Pipeline architecture: example 1

E. Sanchez, M. Sonza Reorda

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

Example 1

Let consider the following MIPS64 architecture:

- **Integer ALU: 1 clock cycle**
- **Data memory: 1 clock cycle**
- **Branch delay slot: 1 clock cycle**
- **Forwarding is enabled.**

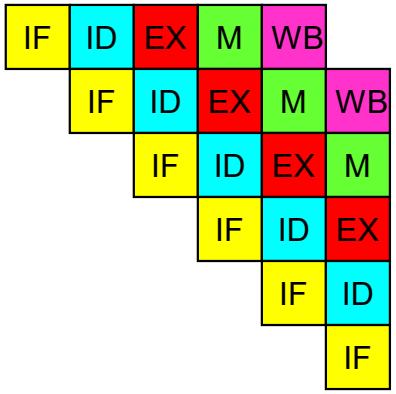
The following assembly program sums 10 integer (64 bits) numbers previously saved in memory.

```
; 10 integers addition
;-------------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10          ;R1 <-- 10
            dadd R2,R0,R0        ;R2 <-- 0 POINTER REG
            dadd R3,R0,R0        ;R3 <-- 0 RESULT REG
LOOP:   ld R4,values(R2)         ;GET A VALUE IN R4
            dadd R3,R3,R4        ;R3 <-- R3 + R4
            daddi R2,R2,8        ;R2 <-- R2 + 8 POINTER INCREMENT
            daddi R1,R1,-1       ;R1 <-- R1 - 1 DECREMENT COUNTER
            bnez R1,LOOP
            sd R3,result(R0)     ;Result in MEM
            halt                 ;the end
```

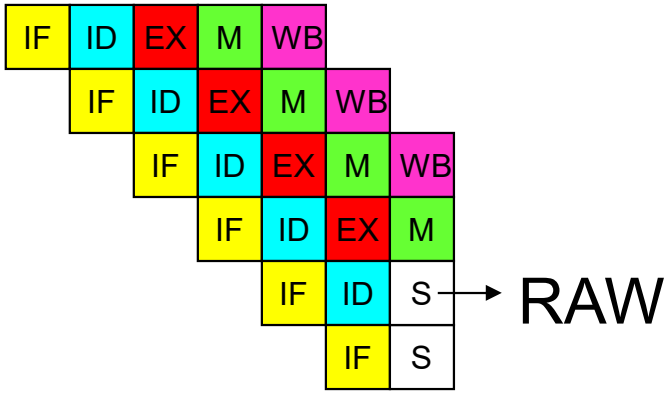
Example 1 – [cont]

You are asked to compute the time (in clock cycles) required by the program to execute.

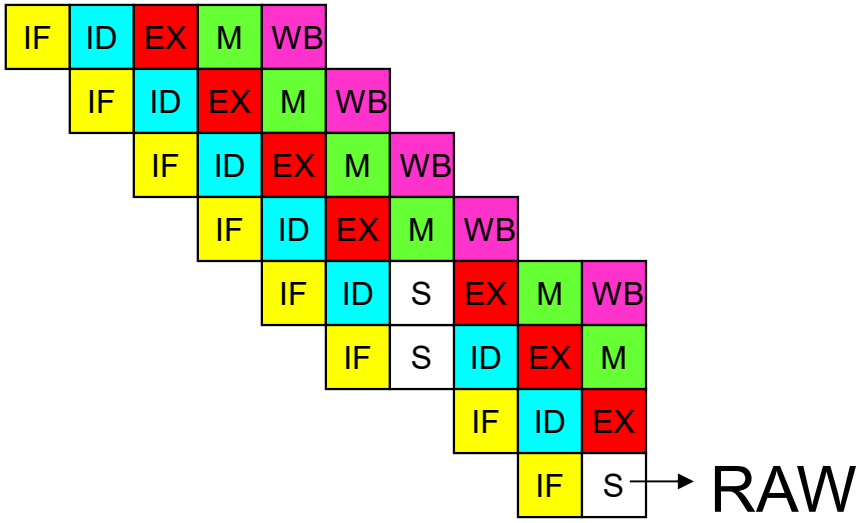
```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
            dadd R2,R0,R0
            dadd R3,R0,R0
LOOP:  ld R4,values(R2)
            dadd R3,R3,R4
            daddi R2,R2,8
            daddi R1,R1,-1
            bnez R1,LOOP
            sd R3,result(R0)
            halt
```



