ARQUITETURA DE COMPUTADORES

2020-2021

Laboratório 4 – Análise de um Processador Pipelined

Este laboratório destina-se a consolidar conhecimentos lecionados sobre as arquiteturas de processadores em pipeline, utilizando o simulador Ripes (https://github.com/mortbopet/Ripes).

O trabalho deve ser realizado fora do horário de laboratório, destinando-se este à demonstração e avaliação do trabalho realizado. Ao longo do enunciado serão apresentadas várias questões às quais os alunos deverão responder (de forma sucinta) num documento Word, que irão submeter através do Fénix, identificando de forma clara a questão à qual estão a dar a resposta. Exemplo:

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

No horário de laboratório serão fornecidos exercícios adicionais que devem ser realizados durante a aula. No final da aula de laboratório deverá submeter no Fénix um ficheiro zip com: um documento Word com as repostas às perguntas do guia e todos os ficheiros Assembly correspondentes às diferentes versões dos programas concebidos.

Arquitetura do Processador Pipelined

O simulador Ripes contempla a simulação de várias variantes da arquitetura de processador em pipeline, tendo como base a arquitetura ilustrada na Figura 1, correspondente a um *pipeline* de 5 estágios (IF, ID, EX, MEM e WB) e barramentos de 32 bits para endereçar as instruções e os dados. A unidade de armazenamento é composta por um banco de 32 registos de inteiros de 32 bits (x0-x31), tal como descrito nas aulas teóricas.

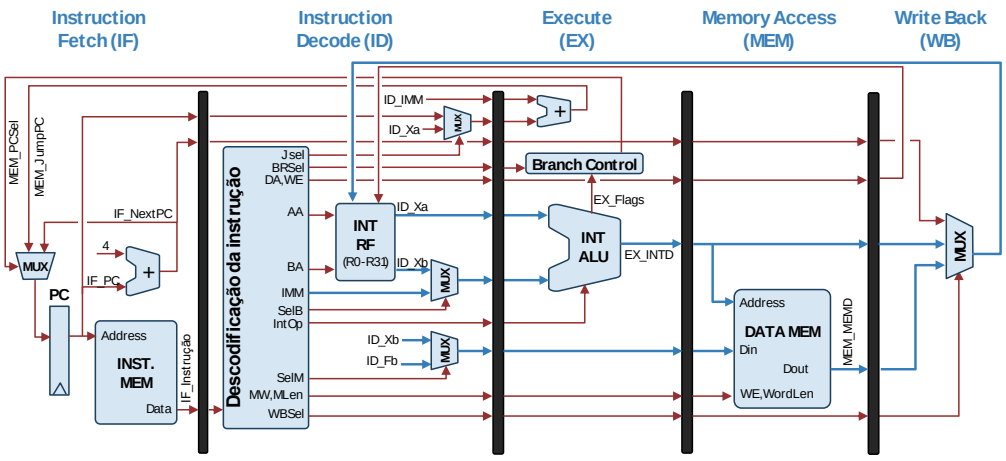

Figura 1 - Arquitetura do Processador.

Ao longo do trabalho irá ser necessário alterar a arquitetura em pipeline considerada. Para o efeito, deverá configurar o simulador de forma que este simule a arquitetura pretendida. Para tal, deverá clicar sobre o símbolo do processador do canto superior esquerdo da janela e selecionar a opção desejada (ver Figura 2).

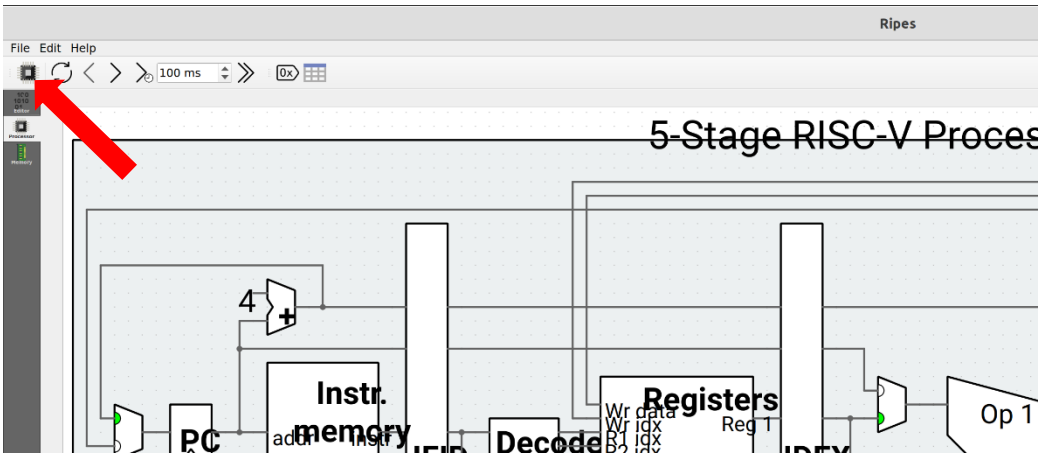

Figura 2 – Seleção da arquitetura do processador

Em qualquer das arquiteturas disponibilizadas, existe ainda a possibilidade de selecionar o Layout "Extended" para que consiga ver os detalhes da implementação do pipeline.

