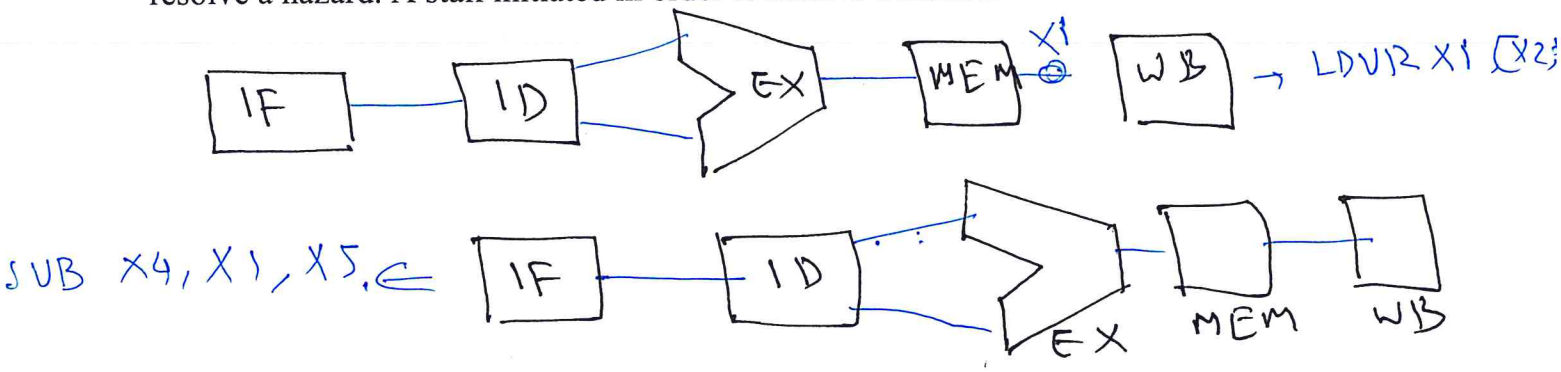
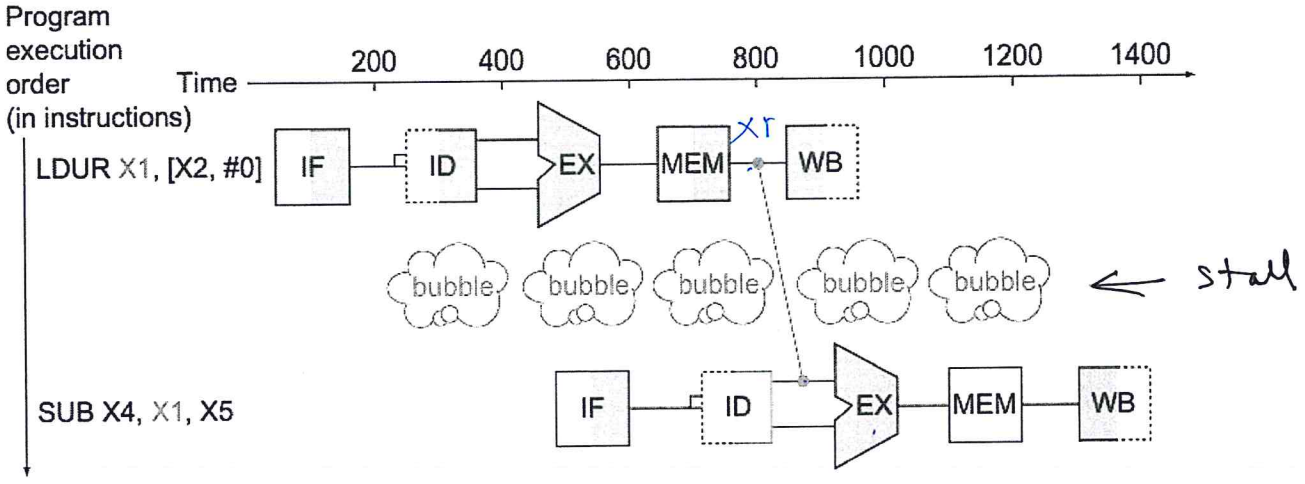
LDUR X1 [X2, #0]
SUB X4, X1, X5.

