



Atividade 05: Pipeline

1) Considerando o programa do arquivo "fibonacci.asm" (link abaixo):

a) Identifique as linhas que possuem conflitos de dados e descreva qual é a causa do conflito.

Linha 7 – RAW – registrador t0 é lido e a escrita da linha 7 não está pronta

Ocorre porque é utilizada a pseudo-instrução LA que faz AUIPC e ADDI em t0

Linha 8 – RAW – registrador t0 é lido e a escrita da linha 7 não está pronta

Linha 11 – RAW - registrador a7 é lido e a escrita da linha 10 não está pronta

Linha 13 – RAW - registrador a7 é lido e a escrita da linha 12 não está pronta

Linha 27 – RAW – registrador t3 é lido e a escrita da linha 26 não está pronta

Linha 34 – RAW – registrador t4 é lido e a escrita da linha 32 não está pronta

b) Identifique as linhas que possuem conflitos de controle e descreva a causa do conflito

Linha 9 – instrução JAL altera o valor de PC provocando o desvio para a linha 24

Linha 27 – instrução BGT pode alterar o valor de PC provocando o desvio para a linha 31

Linha 29 – instrução RET altera o valor de PC provocando o desvio para a linha definida no endereço do registrador ra

Linha 31 – instrução BGT pode alterar o valor de PC provocando o desvio para a linha 38

Linha 36 – instrução J altera o valor de PC provocando o desvio para a linha 31

Linha 39 – instrução RET altera o valor de PC provocando o desvio para a linha definida no endereço do registrador ra

c) Reescreva e apresente a função Fibonacci inserindo NOPs para resolver os conflitos de dados e de controle, considerando que o pipeline não faz detecção e tratamento dos conflitos. PS: deve-se inserir a menor quantidade de NOPs que mantém o programa funcional

```
fibonacci:
    addi t1, zero, 1
    addi t2, zero, 1
    addi t3, zero, 2
```

```

nop
nop
bgt a0, t3, calcula
nop
nop
add s0, zero, t1
ret
nop
nop
calcula:
    beq t3, a0, terminou
    nop
    nop
    add t4, t1, t2
    add t1, zero, t2
    nop
    add t2, zero, t4
    addi t3, t3, 1
    j calcula
    nop
    nop
terminou:
    add a0, zero, t4
    ret

```

d) Apresente o diagrama de tempo do Pipeline para a execução dos 20 primeiros ciclos de clock da função Fibonacci. Considere que o Pipeline com detecção e tratamento de conflitos

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20
addi t1, zero, 1 #fib1	IF	ID	EX	MEM	WB															
addi t2, zero, 1 #fib2		IF	ID	EX	MEM	WB														
addi t3, zero, 2			IF	ID	EX	MEM	WB													
bgt a0, t3, calcula				IF	ID	EX	MEM	WB												
add s0, zero, t1					IF	EX														
ret						IF														
beq t3, a0, terminou							IF	ID	EX	MEM	WB									
add t4, t1, t2 #soma								IF	ID	EX	MEM	WB								
add t1, zero, t2									IF	ID	EX	MEM	WB							
add t2, zero, t4										IF	ID	EX	MEM	WB						
addi t3, t3, 1											IF	ID	EX	MEM	WB					
j calcula												IF	ID	EX	MEM	WB				
add a0, zero, t4													IF	ID						
ret														IF						
beq t3, a0, terminou															IF	ID	EX	MEM	WB	
add t4, t1, t2 #soma																IF	ID	EX	MEM	WB
add t4, t1, t2 #soma																	IF	ID	EX	MEM

e) Em relação a execução da função fibonacci original no modo 5-Stage Processor:

e1) Quantos NOPs são inseridos pelo simulador durante a execução da função devido a conflitos de dados

0 NOPs

No modo 5-Stage Processor os conflitos são todos resolvidos com *forwarding* (para esta implementação da função Fibonacci)

e2) Quantos NOPs são inseridos pelo simulador durante a execução da função devido a conflitos de controle

16 NOPs

São inseridos NOPs quando realiza o FLUSH devido aos desvios, eliminando as instruções já carregadas no pipeline e inserindo NOPs. São inseridos:

- 2 NOPs na linha 27;
- 10 NOPs da linha 36 (2 a cada iteração, ocorrendo 5 vezes);
- 2 NOPs na linha 31;
- 2 NOPs na linha 39.

e3) Qual o total de ciclos de clock que a função leva para executar?

A função Fibonacci entra no pipeline no ciclo de clock 6, e sua última instrução sai do pipeline no ciclo de clock 61, totalizando 55 ciclos de clock para ser completada.

e4) Quantas instruções são executadas?

A primeira instrução da função Fibonacci a ser completada ocupa a 4ª posição, e a última ocupa a 41ª posição. Portanto a função Fibonacci é executada em 37 instruções.

e5) Qual a CPI da função?

$CPI = 1,486$

$CPI = \text{número de ciclos de clock} / \text{número de instruções finalizadas} = 55 / 37$