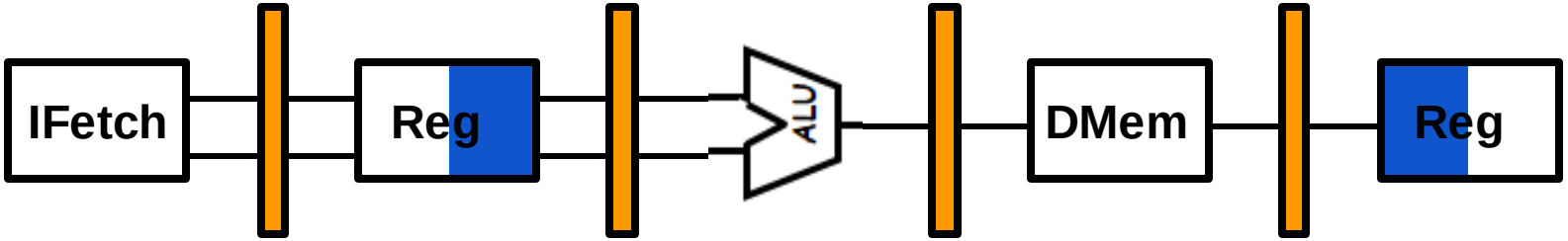
Exe 4 : Pipelining and Performance




```
lw      $1,  OFF($2)
addi    $3, $1, 4
sub     $4, $1, $3
addi    $2, $1, -8
sw      $4, OFF($2)
```

No optimization in the **MIPS** pipeline (e.g., forwarding paths) just our "optimization" (i.e., RF access R/W)
The processor has a clock cycle of 2ns

A. Draw the pipeline schema and highlight possible hazards
B. Represent the real execution (Insert the stalls)
C. Calculate IC, CPI, MIPS
D. Do the same considering the existence of path forwarding

```
lw      $1, OFF($2)
addi    $3, $1, 4
sub     $4, $1, $3
addi    $2, $1, -8
sw      $4, OFF($2)
```