Programa a Executar

Uma dada aplicação de processamento de sinal utiliza um determinado algoritmo, representado através do seguinte troço (em pseudo-código):

```
#define  N  12

int a[N] = { ..... }
int b[N] = { ..... }
int x = 1;
int n = 0;
register int i = 0;      # a keyword "register" aloca a variável em registo

while (b[i]>0){
    x *= a[i] + a[N-1-i];
    n += (N-1-i) - i;
    i ++;
}
```

Este algoritmo processa dois vetores unidimensionais, constituídos por N elementos inteiros. A figura seguinte representa o código Assembly correspondente, utilizando o ISA RV32IM. Este código foi-lhe fornecido no ficheiro "`L4.s`".

```
# Data section
        .data
a:      .word  1, 5, 6, 6, 7, 2, 5, 2, 3, 2, 3, 4
b:      .word  4, 3, 1, 7, 2, 4, 9, -3, 5, 8, 1, 9
x:      .word  1
n:      .word  0

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

        lw      x22, 0(x11)       # x22 = a[i]
        lw      x23, 0(x12)       # x23 = a[N-1-i]
        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

end:    sw      x14, 100(x3)      # store n's final value
        sw      x15, 96(x3)       # store x's final value

        addi    a7, x0, 10
        ecall


# Expected result: M[x] = 1270080 = 136140h
#                  M[n] = 35      = 23h
```

Para fins meramente exemplificativos, o presente código considera um conjunto de dados pré-preenchidos nos vetores a processar, em que cada elemento foi representado com uma palavra de 32-bits (`int`).

NOTA: De modo a facilitar a compreensão dos conceitos que irão ser discutidos ao longo deste trabalho, o código Assembly deste programa foi devidamente preparado de modo a evitar a utilização de pseudo-instruções. Em particular, a generalidade dos endereços das estruturas de dados acedidas estão indexados ao início da secção .data, cujo endereço é passado no início do programa através do registo x3 (Global-Pointer).

Exercício 1

1. Configure o simulador de forma que este simule a arquitetura pipeline mais simples, que <u>não</u> realiza o adiantamento de dados (*forwarding*) entre estágios do *pipeline* <u>nem</u> 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`

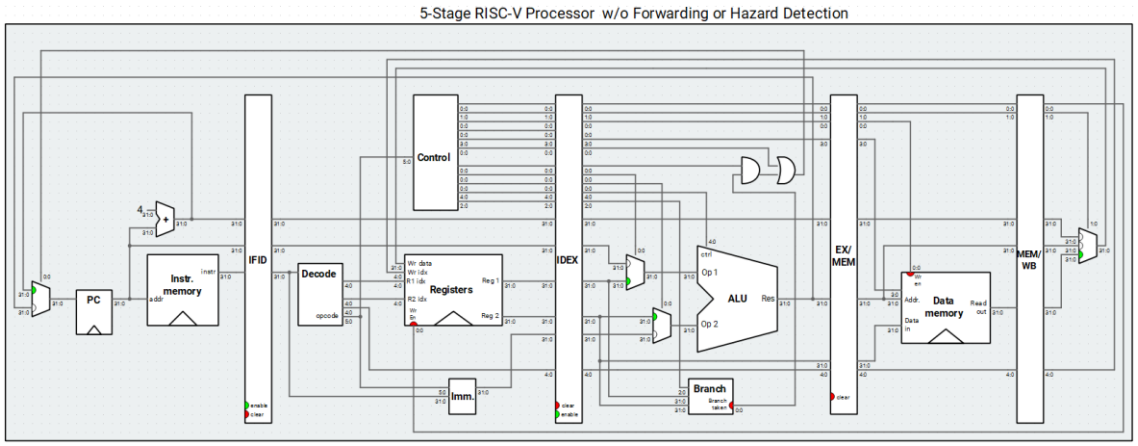

Figura 3 – Arquitetura do processador em pipeline sem adiantamento de dados (forwarding) entre estágios do pipeline nem deteção de dependências de controlo.

2. Carregue o programa L4.s que lhe foi fornecido.
3. 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)
4. Coloque um *Break-Point* na primeira instrução do ciclo "while" e execute o programa até esse instante, pressionando o botão >>
5. 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**

6. 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**

7. 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.
8. 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**

9. Aplique a mesma alteração de código que fez no preâmbulo ao resto do programa, de modo a garantir o seu correto funcionamento.

