ARQUITETURA DE COMPUTADORES

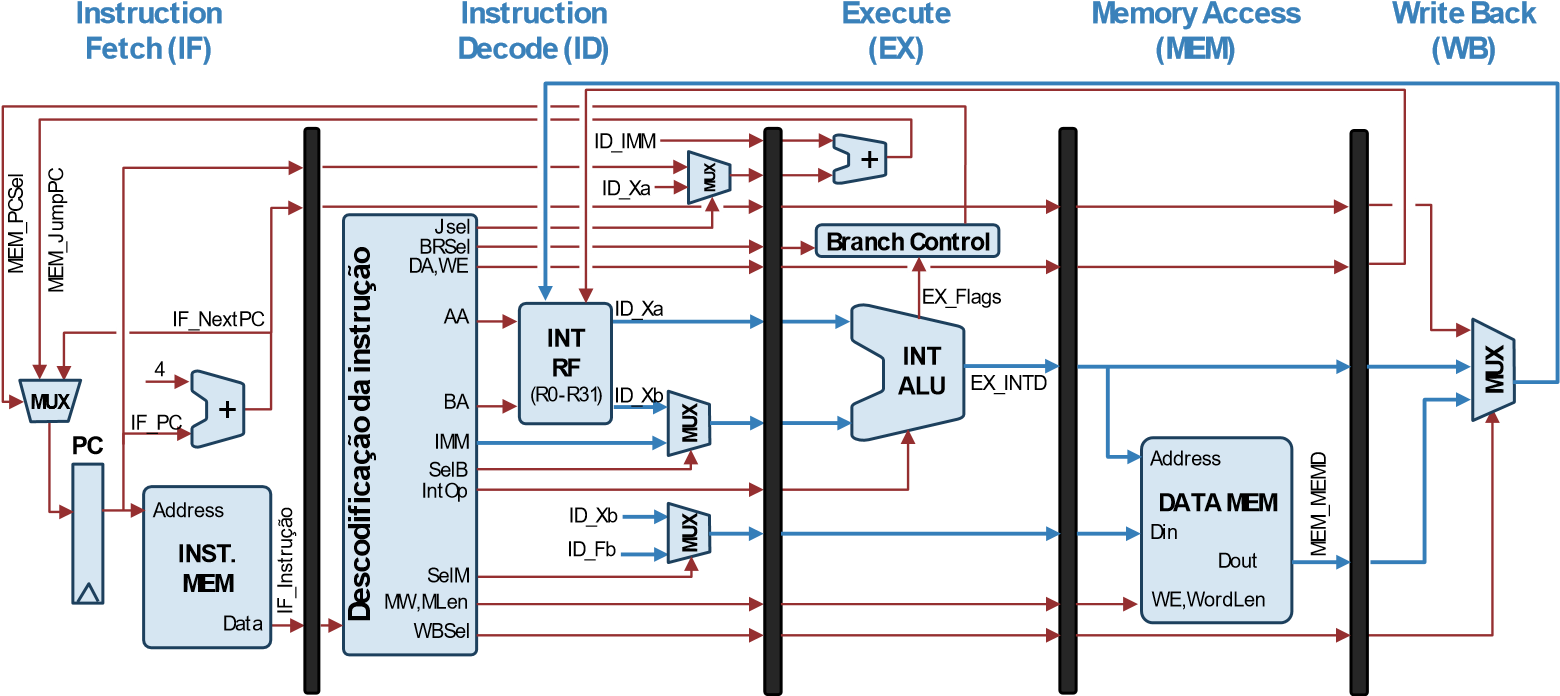
2020-2021

Laboratório 4 – Análise de um Processador Pipelined

Arquitetura do Processador Pipelined

Maria Costa nº100022

Tomás Marques nº100104

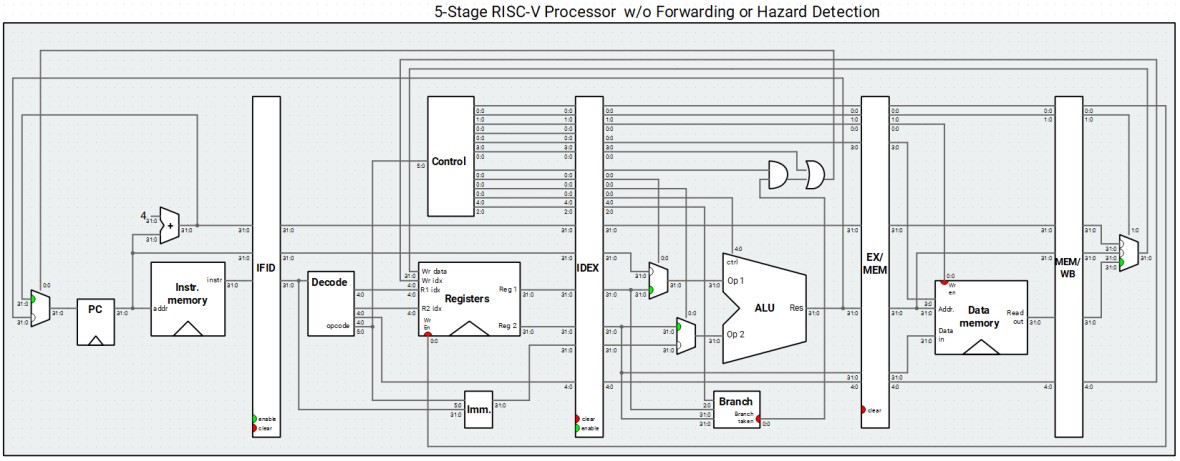


**Figura 1 - Arquitetura do Processador.**

# Exercício 1

1. Configure o simulador de forma que este simule a arquitetura pipeline mais simples, que não realiza o adiantamento de dados (*forwarding*) entre estágios do *pipeline* nem a deteção de dependências de controlo. Para tal, clique sobre o símbolo do processador do canto superior esquerdo e selecione a seguinte opção (selecionando também o Layout “Extended” para que consiga ver os detalhes da implementação do pipeline):

Select Processor → 5-Stage Processor w/o forwarding or hazard detection



**Figura 3 – Arquitetura do processador em pipeline sem adiantamento de dados (forwarding) entre estágios do pipeline nem deteção de dependências de controlo.**

1. Carregue o programa L4.s que lhe foi fornecido.
2. Identifique os endereços de início dos vetores a[] e b[]. Deverá/poderá fazê-lo através de dois métodos:
   * Por inspeção do código (sugestão: procure identificar os valores que deverão ser escritos nos registos x11 e x13
   * Por observação da zona de Memória do simulador (não se esqueça de selecionar a secção .data)
3. Coloque um *Break-Point* na primeira instrução do ciclo “while” e execute o programa até esse instante, pressionando o botão >>
4. Compare os valores nos registos x11 e x13 com os valores que identificou anteriormente. São iguais? Porquê?

Identifique a sua resposta no documento Word que vai entregar com o número **Q1.5**

Por inspeção do código:

X11= 268435456

X13= 268435456 + 48 = 268,435,504