Load-Use Data Hazard

A specific form of data hazard in which the data being loaded by a load instruction have not yet become available when they are needed by another instruction.

pipeline stall / bubble: A pipeline stall is a delay in execution of an instruction in order to resolve a hazard. A stall initiated in order to resolve a hazard.





Problem:

a. **Identify Data hazards and determine total number of stalls in the assembly code. Assume there is no forwarding. Rewrite the assembly code and show stalls in the code if any. Draw the multicycle pipeline diagram.**

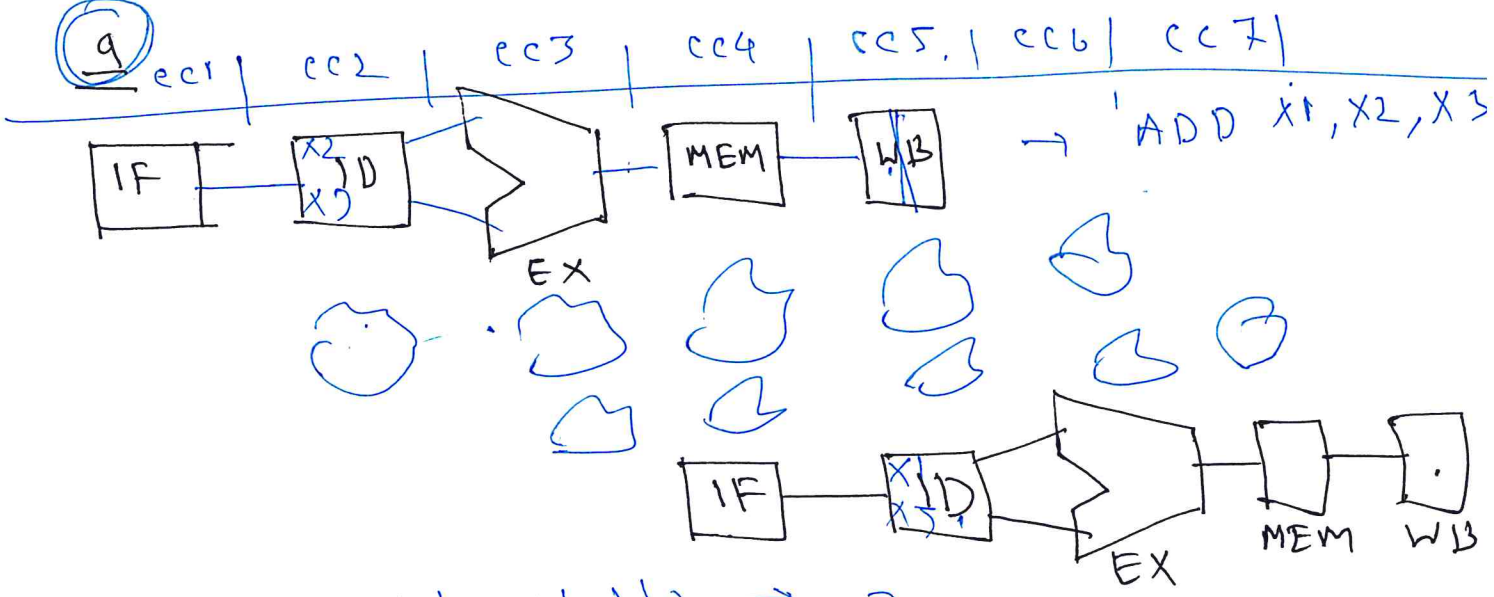
b. **Apply forwarding to reduce the stalls without changing the functionality and order. Rewrite the assembly code.**

c. **Draw the multi-cycle pipeline diagram (one loop) for the optimized assembly code from section b.**

```
ADD X1, X2, X3
ADD X4, X1, X5
```

Data Hazard.