10. Execute o programa completo, verificando se os resultados dos cálculos correspondem aos valores esperados (ver comentário final, no código fonte).
11. 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:
    - Número de instruções executadas:

12. 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**

13. 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**

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`

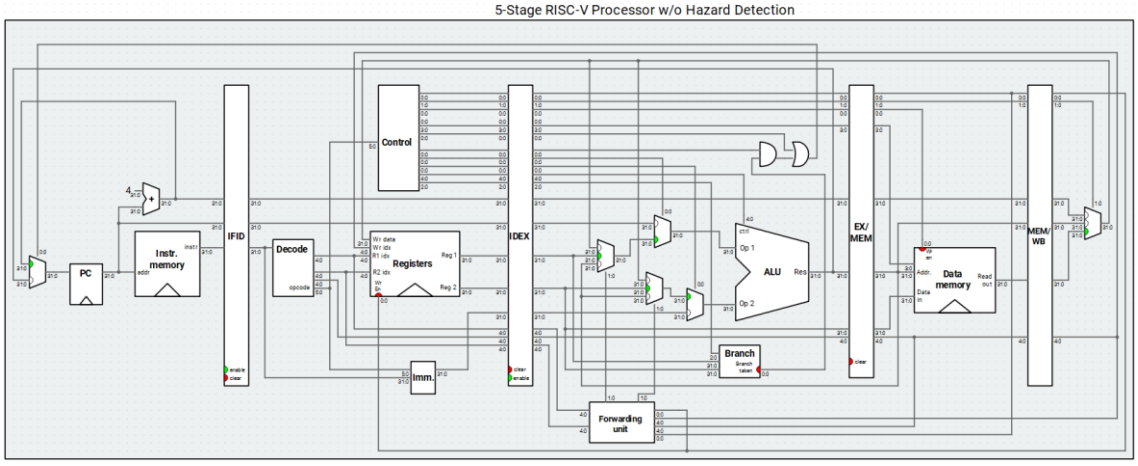

Figura 4 – Arquitetura do processador em pipeline com adiantamento de dados (forwarding) entre estágios do pipeline mas sem deteção de dependências de controlo.

2. Carregue o programa L4.s que lhe foi fornecido (versão original) e modifique-o de modo a garantir o seu correto funcionamento.

3. Quais foram as alterações que introduziu? Porquê?

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

4. 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:
   - Número de instruções executadas:

5. 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**

6. 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**

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`

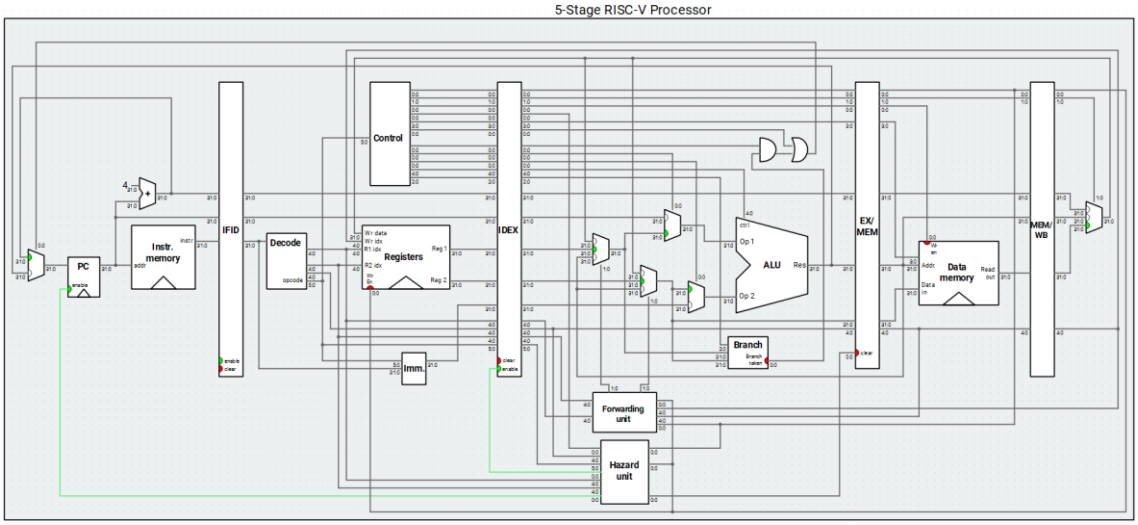

Figura 5 – Arquitetura do processador em pipeline com adiantamento de dados (forwarding) entre estágios do pipeline e com deteção de dependências de controlo.

2. 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).

3. Quais foram as alterações que introduziu? Porquê?

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

4. 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:
   - Número de instruções executadas:

5. 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**

6. 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**

7. Volte a executar o programa, carregando agora na tecla F6 ("*Clock the circuit with the selected frequency*").
8. Após terminar a execução, pressione o botão "*Show Stage Table*".
9. 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**

10. 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:
    - Número de stalls introduzidos em consequência de conflitos de controlo:

11. 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**

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**

3. 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:
   - Número de instruções executadas:

4. 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**

5. Volte a executar o programa, carregando agora na tecla F6 ("Clock the circuit with the selected frequency").
6. Após terminar a execução, pressione o botão "*Show Stage Table*".
7. 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**

8. 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:
   - Número de stalls introduzidos em consequência de conflitos de controlo:

9. 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**