Atividade Pipeline

Prof. Luciano L. Caimi
lcaimi@uffs.edu.br

Objetivo

- **Entender e avaliar o impacto do algoritmo e do compilador na performance de execução**

Ações

- Implementar e compilar o algoritmo em linguagem C
- Gerar o código assembly RISC-V
- Executar o código no simulador RIPES
- Contar stalls e flushes
- Avaliar a performance: CPI, IPC
- Otimizar algoritmo e repetir atividades

Softwares

Compilador online: C para ISA-RISC-V

https://godbolt.org/

Simulador RISC-V: RIPES

https://github.com/mortbopet/Ripes

Roteiro da Atividade 1

a) Implementar em linguagem C um programa que contém um vetor de 1000 elementos inteiros. O programa deve conter um laço que percorre o vetor preenchendo cada posição da seguinte forma: índices pares (0, 2, 4, 6, …) recebem o valor 2; indices impares (1, 3, 5, 7, …) recebem o valor 1.

b) Utilizar o compilador online https://godbolt.org/ para obter o código assembly (ISA RV-32I) do programa.

c) Ajustar o código assembly para execução no simulador RIPES.

d) Avaliar a quantidade de stalls e flushes do programa utilizando diferentes microarquiteturas:
   d1) pipeline sem detecção de hazard e sem adiantamento
   d2) pipeline com detecção de hazard e com adiantamento

e) Avaliar a performance

Roteiro da Atividade 1

a) **Implementar em linguagem C um programa que contém um vetor de 1000 elementos inteiros. O programa deve conter um laço que percorre o vetor preenchendo cada posição da seguinte forma: índices pares (0, 2, 4, 6, …) recebem o valor 2; índices impares (1, 3, 5, 7, …) recebem o valor 1.**

```
int i;
int vetor[1000];
for(i = 0; i < 1000; i++) {
    if((i % 2) == 0)
        vetor[i] = 2;
    else
        vetor[i] = 1;
}
```

Roteiro da Atividade 1

b) **Utilizar o compilador online https://godbolt.org/ para obter o código assembly (ISA RV-32I) do programa**

- **Utilizar compilador CLANG:**

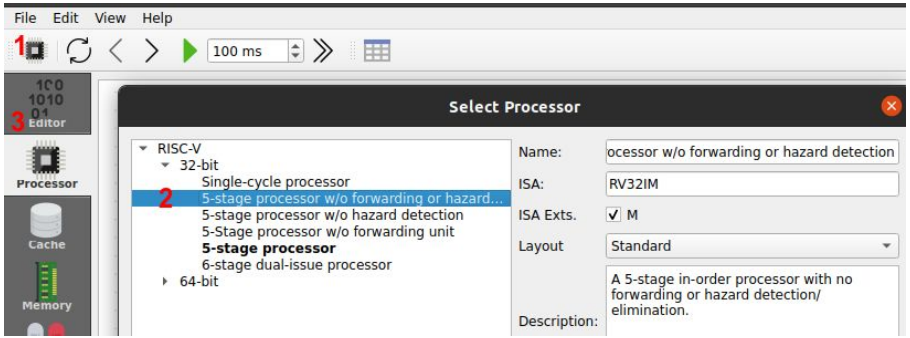
c) **Ajustar os rótulos do programa em um editor de texto retirando o "." pois o RIPES não aceita rótulos iniciando com "." (.label1 é inválido)**