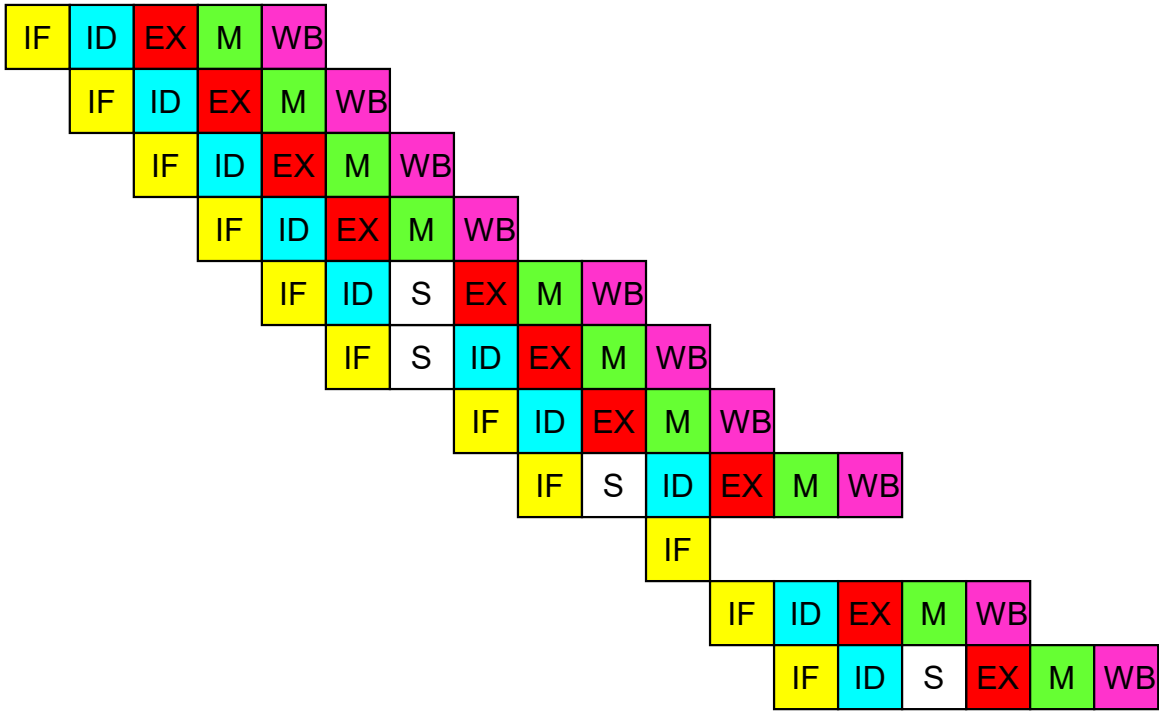
```
; 10 integers addition
;------------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
        dadd R2,R0,R0
        dadd R3,R0,R0
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
        daddi R2,R2,8
        daddi R1,R1,-1
        bnez R1,LOOP
        sd R3,result(R0)
        halt
```



