

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Luiza Batista Laquini – 2019107786

7. CPU MIPS - Pipeline, Conflitos e Desempenho

Objetivos

Após completar as atividades deste laboratório, você irá:

- Conhecer como funciona o pipeline da CPU MIPS;
- Verificar o ganho de desempenho da CPU pipeline com relação à CPU monociclo;
- Verificar a influência dos conflitos no desempenho da CPU pipeline;
- Verificar com alguns conflitos podem ser evitados pela reorganização de instruções.

Introdução

Para executar esta atividade você poderá utilizar os simuladores MARS e DrMIPS. Considere o programa *fibonacci.asm* a seguir para executar esta atividade prática. O programa deveria calcular a famosa sequência de Fibonacci e armazenar os valores de cada passo do algoritmo na memória (Vout) e o último valor calculado em res.

```
.data
num: .word 10
res: .word 0
Vout: .space 400

.text
#carregando registradores com valores aleatorios
la $s1, Vout
lw $a0, num
li $t0, 1
li $t1, 1
li $t2, 1

loop:
ble $a0, $t0, end
sw $t2, 8($s1)
addi $s1, $s1, 4
addi $t0, $t0, 1
addi $t1, $t0, -2
move $t3, $t2
add $t2, $t2, $t1
move $t1, $t3
b loop

end:
sw $t2, 8($s1)
sw $t2, res
```

Tarefa 1

Avalie e, se necessário, corrija o programa Fibonacci.asm no DrMIPS com a CPU monociclo (unicycle.cpu) para valores de num = { 10, 20, 30, 40} e preencha a tabela a seguir com a variável res no final da execução e a quantidade de instruções executadas.

Dica: Use a ferramenta de geração de estatísticas de execução para auxiliar na resposta.

<i>num</i>	<i>res</i>	<i>#instrucoes</i>
10	55	99
20	6765	199
30	832040	299
40	102334155	399

Tarefa 2:

A seguir você deverá executar o seu código numa CPU pipeline de 5 estágios, conforme projeto apresentado no Capítulo 4 do livro texto (pipeline.cpu). Analise o seu código em busca de possíveis conflitos que causariam conflitos no pipeline. Liste quais são estes possíveis os conflitos, a sua causa (dependência dados, mudança de fluxo). Analise quais poderiam ser resolvidos por reorganização das instruções durante a compilação.

R: Podem ser resolvidas por reorganização das instruções durante a compilação as funções “b” e “ble”, pois ambas podem causar conflito por salto.

Tarefa 3:

Execute o seu programa fibonacci.asm (corrigido) na CPU pipeline (pipeline.cpu) do simulador DrMIPS e preencha a tabela abaixo, mostrando: a variável res no final da execução, a quantidade de instruções executadas e o número de bolhas geradas durante a execução do seu programa.

<i>num</i>	<i>res</i>	<i>#instrucoes</i>	<i>#bolhas</i>
10	55	99	34
20	6765	199	64
50	overflow	499	154
90	0 (erro)	769	232

Tarefa 4:

Calcule o ganho de desempenho da CPU pipeline em comparação com a CPU monociclo para as mesmas entradas. Justifique sua resposta!

R: Ganho de desempenho da CPU = (tempo do Monociclo / tempo do Pipeline)

num	monociclo (ps)	pipeline (ps)	ganho
10	92070	53200	1,730
20	185070	105200	1,759
30	278070	157200	1,768
50	464070	261200	1,777