Apos executar com o break-point:

X11=268435456

X13= 48

Porque quando a instrução addi x13, x11, 48 foi executada o valor de x11 ainda não tinha sido

escrito no registo.

1. Tendo em consideração a arquitetura do processador, identifique todos os conflitos de dados, indicando a instrução que escreve e a instrução que lê e qual o operando que provoca o conflito.

Identifique a sua resposta no documento Word que vai entregar com o número **Q1.6**

Azul- conflito de dados  
Vermelho- conflito de controlo

.text

addi x11, x3, 0

addi x13, x11, 48 #escreve em x13 antes de o valor de x11 ter sido escrito no registo

addi x12, x13, -4 #escreve em x12 antes de o valor de x13 ter sido escrito no registo

lw x14, 100(x3)

lw x15, 96(x3)

li x16, 0

while: add x20, x13, x16 #escreve em x20 antes de o valor de x16 ter sido escrito no registo

lw x21, 0(x20) #escreve em x21 antes de o valor de x20 ter sido escrito no registo

blez x21, end

lw x22, 0(x11)

lw x23, 0(x12)

add x22, x22, x23 #escreve em x22 antes de o valor de x22 e x23 terem sido escritos nos registos

mul x15, x15, x22 #escreve em x15 antes de o valor de x22 ter sido escrito no registo

sub x22, x12, x11

srai x22, x22, 2 #escreve em x22 antes de o valor de x22 ter sido escrito no registo

add x14, x14, x22 #escreve em x22 antes de o valor de x16 ter sido escrito no registo

addi x16, x16, 4

addi x11, x11, 4

addi x12, x12, -4

jal x0, while

end: sw x14, 100(x3)

sw x15, 96(x3)

addi a7, x0, 10

ecall

1. Altere o código do preâmbulo do programa (antes do ciclo “while”) de modo a garantir o correto funcionamento, assegurando a resolução de todos os conflitos de dados.
2. Compare agora os valores nos registos x11 e x13 com os valores que identificou anteriormente. São iguais? Porquê?