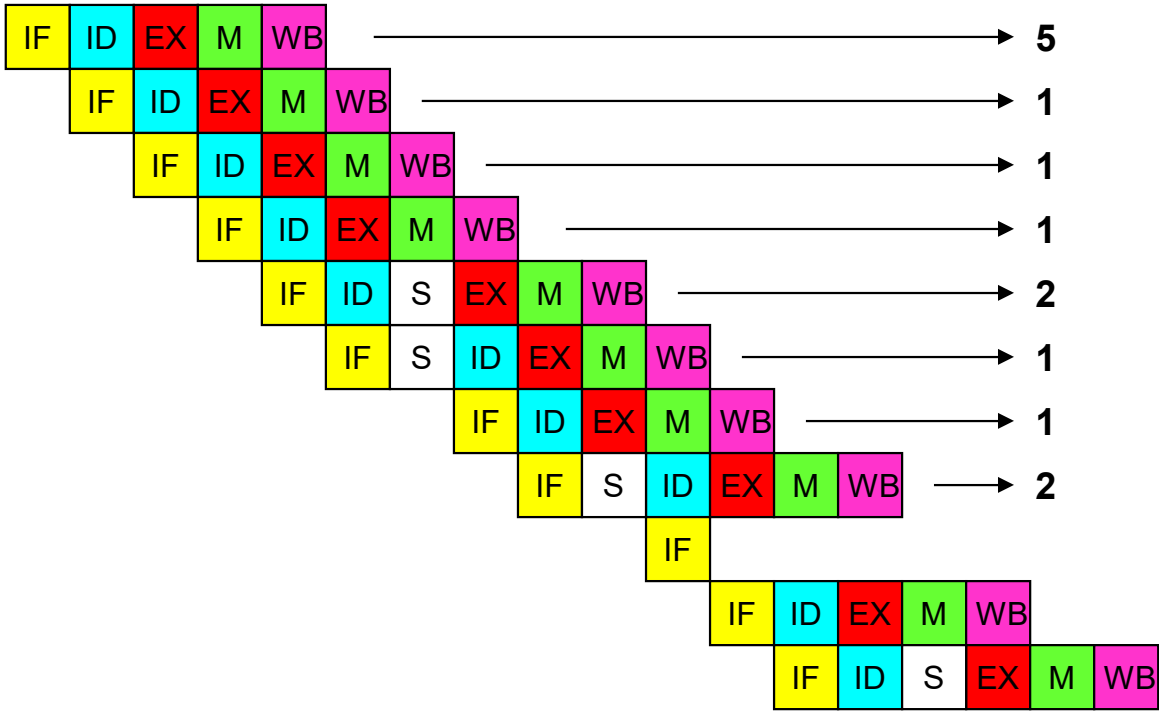
```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
        dadd R2,R0,R0
        dadd R3,R0,R0
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
        daddi R2,R2,8
        daddi R1,R1,-1
        bnez R1,LOOP
        sd R3,result(R0)
        halt
```



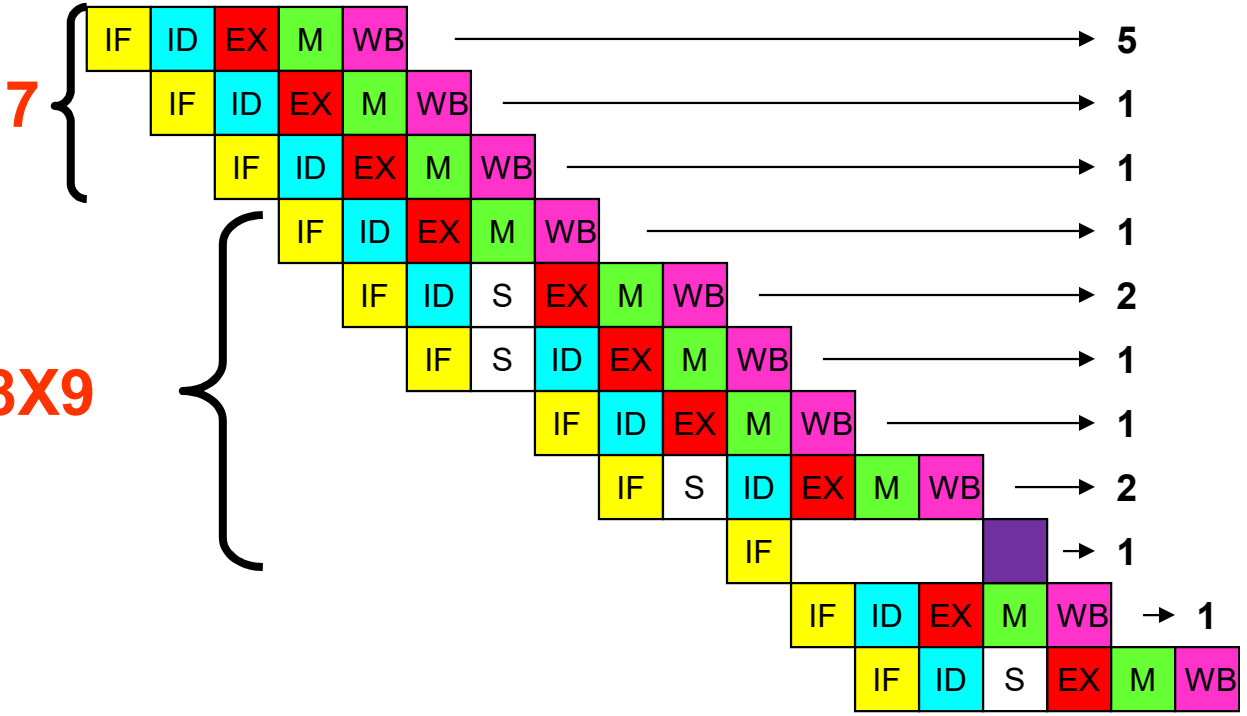
```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
        dadd R2,R0,R0
        dadd R3,R0,R0
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
        daddi R2,R2,8
        daddi R1,R1,-1
        bnez R1,LOOP
        sd R3,result(R0)
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
```



