

# CS 341 Assignment 3

Devansh Jain, 190100044

September 11, 2021

## Contents

<b>1</b>	<b>5 Stage, without forwarding or hazard detection</b>	<b>1</b>
<b>2</b>	<b>5 Stage, without forwarding, with hazard detection</b>	<b>3</b>
a.	.....	3
b.	.....	4

# 1 5 Stage, without forwarding or hazard detection

## Code

```
.text
```

```
main:
```

```
    addi s0 zero 5
```

```
    add s1 s0 zero
```

## Behaviour

Expectation: At the end of the program, we expect registers **s0** and **s1** to have value 5.

Reality: At the end of the program, only register **s0** has value 5. (**s1** is default to 0)

## Explanation

The incorrect value is due to data hazard.

To be more specific, we face a **read-before-write (RAW) data hazard**.

`addi s0 zero 5` writes to register **s0** in WB stage during 4th cycle.

`add s1 s0 zero` reads from register **s0** in ID stage during 2nd cycle.

The value read here is not updated yet causing the incorrect computation.

## Screenshots

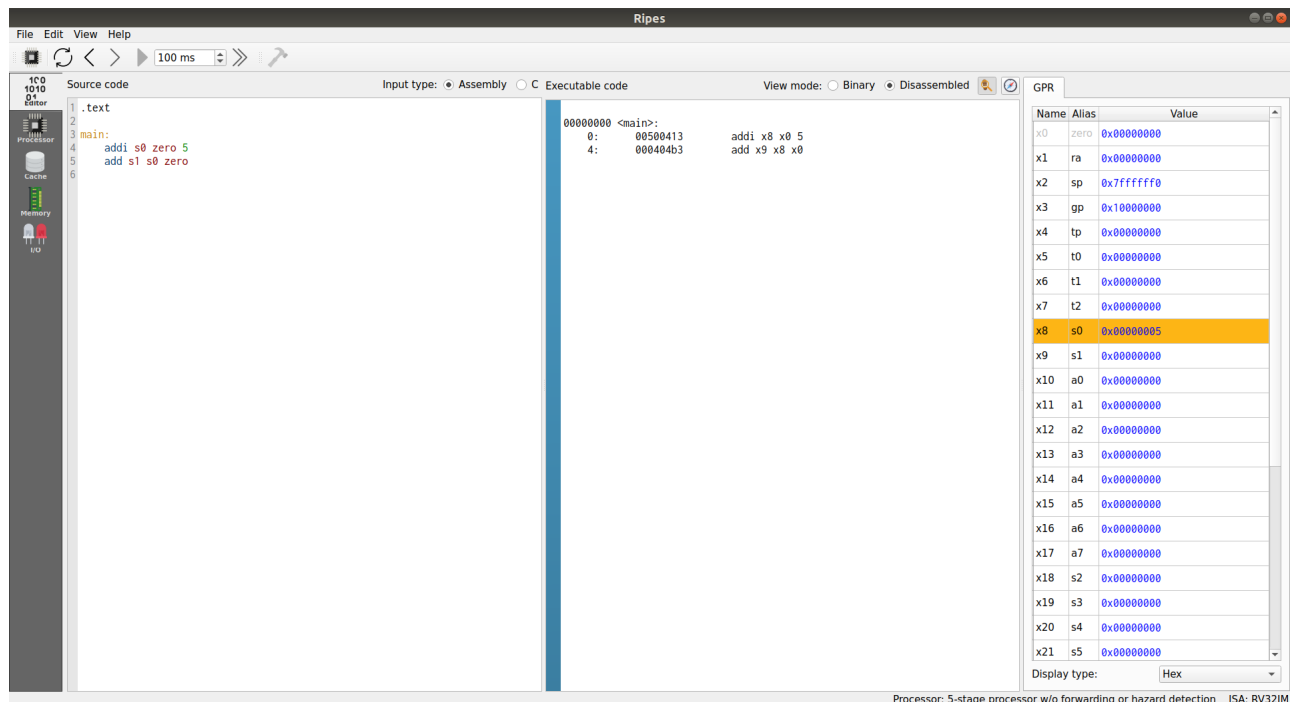


Figure 1: Without hazard detection; Without forwarding;