Identifique a sua resposta no documento Word que vai entregar com o número **Q1.8**

.text

addi x11, x3, 0 # Done

lw x14, 100(x3) # Done

lw x15, 96(x3) # Done

li x16, 0 # Done

addi x13, x11, 48 # Done

nop # WB

nop # MEM

nop # EX

addi x12, x13, -4 # ID

while: add x20, x13, x16 # IF

Por inspeção do código:

X11= 268435456

X13= 268435456 + 48 = 268,435,504

Apos executar com o break-point:

X11= 268435456

X13= 268435456 + 48 =268,435,504

O valor de x11 já foi escrito quando a instrução foi executada.

1. Aplique a mesma alteração de código que fez no preâmbulo ao resto do programa, de modo a garantir o seu correto funcionamento.
2. Execute o programa completo, verificando se os resultados dos cálculos correspondem aos valores esperados (ver comentário final, no código fonte).
3. Registe as seguintes estatísticas de execução:

Identifique a sua resposta no documento Word que vai entregar com o número **Q1.11**

* + Número de ciclos de relógio: 211
  + Número de instruções executadas: 191

1. Como comenta a execução do programa neste processador, no que diz respeito à sua eficiência? Indique o rácio entre instruções úteis e não úteis do programa a que chegou.

Identifique a sua resposta no documento Word que vai entregar com o número **Q1.12**

Mais eficiente

99/92

1. Volte a executar este programa e analise a sua execução de modo a determinar a política de predição de salto adotada por este simulador. Justifique.

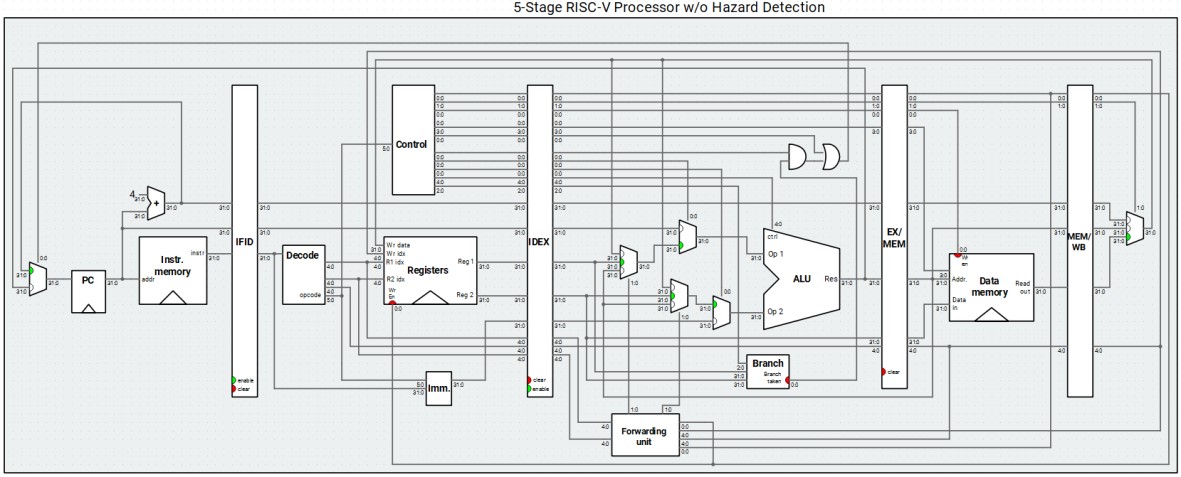
Identifique a sua resposta no documento Word que vai entregar com o número **Q1.13**

Static Branch Prediction: Predict not taken

# Exercício 2

1. Carregue agora o modelo de processador correspondente a "5-Stage Processor with forwarding but no hazard detection/elimination", selecionando também o Layout “Extended” para que consiga ver os detalhes da implementação do pipeline. Conforme pode observar, este modelo distingue-se do anterior pelo facto de contemplar já os circuitos de encaminhamento de dados (*forwarding*) dos estágios MEM e WB para o estágio EXE. Contudo, estes mecanismos de forwarding não contemplam as instruções de controlo (i.e., para estas instruções os conflitos de dados não são resolvidos), nem a introdução de stalls quando os conflitos de dados não são solúveis por forwarding.