Solution:

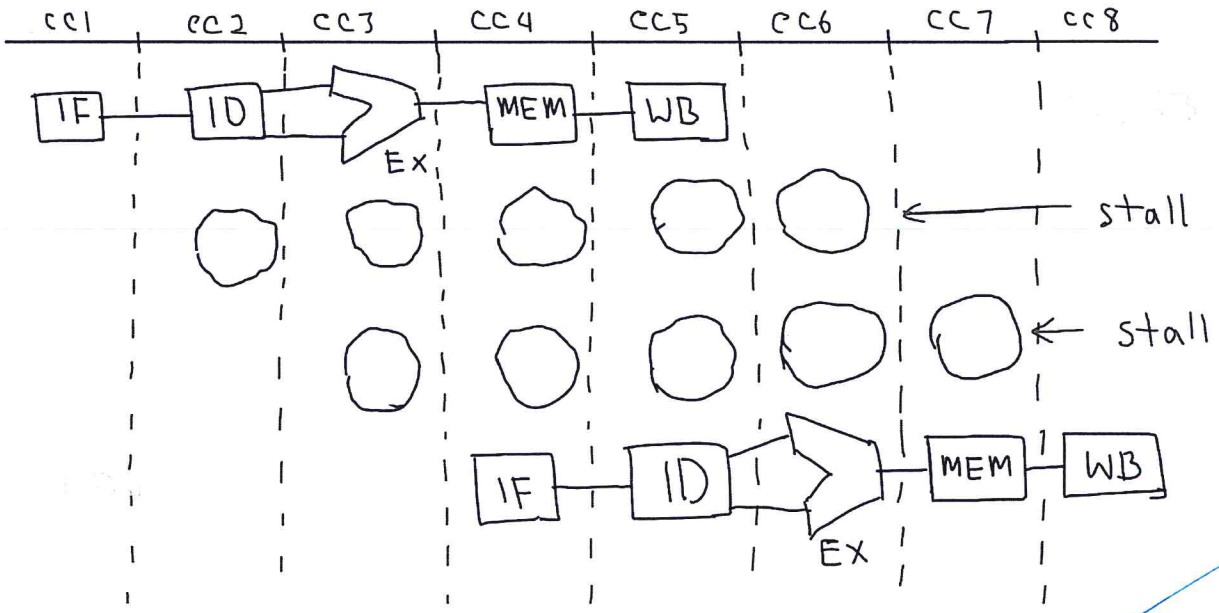


Total stalls → 2

~~ADD X1, X2, X3~~

~~ADD X4, X1, ,~~

```
ADD  X1, X2, X3
stall
stall
ADD  X4, X1, X5.
```

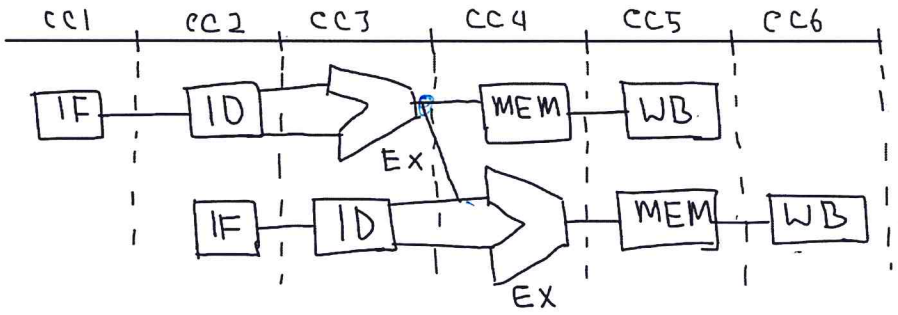


(b)

ADD X1, X2, X3

ADD X4, X1, X5.

(c) Multicycle pipeline diagram for b



Problem:

a. Identify Data hazards and determine total number of stalls in the assembly code. Assume there is no forwarding. Rewrite the assembly code and show stalls in the code if any. Draw the multicycle pipeline diagram.

b. Apply forwarding to reduce the stalls without changing the functionality and order. Rewrite the assembly code.

c. Draw the multi-cycle pipeline diagram (one loop) for the optimized assembly code from section b.

```
SUBI X2, X2, #1
LSL X4, X3, #1
SUB X5, X4, X6
LDUR X7, [X0, #0]
ADDI X7, X7, #1
```

