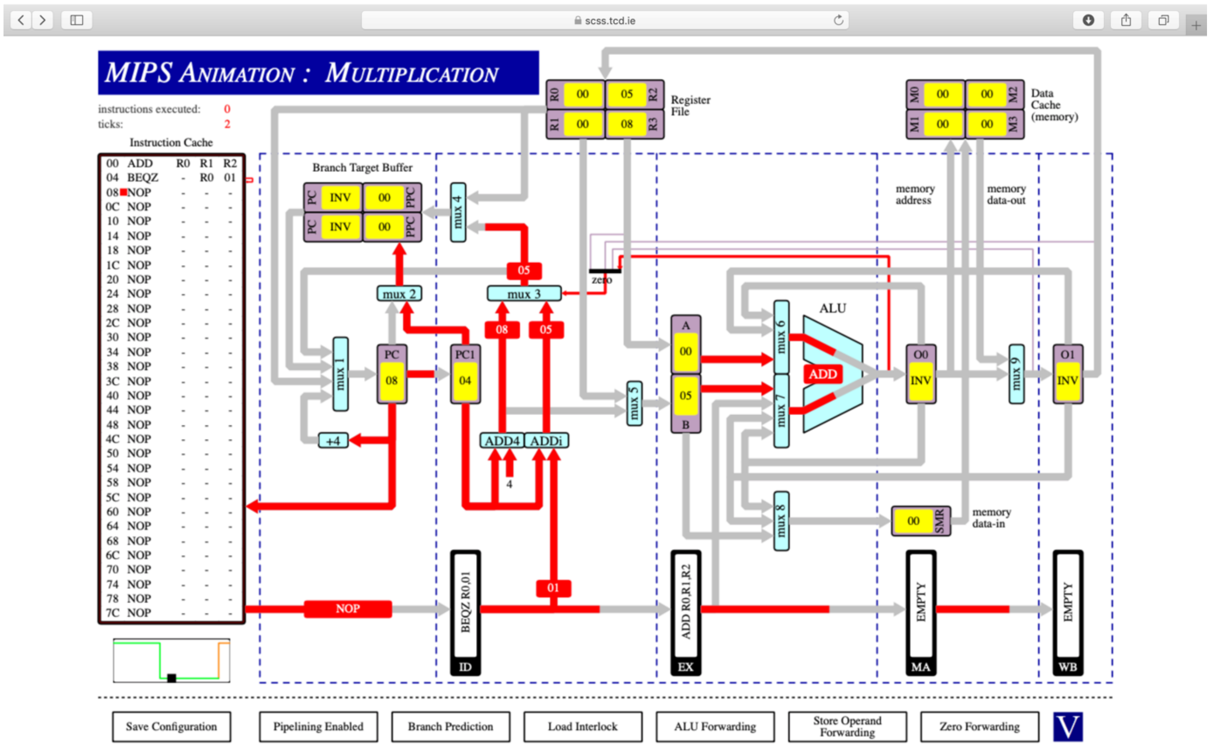


Instruction Cache

```
00  LD    R0  R1  01
04  NOP   -   -   -
08  NOP   -   -   -
0C  NOP   -   -   -
10  NOP   -   -   -
14  NOP   -   -   -
18  NOP   -   -   -
1C  NOP   -   -   -
20  NOP   -   -   -
24  NOP   -   -   -
28  NOP   -   -   -
2C  NOP   -   -   -
30  NOP   -   -   -
34  NOP   -   -   -
38  NOP   -   -   -
3C  NOP   -   -   -
40  NOP   -   -   -
44  NOP   -   -   -
48  NOP   -   -   -
4C  NOP   -   -   -
50  NOP   -   -   -
54  NOP   -   -   -
58  NOP   -   -   -
5C  NOP   -   -   -
60  NOP   -   -   -
64  NOP   -   -   -
68  NOP   -   -   -
6C  NOP   -   -   -
70  NOP   -   -   -
74  NOP   -   -   -
78  NOP   -   -   -
7C  NOP   -   -   -
```