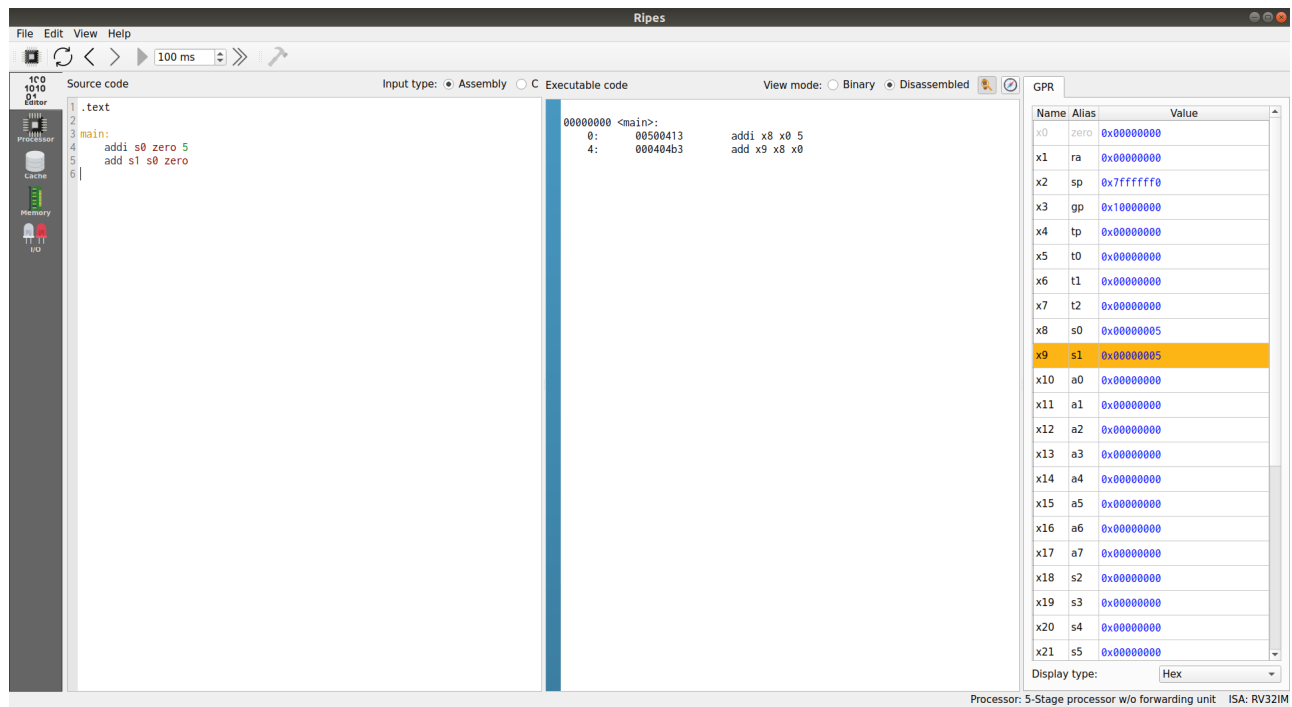


Figure 2: With hazard detection; Without forwarding;

## 2 5 Stage, without forwarding, with hazard detection

a.

Code

```
.text
```

```
main:
```

```
    addi s0 zero 5
```

```
    add s1 s0 zero
```

Behaviour

At the end of the program, we expect registers `s0` and `s1` to have value 5.

Without forwarding, we observe that ID for second instruction takes 3 cycles (2 stalls).

There are no stalls with forwarding.