Select Processor → 5-Stage Processor w/o hazard detection



**Figura 4 – Arquitetura do processador em pipeline com adiantamento de dados (forwarding) entre estágios do pipeline mas sem deteção de dependências de controlo.**

1. Carregue o programa L4.s que lhe foi fornecido (versão original) e modifique-o de modo a garantir o seu correto funcionamento.
2. Quais foram as alterações que introduziu? Porquê?

Identifique a sua resposta no documento Word que vai entregar com o número **Q2.3**

.text

# NOTE: Upon start, the Global-Pointer (gp=x3) points to the beginning of .data section

addi x11, x3, 0 # x11 - a's left index

addi x13, x11, 48 # x13 - b's left index

addi x12, x13, -4 # x12 - a's right index

lw x14, 100(x3) # x14 - n - index distance accumulator

lw x15, 96(x3) # x15 - x

li x16, 0 # x16 - i

while: add x20, x13, x16 # x20 = &b[i]

lw x21, 0(x20) # x21 = b[i]

lw x22, 0(x11) # x22 = a[i]

lw x23, 0(x12) # x23 = a[N-1-i]

blez x21, end # if b[i] <= 0 end the loop

add x22, x22, x23 # x22 = a[i] + a[N-1-i]

mul x15, x15, x22 # x15 = x15\*x22 (x \*= a[i] + a[N-1-i])

sub x22, x12, x11 # x22 = 4\*((N-1-i)-i)

srai x22, x22, 2 # x22 = x22/4

add x14, x14, x22 # n += x22

addi x16, x16, 4 # i++

addi x11, x11, 4

addi x12, x12, -4

jal x0, while

nop

nop

end: addi a7, x0, 10

sw x14, 100(x3) # store n's final value

sw x15, 96(x3) # store x's final value

nop

ecall

Colocamos nops após o jal porque a arquitetura não apresenta predição de salto o que gera um conflito de controlo e são executadas sempre mais 2 instruções após o jal.

1. Registe as seguintes estatísticas de execução:

Identifique a sua resposta no documento Word que vai entregar com o número **Q2.4**

* + Número de ciclos de relógio: 134
  + Número de instruções executadas: 114

1. Como comenta a execução do programa neste processador, no que diz respeito à sua eficiência? Indique o rácio entre instruções úteis e não úteis do programa a que chegou.

Identifique a sua resposta no documento Word que vai entregar com o número **Q2.5**

99/15

1. Comparando estas estatísticas com as observadas no exercício anterior, calcule o Speedup conseguido com este modelo de processador. Admita que a frequência de operação não foi alterada com a introdução dos mecanismos de *forwarding*.

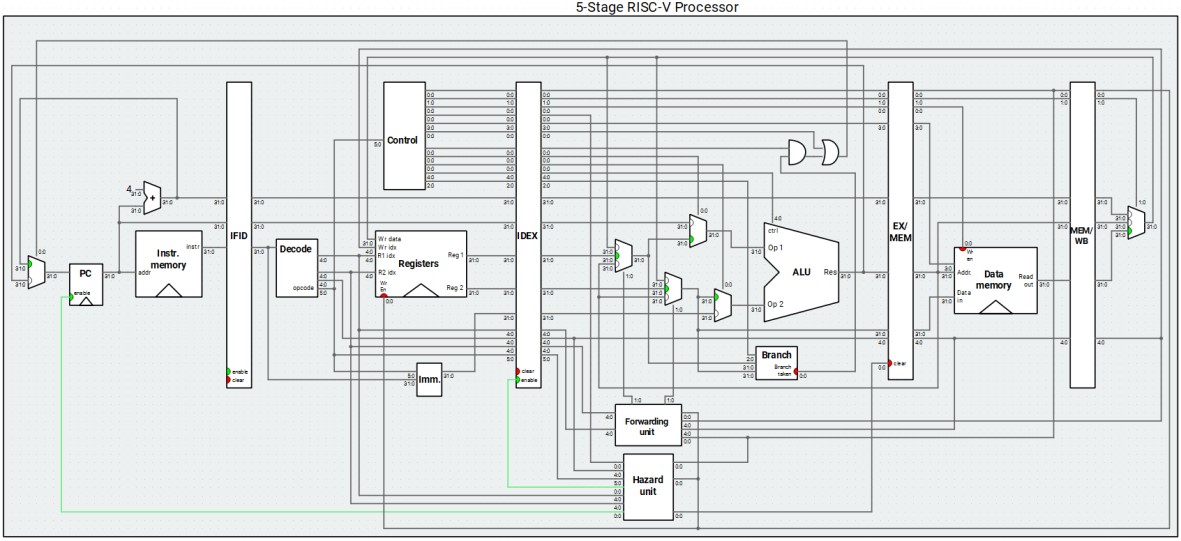
Identifique a sua resposta no documento Word que vai entregar com o número **Q2.6**