LD R0, R1, 01

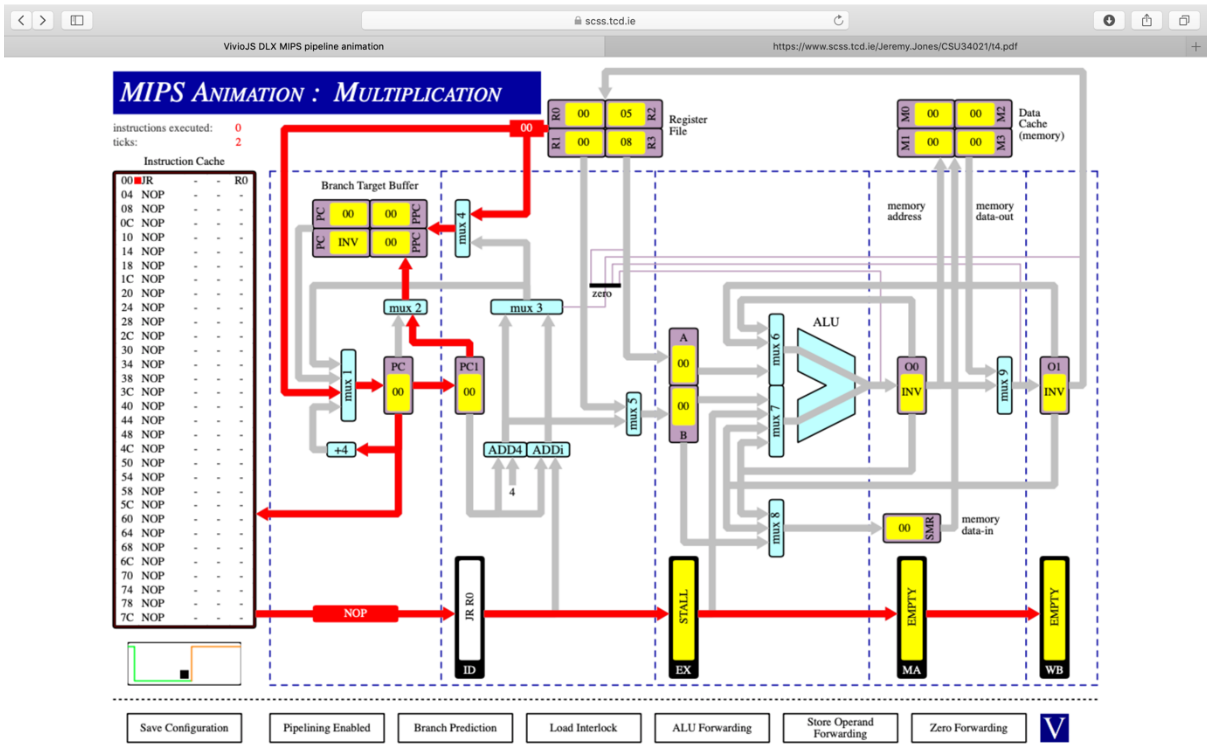
6



ADD R0, R1, R2

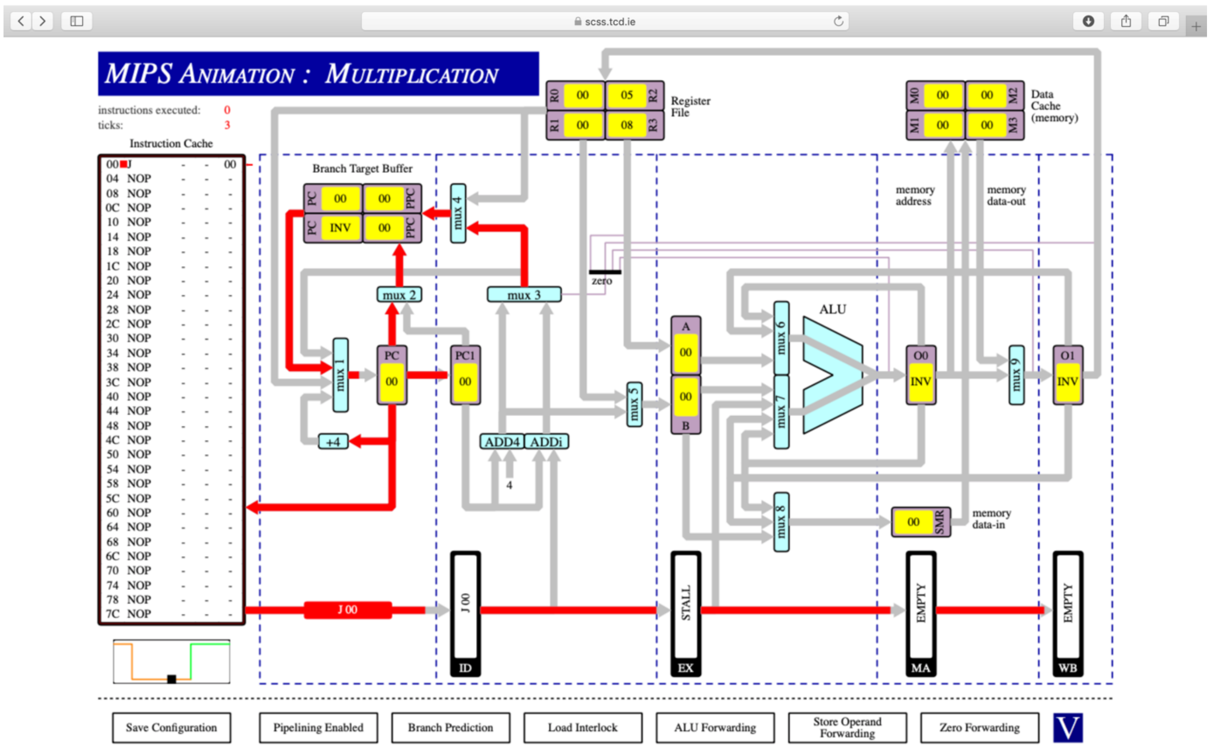
BEQZ – R0, 01

7



JR R0

8



J 00

Q2)

(i)

ticks = 10

R1 = 15

(ii)

ticks = 18

R1= 15

(iii)

ticks = 10

R1 = 6

The number of clock cycles is different with no ALU forwarding because the result from one instruction cannot be fed back into the ALU as an input. When there is no ALU forwarding or CPU data dependency stalls will not occur due to the program not waiting for O0 and O1 to be stored into registers. However without ALU forwarding, the program still needs to take values from registers and since it doesn’t wait for the correct register values the answer stored in R1 is incorrect

Q3

(i)

Ticks = 51

Instructions = 38

When branching initially or shifting, stall cycles happen. This leads to the ticks and instructions not being equal, since no instructions are executed during these cycles, the number of cycles increases independent of the number of instructions

(ii)

Ticks = 53

Instructions = 38

Once again the number of instructions and ticks is different due to the same reason as above. In this one though there is an extra two stall cycles due to branch prediction being disabled in this case causing the program to stall for every branch even after the initial branch for every loop hence leading to two extra ticks

(iii)

Ticks = 47

Instructions = 38

If the two shift instructions are swapped in the program this will lead to there being no longer a stall before the first shift. This leads to the clock cycle decreasing by four ticks