Screenshots

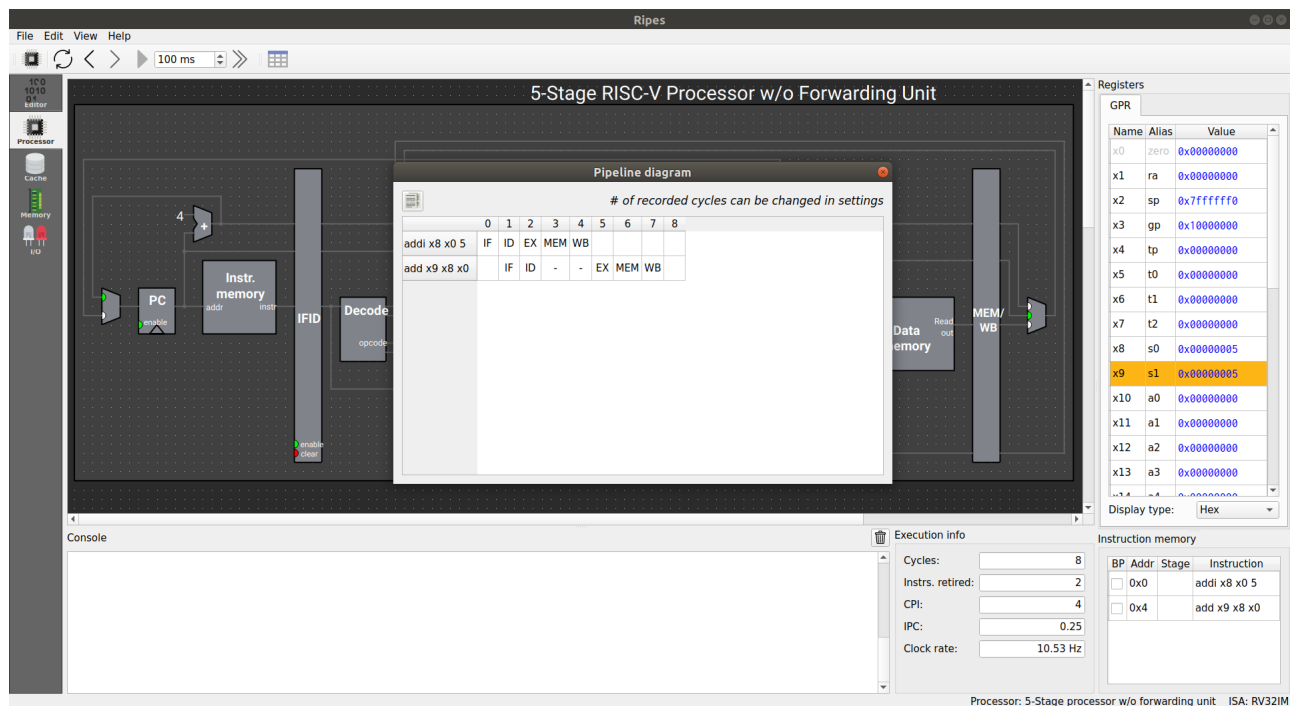


Figure 3: With hazard detection; Without forwarding;