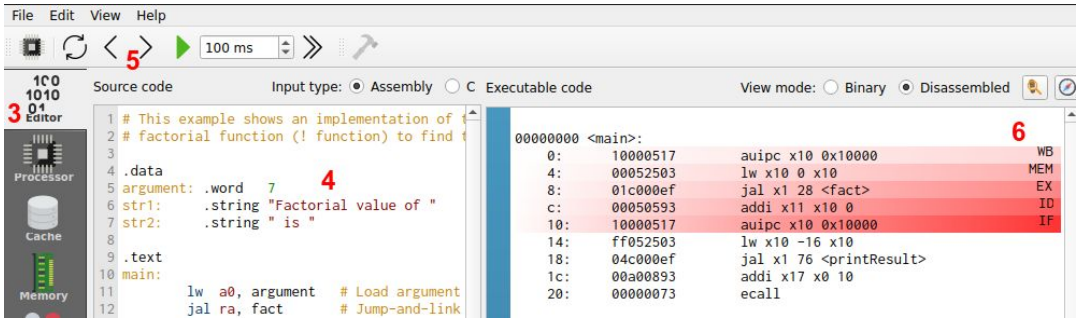
Roteiro da Atividade 1

d) Avaliar a quantidade de stalls e flushes do programa utilizando diferentes microarquiteturas:

d1 - pipeline sem detecção de hazard e sem adiantamento

1) configurar microaquitetura
2) seleção d1
3) para inserir código
4) área do código fonte
5) execução passo a passo
6) indicação de estágio do pipeline que a instrução se encontra

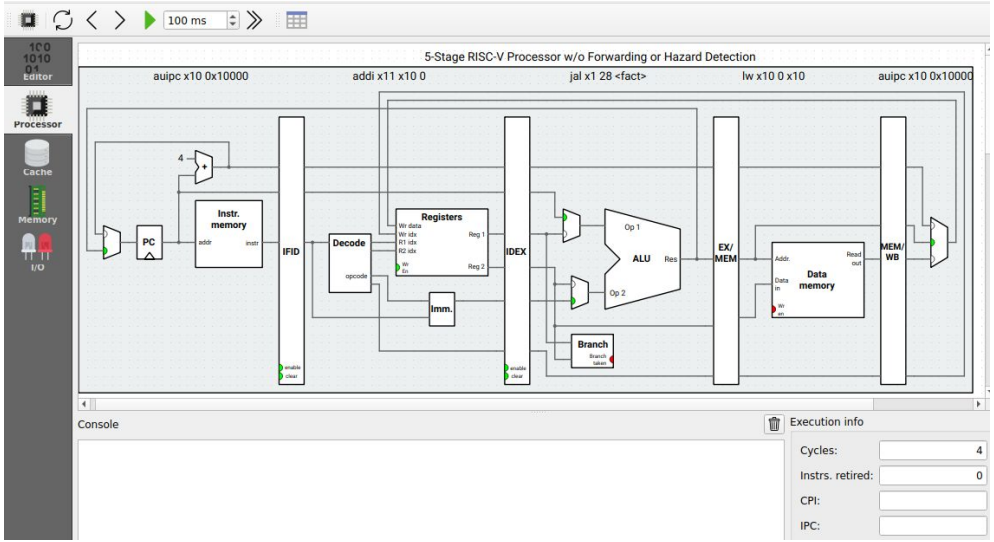


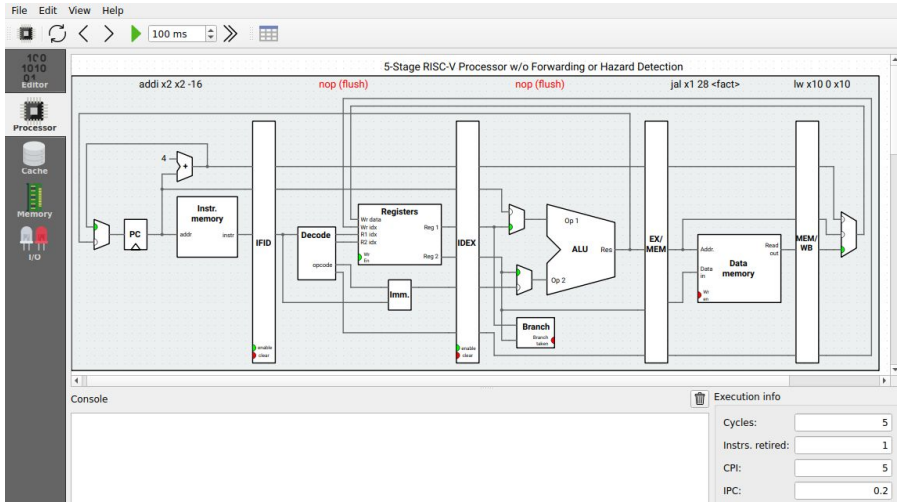


Roteiro da Atividade 1

d) Avaliar a quantidade de stalls e flushes do programa utilizando diferentes microarquiteturas:

d1 - pipeline sem detecção de hazard e sem adiantamento





Roteiro da Atividade 1

e) Avaliar a performance

e1) apresentar todos os stalls ocorridos na execução esclarecendo a razão pelo qual o mesmo aconteceu

e2) apresentar todos os flushes ocorridos na execução esclarecendo a razão pelo qual o mesmo aconteceu

e3) apresentar a quantidade total de stall e flushs presentes no código

e4) apresentar o total de instruções executadas e a quantidade de ciclos de clock

e5) apresentar a CPI média

Roteiro da Atividade 1

e) Avaliar a performance

e6) alterar a sequencia de instruções manualmente de forma a diminuir a quantidade de stalls e/ou flushes mantendo o execução correta do programa

e7) após ajuste indicar quais stalls e flushes foram suprimidos em relação aos apontados em e1 e e2

e8) apresentar a quantidade total de stall e flushs presentes após ajuste

e9) apresentar o total de instruções executadas e a quantidade de ciclos de clock após ajuste

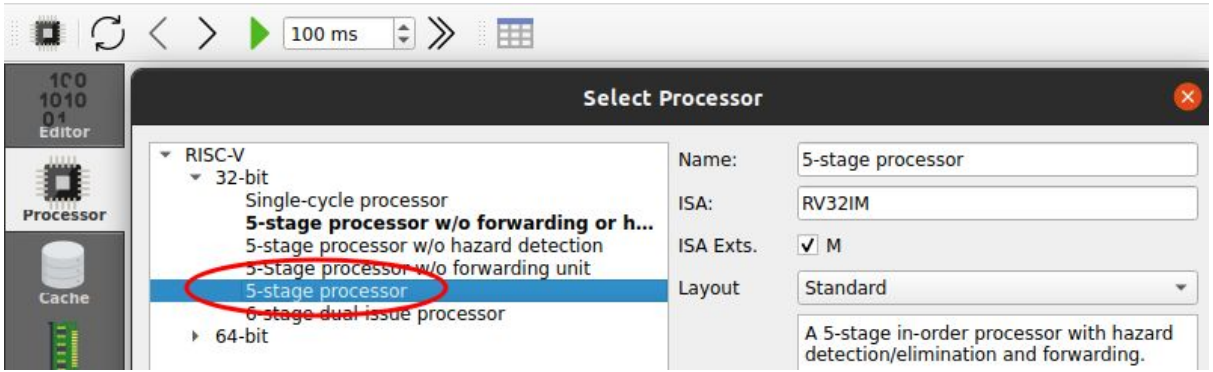
e10) apresentar a CPI média após ajuste

Roteiro da Atividade 2

igual a atividade 01 até item c

d) Avaliar a quantidade de stalls e flushes do programa utilizando diferentes microarquiteturas:

d2) pipeline com detecção de hazard e com adiantamento



e) Igual atividade 01 dos itens e1 até e5

Roteiro da Atividade 3

Mesmas ações da atividade 1 mas utilizando o pseudo código abaixo:

```
int i;
int vetor[1000];
for(i = 0; i < 1000; i+=2) {
        vetor[i] = 2;
        vetor[i+1] = 1;
}
```

Roteiro da Atividade 4

Mesmas ações da atividade 2 mas utilizando o pseudo código abaixo:

```
int i;
int vetor[1000];
for(i = 0; i < 1000; i+=2) {
        vetor[i] = 2;
        vetor[i+1] = 1;
}
```

Roteiro da Atividade 5

- **Realizar uma análise comparativa com os resultados obtidos em relação a performance nas 4 atividades realizadas e suas variações**

- **A análise deve enfatizar as diferenças e os porquês das diferentes performances**