0.905/0.851=1,063

# Exercício 3

1. Carregue agora o modelo de processador correspondente a "5-Stage Processor", selecionando também o Layout “Extended” para que consiga ver os detalhes da implementação do pipeline. Conforme pode observar, este modelo distingue-se do anterior pelo facto de introduzir os circuitos de encaminhamento de dados (*forwarding*) para o bloco de resolução de salto e ainda de um mecanismo que permite identificar quando os conflitos de dados não são resolúveis por *forwarding*.

Select Processor → 5-Stage Processor



**Figura 5 – Arquitetura do processador em pipeline com adiantamento de dados (forwarding) entre estágios do pipeline e com deteção de dependências de controlo.**

1. Carregue o programa L4.s que lhe foi fornecido (versão original) e modifique-o de modo a garantir o correto funcionamento do circuito, procurando corrigir todas as anomalias (não se preocupe em introduzir otimizações de desempenho).
2. Quais foram as alterações que introduziu? Porquê?

Identifique a sua resposta no documento Word que vai entregar com o número **Q3.3**

Não foram realizadas alterações para o correto funcionamento do circuito.

1. Registe as seguintes estatísticas de execução:

Identifique a sua resposta no documento Word que vai entregar com o número **Q3.4**

* + Número de ciclos de relógio: 148
  + Número de instruções executadas: 111

1. Como comenta a execução do programa neste processador, no que diz respeito à sua eficiência? Indique o rácio entre instruções úteis e não úteis do programa a que chegou.

Identifique a sua resposta no documento Word que vai entregar com o número **Q3.5**

111/0

1. Calcule o número médio de instruções executadas por ciclo de relógio (IPC). Porque razão este valor é inferior a 1?

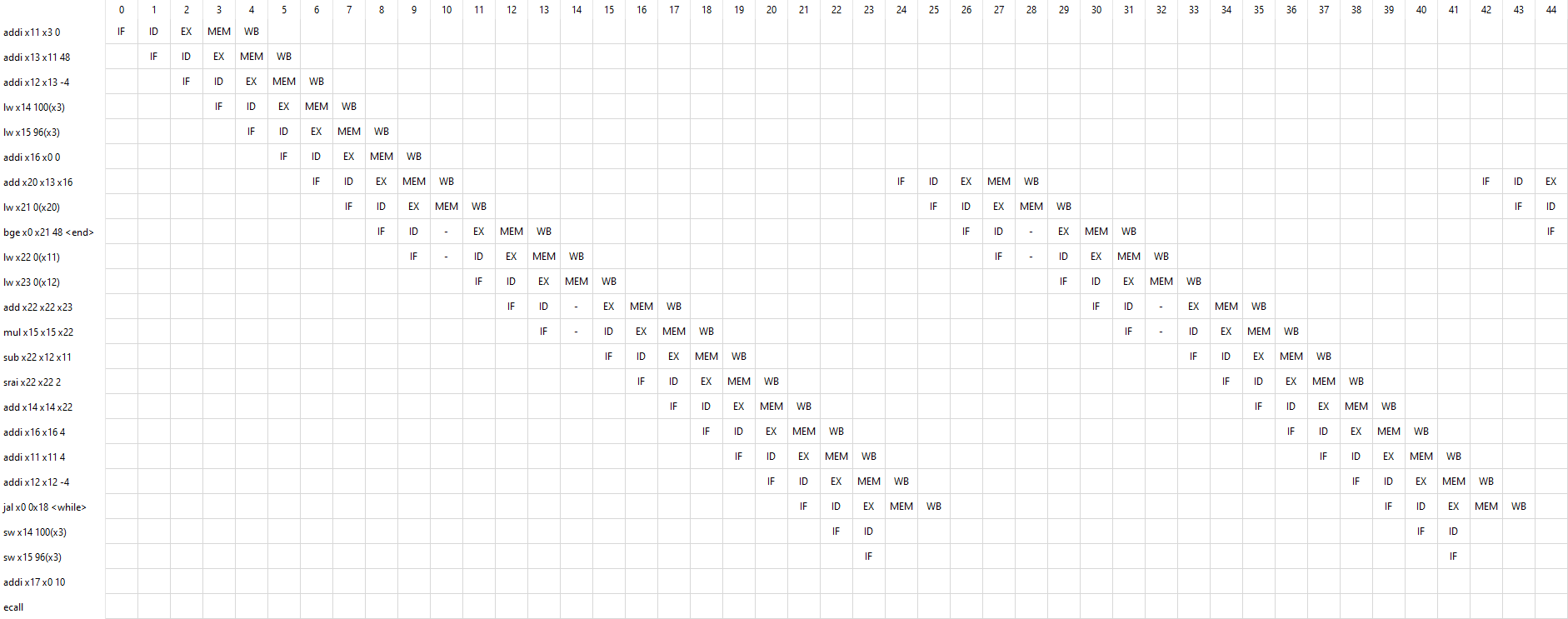
Identifique a sua resposta no documento Word que vai entregar com o número **Q3.6**