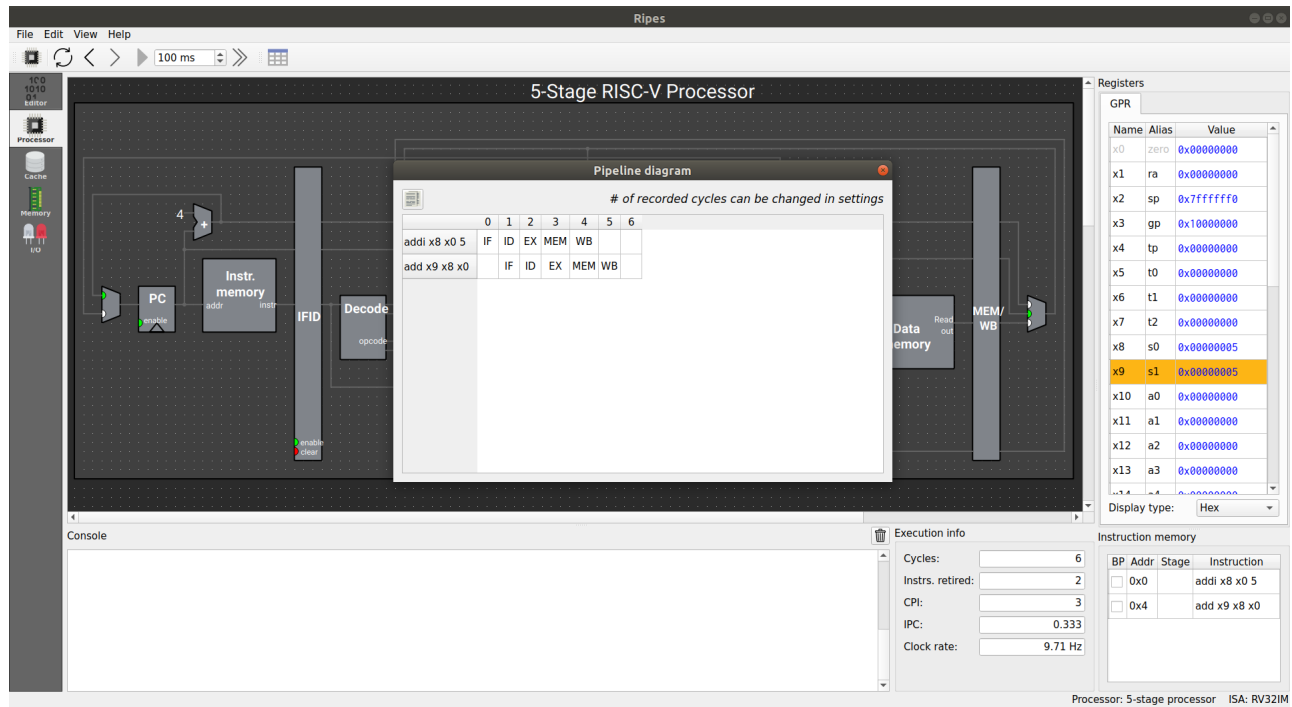


Figure 4: With hazard detection; With forwarding;

b.

**Initial Code**

```
.text

main:
    addi s0 zero 5
    add s1 s0 zero
    addi s2 zero 6
    addi s3 zero 7
```

**Optimized Code**

```
.text

main:
    addi s0 zero 5
    addi s2 zero 6
    addi s3 zero 7
    add s1 s0 zero
```

**Behaviour**

At the end of both the program, we expect registers `s0` and `s1` to have value 5, register `s2` to have value 6 and, register `s3` to have value 7.