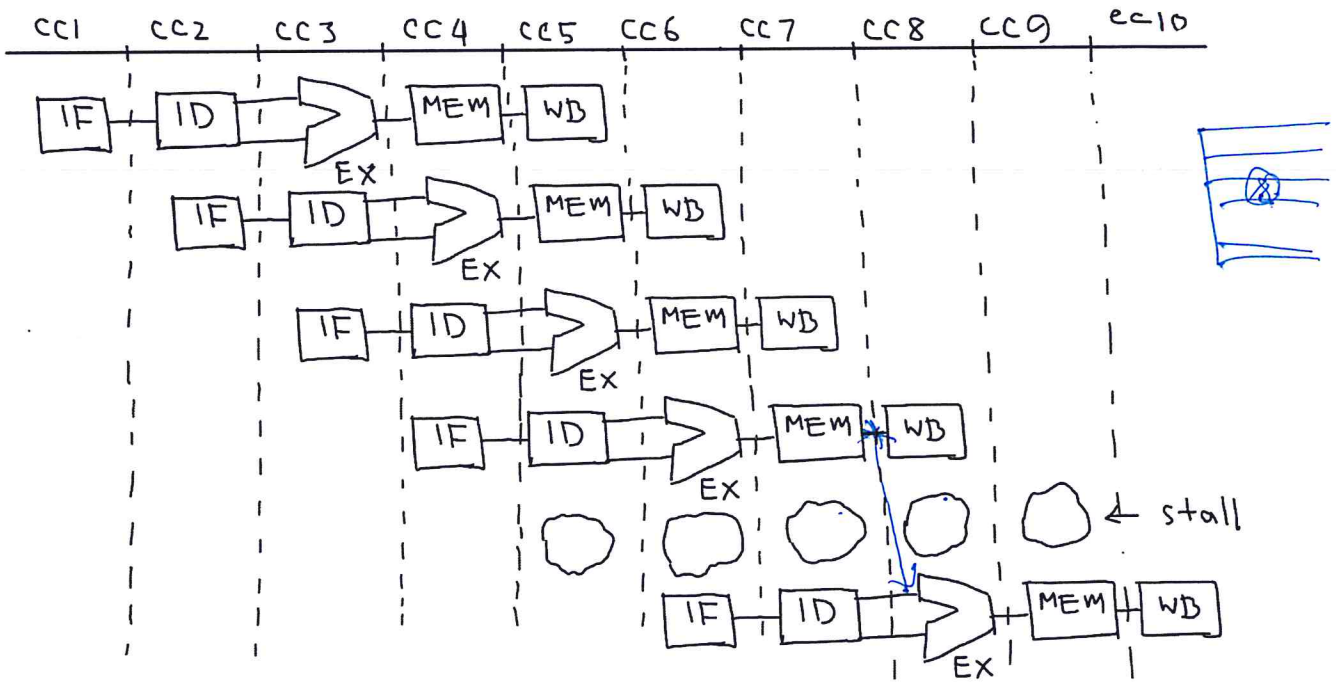
a

```
SUBI X2, X2, #1
LSL  X4, X3, #1
stall
stall
SUB  X5, X4, X6
LDUR X7, [X0, #0]
stall
stall
ADDI X7, X7, #1
```

b After forwarding,

```
SUBI  X2, X2, #1
LSL   X4, X3, #1
SUB   X5, X4, X6
LDUR  X7, [X0, #0]
stall
ADDI  X7, X7, #1
```

(c) Multicycle pipeline diagram for section b.



Data conversion

1 byte = 8 bit

1 KB = 1024 byte

~~1 KB = 1000 byte~~

1 MB = 1024 KB.

1 GB = 1024 MB

* 1048576 KB = 1 GB

∵ 1 GB = 1024 MB

= 1024 * 1024 KB.

∴ 1 KB = $\frac{1}{1024 * 1024}$ GB

1048576 KB = $\frac{1048576}{1024 * 1024}$ GB

= 1 GB

* 1 GB = [ ] bits.

1 GB = 1024 MB

= 1024 * 1024 KB

= 1024 * 1024 * 1024 Byte

= 1024 * 1024 * 1024 * 8 bits

= 8589934592 bits.