

Exercícios sobre *pipelining* e dependência de dados

1. Simular cada um dos trechos de código abaixo e anotar:
 - tempo sequencial, tempo com *pipelining* e ganho (*speedup*) do trecho de código.
 - as dependências de dados (RAW – *read after write*) existentes no código.
 - usar a técnica de *forwarding/bypassing* do simulador para resolver ou minimizar as dependências de dados. Para isso, estudar as dependências existentes (em quais estágios do *pipelining* elas ocorrem) e escolher as opções de *forwarding* (MEM -> EX, WB -> EX, WB -> MEM) mais adequadas para resolvê-las.
 - anote novamente o tempo sequencial, tempo com *pipelining* e ganho (*speedup*) do trecho de código ajustado com *forwarding*.

2. Questões:

CÓDIGO 1

```
lw  $1, 0($1)
lw  $2, 4($0)
add $3, $1, $2
sw  $3, 12($0)
lw  $4, 8($0)
add $5, $1, $4
sw  $5, 16($0)
```

CÓDIGO 2

```
add $t1, $t0, $t0
addi $t2, $t1, #5
addi $t4, $t1, #5
```

CÓDIGO 3

```
addi $t1, $t0, #1
addi $t2, $t0, #2
addi $t3, $t0, #3
addi $t3, $t0, #4
addi $t5, $t0, #5
```

CÓDIGO 4

```
LW $1 0( $1 )
LW $2 0( $2 )
ADD $3 $1 $2
SW $3 100( $1 )
```

CÓDIGO 5

```
BEQ $4 $5 PTO
ADD $2 $1 $4
PTO: ADD $1 $3 $2
SW $1 100( $1 )
ADD $9 $9 $9
ADD $3 $4 $3
ADD $2 $1 $3
```