0.75

As instruções são divididas em 5 estágios e cada estágio demora um ciclo de relógio, no entanto por vezes o valor dos registos ainda não foi calculado para poder ser utilizado e é necessário aguardar um ciclo para que seja possível fazer forwarding desse valor. Sendo assim, algumas instruções demoram mais que 5 ciclos até serem completamente executadas.

1. Volte a executar o programa, carregando agora na tecla F6 (“*Clock the circuit with the selected frequency*”).
2. Após terminar a execução, pressione o botão “*Show Stage Table*”.
3. Copie a parte da tabela correspondente às duas primeiras iterações do ciclo “while”.

Identifique a sua resposta no documento Word que vai entregar com o número **Q3.9**



1. Por inspeção desta tabela, indique:

Identifique a sua resposta no documento Word que vai entregar com o número **Q3.10**

* + Número de stalls introduzidos em consequência de conflitos de dados RAW: 15
  + Número de stalls introduzidos em consequência de conflitos de controlo: 1

1. Com base nesta observação, procure identificar a razão para o facto de a métrica IPC calculada anteriormente ter sido inferior a 1.

Identifique a sua resposta no documento Word que vai entregar com o número **Q3.11**

Porque em certos casos são induzidos stalls que não contam como um estágio de pipeline, logo 1 instrução acaba por demorar mais que 5 ciclos até ser terminada.

# Exercício 4

1. Utilizando o mesmo modelo de processador que considerou no exercício anterior ("5-Stage Processor"), modifique o programa L4.s utilizando técnicas de reordenação da sequência de instruções do programa de modo a minimizar o número de ciclo de relógios necessários à sua execução, otimizando assim o seu desempenho.
2. Quais foram as alterações que introduziu? Porquê?

Identifique a sua resposta no documento Word que vai entregar com o número **Q4.2**

while: add x20, x13, x16

lw x21, 0(x20)

addi x16, x16, 4

blez x21, end

lw x22, 0(x11)

lw x23, 0(x12)

addi x11, x11, 4

add x22, x22, x23

mul x15, x15, x22

sub x22, x12, x11

srai x22, x22, 2

add x14, x14, x22

addi x12, x12, -4

jal x0, while

Reordenação das funções com o objetivo de diminuir o número de STALLS necessários.

1. Registe as seguintes estatísticas de execução:

Identifique a sua resposta no documento Word que vai entregar com o número **Q4.3**

* + Número de ciclos de relógio: 134
  + Número de instruções executadas: 112

1. Calcule o número médio de instruções executadas por ciclo de relógio (IPC). De que forma este parâmetro foi alterado, face ao que observou no exercício anterior?