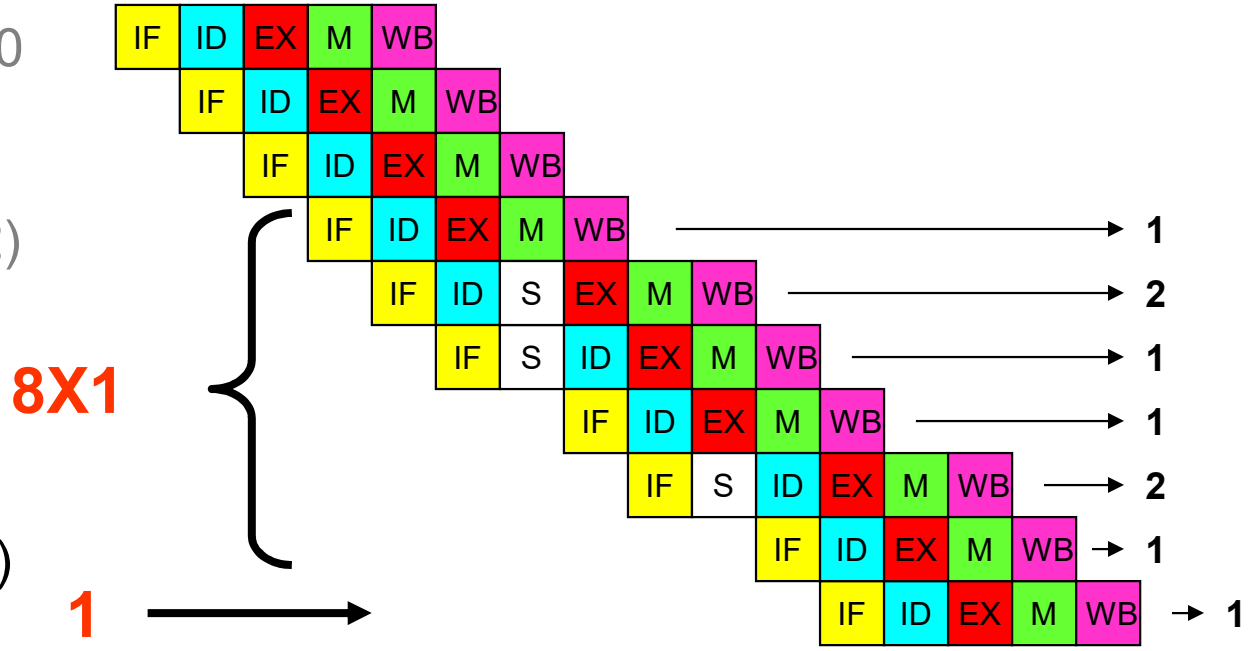
```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
        dadd R2,R0,R0
        dadd R3,R0,R0
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
        daddi R2,R2,8
        daddi R1,R1,-1
        bnez R1,LOOP
        sd R3,result(R0)
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
```



```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
        dadd R2,R0,R0
        dadd R3,R0,R0
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
        daddi R2,R2,8
        daddi R1,R1,-1
        bnez R1,LOOP
        sd R3,result(R0)
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
```



```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10
        dadd R2,R0,R0
        dadd R3,R0,R0
LOOP:   ld R4,values(R2)
        dadd R3,R3,R4
        daddi R2,R2,8
        daddi R1,R1,-1
        bnez R1,LOOP
        sd R3,result(R0)
        halt
```



```
; 10 integers addition
;-----------------------
            .data
values: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ;64-bit integers
result: .space 8
            .text
MAIN:   daddui R1,R0,10     5
        dadd R2,R0,R0       1
        dadd R3,R0,R0       1
LOOP:   ld R4,values(R2)    1
        dadd R3,R3,R4       2
        daddi R2,R2,8       1
        daddi R1,R1,-1      1
        bnez R1,LOOP        2
        sd R3,result(R0)    1
        halt                1
```

7

8X10

1

88

20 RAW