RAW

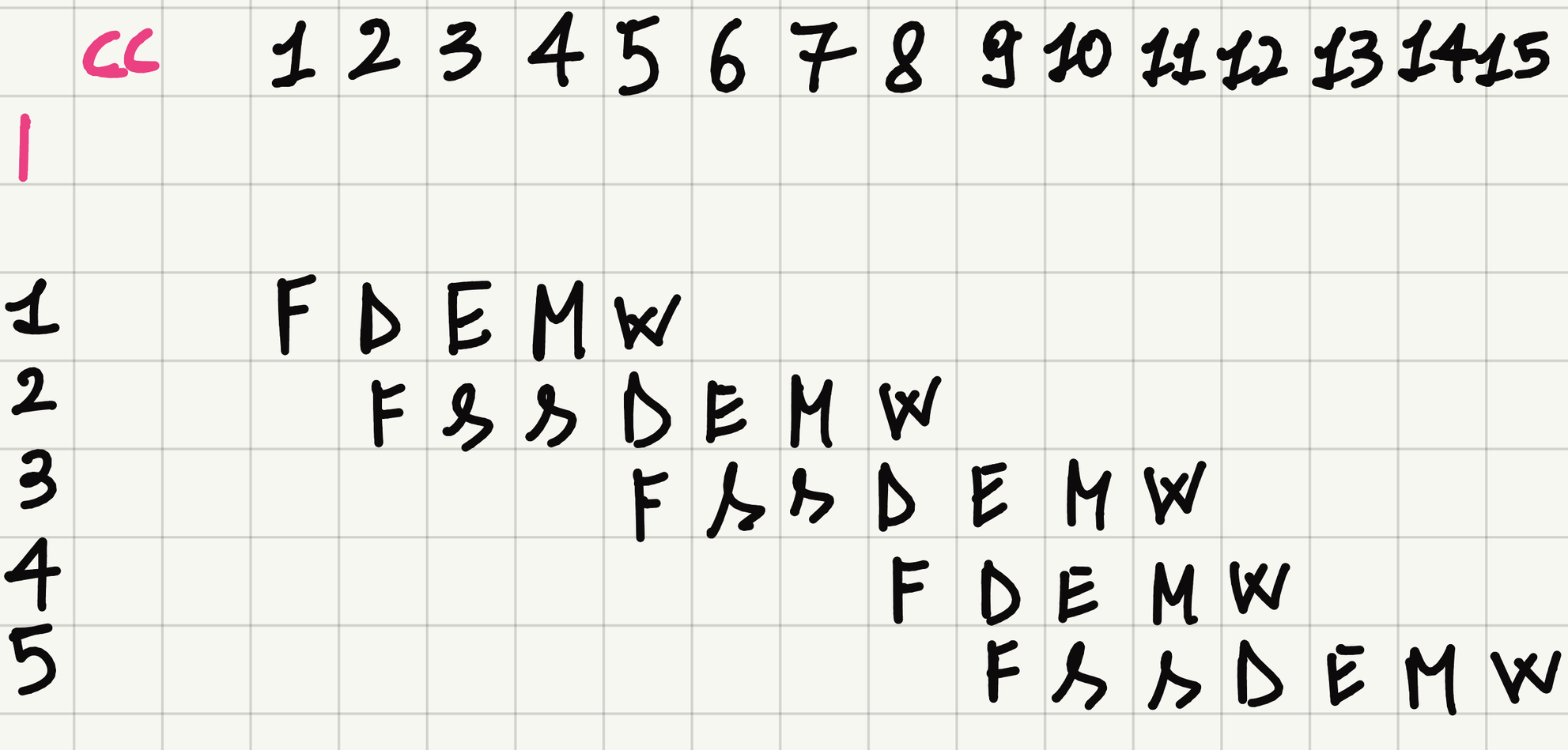
RAW

RAW

T = 2ns

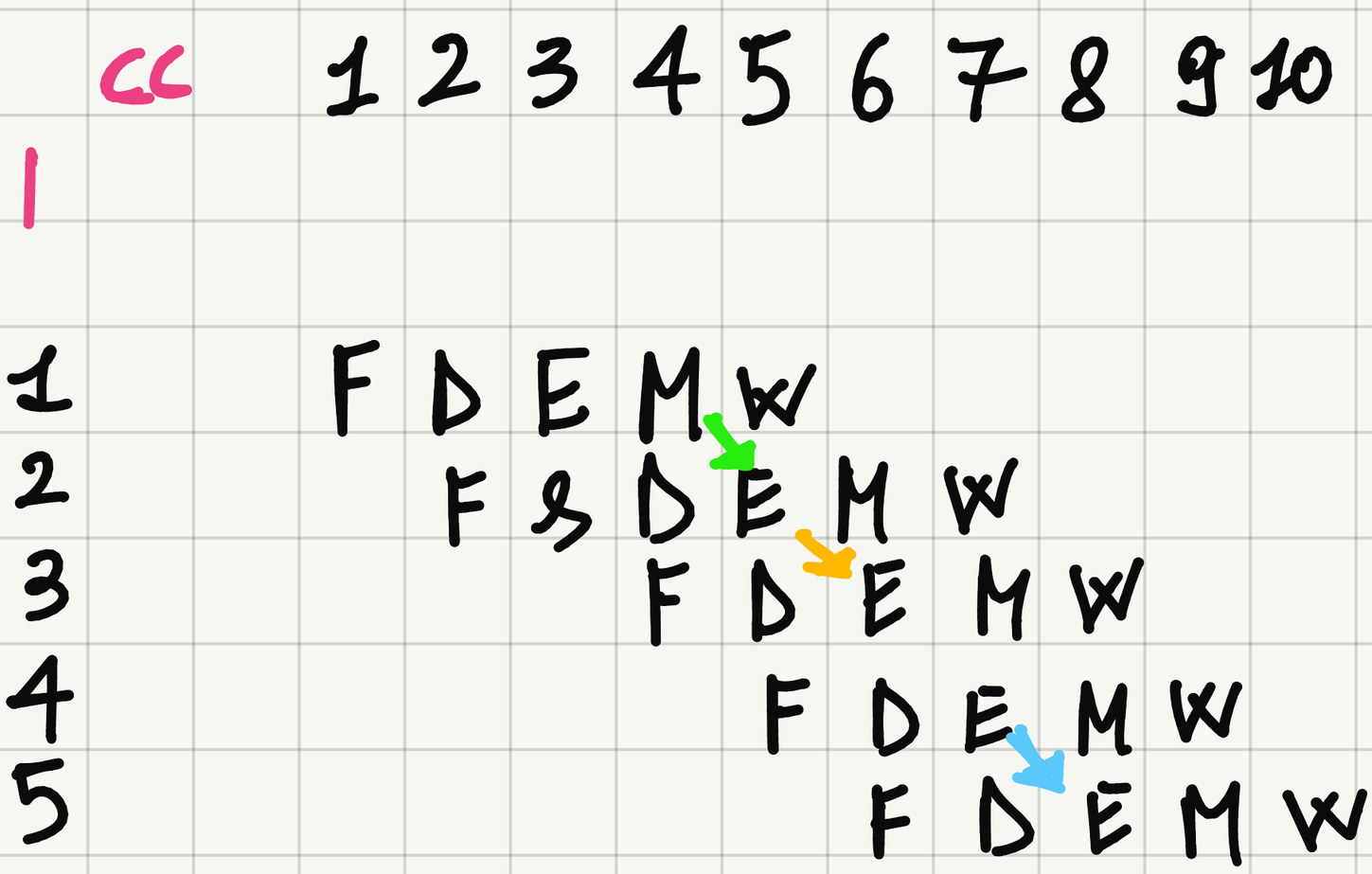
⇓

f = 0,5 · 10⁹ Hz

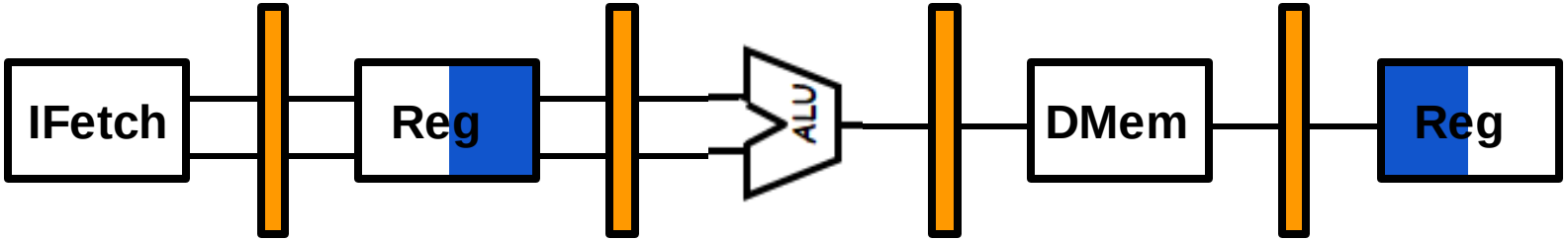


$$CPI = \frac{CC}{\#I} = \frac{15}{5} = 3$$

$$MIPS = \frac{f}{CPI \cdot 10^6} = 166$$



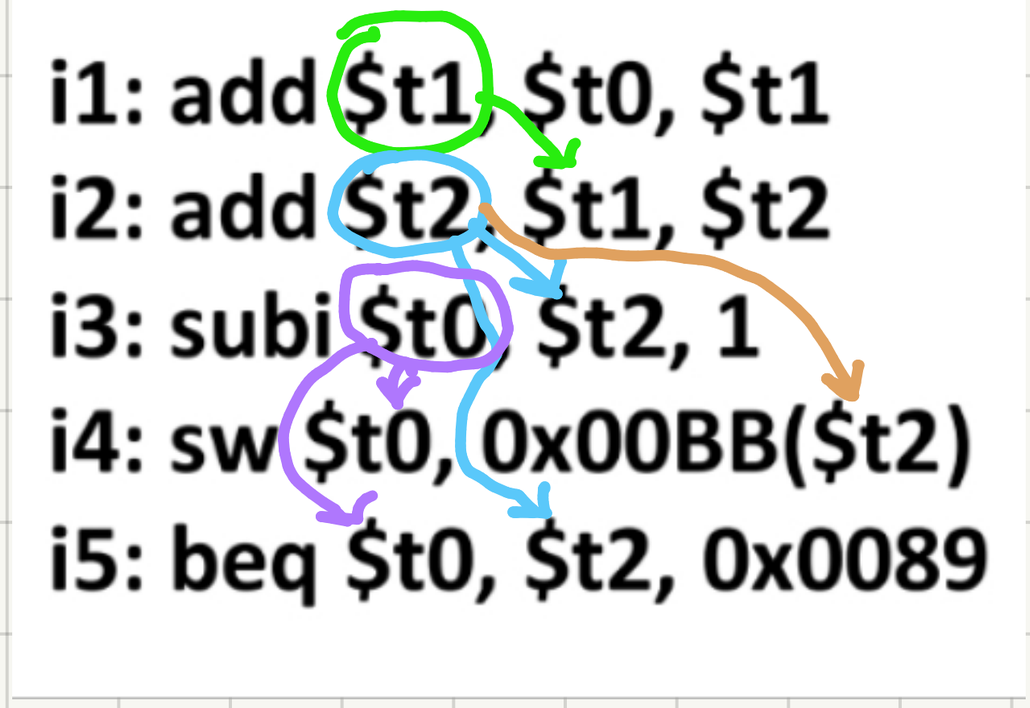
Exe : Pipelining



i1: add $t1, $t0, $t1
i2: add $t2, $t1, $t2
i3: subi $t0, $t2, 1
i4: sw $t0, 0x00BB($t2)
i5: beq $t0, $t2, 0x0089

- **No forwarding** paths
- RF access **R/W optimization**
- **Control** Hazard solved in **ID**

1) **Define all conflicts/dependencies**. For each of them indicate **whether** it causes an **hazard** and the **theoretical** amount of **stalls**
2) Draw the effective **pipeline schema**