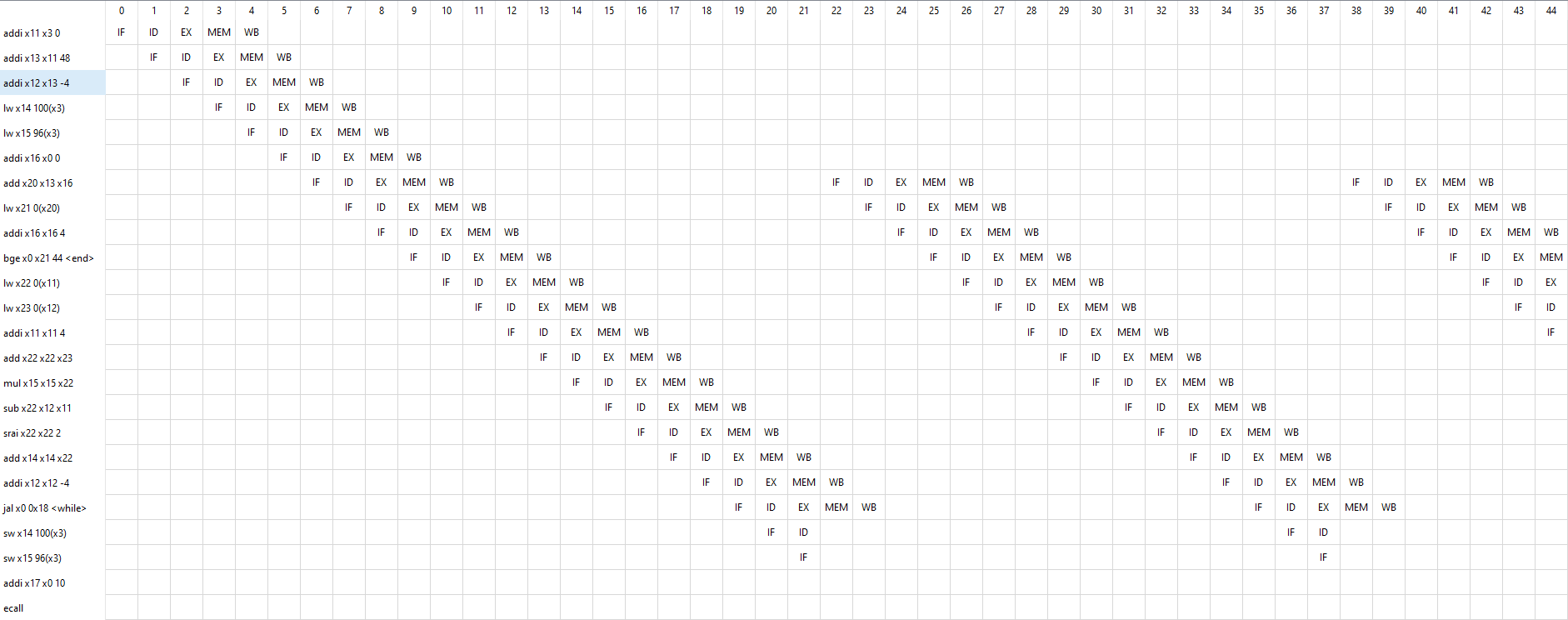
Identifique a sua resposta no documento Word que vai entregar com o número **Q4.4**

0.836

Aumentou

1. Volte a executar o programa, carregando agora na tecla F6 (“Clock the circuit with the selected frequency”).
2. Após terminar a execução, pressione o botão “*Show Stage Table*”.
3. Copie a parte da tabela correspondente às duas primeiras iterações do ciclo “while”.

Identifique a sua resposta no documento Word que vai entregar com o número **Q4.7**



1. Por inspeção desta tabela, indique:

Identifique a sua resposta no documento Word que vai entregar com o número **Q4.8**

* + - Número de stalls introduzidos em consequência de conflitos de dados RAW: 0
    - Número de stalls introduzidos em consequência de conflitos de controlo: 1

1. Com base nesta observação, procure identificar a razão para o facto de a métrica IPC calculada anteriormente ainda ter sido inferior a 1.

Identifique a sua resposta no documento Word que vai entregar com o número **Q4.9**

Porque no jump final é induzido um stalls que não conta como um estágio de pipeline, logo 1 instrução acaba por demorar mais que 5 ciclos até ser terminada.