## Screenshots

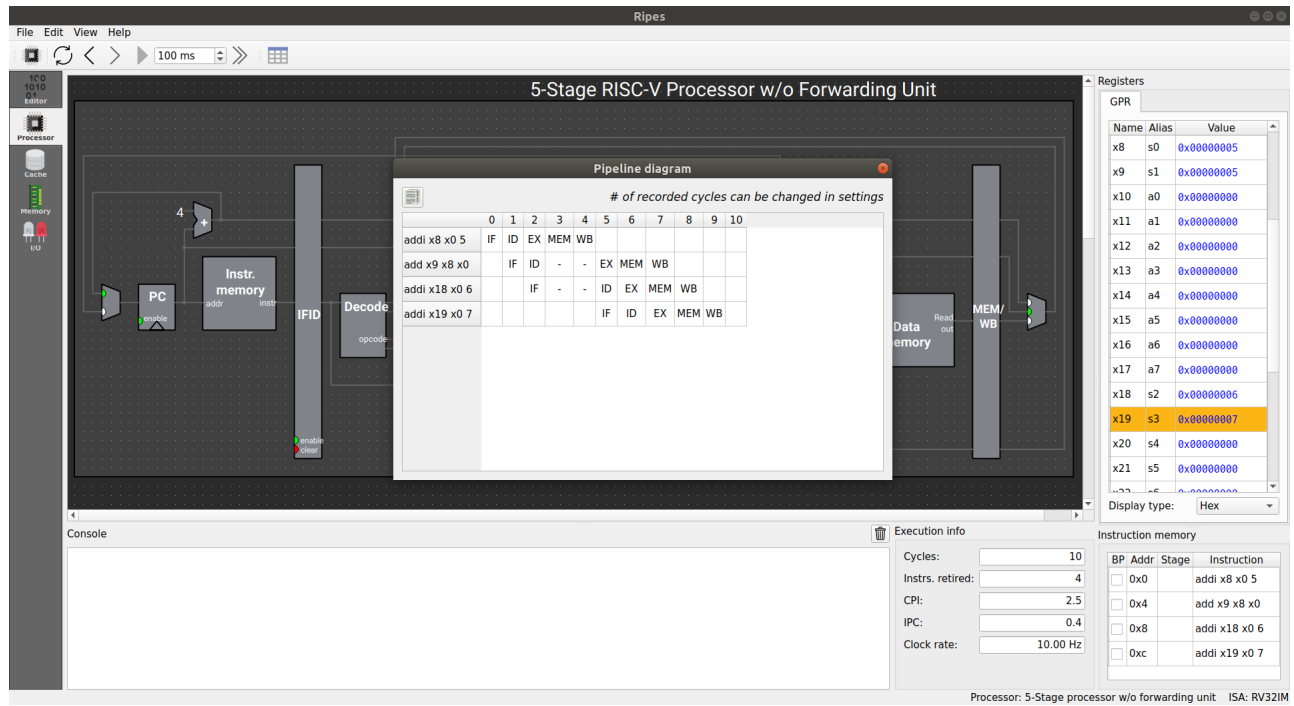


Figure 5: With hazard detection; Without forwarding; Initial code

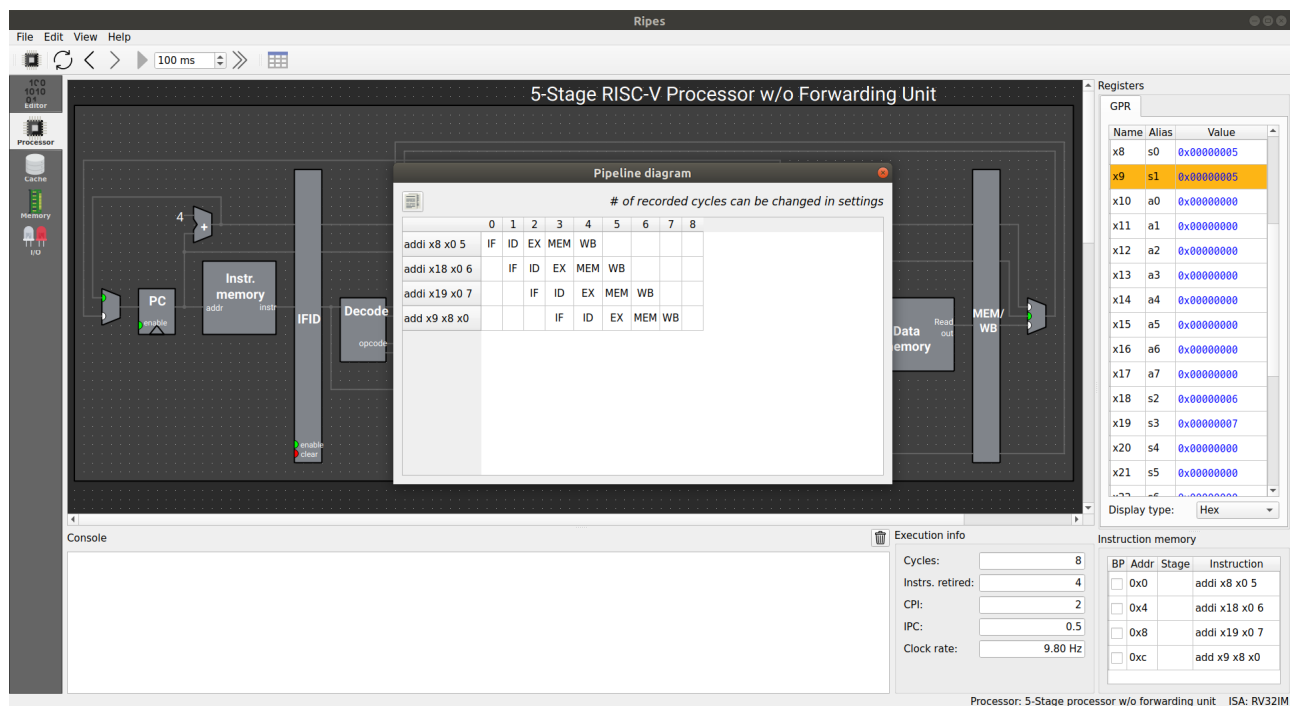


Figure 6: With hazard detection; With forwarding; Optimized code