3) Assuming EX/EX, MEM/EX, and MEM/MEM **forwarding paths** available + 2)
4) **Assuming** EX/ID + 3)



RAW

RAW

RAW

CC 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

I

1 F D E M W
2 F S S D E M W
3 F S S D E M W
4 F S S D E M W
5 F D E M W

CC 1 2 3 4 5 6 7 8 9 10

I

1 F D E M W
2 F D E M W
3 F D E M W
4 F D E M W
5 F S D E M W

CC 1 2 3 4 5 6 7 8 9

I

1 F D E M W
2 F D E M W
3 F D E M W
4 F D E M W
5 F D E M W

ONCE DATA IS MOVING FROM ONE STAGE TO THE FOLLOWING ONE, WE CANNOT USE ANY LONGER THE PREVIOUS STAGE TO FORWARD IT

Exe 3 Simple Pipelining : the Code

```
I1:  addi $s3, $s2, 2
I2:  add $s5, $s4, $s3
I3:  sw $s5, 4($s3)
I4:  sub $s7, $s5, $s6
I5:  lw $s6, 4($s7)
```

- SHOW ALL CONFLICTS AND DRAW PIPELINE SCHEMA
- RESCHEDULE INSTRUCTIONS TO REDUCE THE NUMBER OF STALLS
- USE PATH FORWARDING TO REDUCE THE NUMBER OF STALLS OF THE STARTING CODE

```
I1:   addi $s3, $s2, 2
I2:   add $s5, $s4, $s3
I3:   sw $s5, 4($s3)
I4:   sub $s7, $s5, $s6
I5:   lw $s6, 4($s7)
```

RAW

RAW

WAR

RAW

cc 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

1

1 F D E M W
2 F S S D E M W
3 F S S D E M W
4 F D E M W
5 F S S D E M W

addi $s3, $s2, 2
add $s5, $s4, $s3
sw $s5, 4($s3)
sub $s7, $s5, $s6
lw $s6, 4($s7)

cc 1 2 3 4 5 6 7 8 9 10 11 12 13 14

1

1 F D E M W
2 F S S D E M W
3 F S S D E M W
4 F D E M W
5 F S D E M W

cc	1	2	3	4	5	6	7	8	9
1									
1	F	D	E	M	W				
2		F	D	E	M	W			
3			F	D	E	M	W		
4				F	D	E	M	W	
5					F	D	E	M	W