

Módulo 5

Pipeline: Princípios Fundamentais

A duração de ciclo de relógio (T_{cc}) de um processador com uma organização encadeada é determinada pela latência da lógica combinatória do estágio mais demorado somada com a latência do registo que preserva os resultados de cada estágio.

$T_{estagio_i}$ – latência da lógica combinatória do estágio i

$T_{registo}$ – latência dos registos

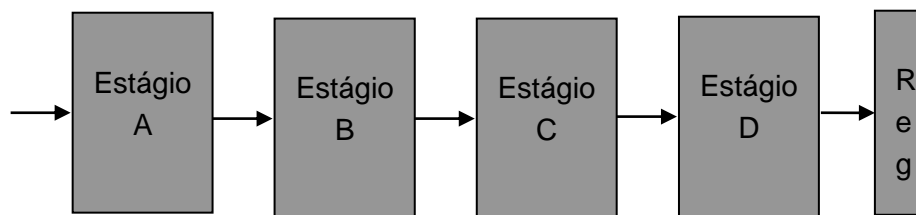
$$T_{cc} = \max(T_{estagio_i}) + T_{registo}$$

Assumindo que a instrução não é atrasada devido à ocorrência de anomalias, então:

- A **frequência do relógio** determina a taxa à qual o sistema pode mudar de estado. Se o $CPI=1$ então é igual ao débito de instruções, isto é, número de instruções executadas por unidade de tempo.
- A **latência de uma instrução** (i.e., tempo de execução de uma instrução) é o produto do número de estágios pela duração do ciclo de relógio (i.e., período do relógio)

Exercício 1

Considere que a lógica combinatória de um processador pode ser decomposta em 4 blocos de igual duração (60 ps) conforme ilustrado na figura.

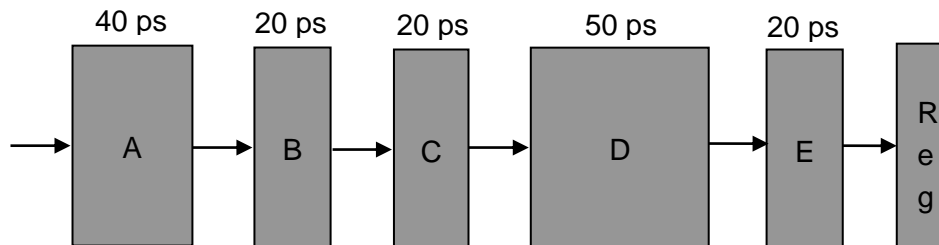


Sabendo que a latência dos registos é de 20 ps calcule o tempo de execução de uma instrução e a frequência máxima para uma organização de ciclo único (isto é, um *pipeline* degenerado num único estágio) e organizações com 2 e 4 estágios encadeados.

1 estágio (SEQ)		2 estágios		4 estágios	
Tcc		Tcc		Tcc	
Tinst		Tinst		Tinst	
f		f		f	

Exercício 2

Considere que a lógica combinatória de um processador pode ser decomposta em 5 blocos com a duração indicada na figura.



Sabendo que a latência dos registos é de 20 ps calcule:

- Para uma organização encadeada com 2 estágios como deve ser agrupados os blocos para maximizar a frequência? Qual a frequência máxima do relógio possível para esta organização e a latência de cada instrução?
- Qual a máxima frequência que pode ser obtida e a quantos estágios corresponde.

Exercício 3

Pretende-se analisar o desempenho de um programa com 1000 instruções a executar nas organizações propostas abaixo.

Considere que a lógica combinatória de uma organização sequencial tem uma latência de 500 ps. Um bloco de registos tem uma latência de 20 ps. Considere também que a lógica combinatória pode ser dividida em qualquer ponto, permitindo sub-blocos com latências arbitrárias (exigindo-se apenas que a soma das latências de todos os sub-blocos combinatórios seja de 500 ps).

A partir das condições descritas acima pretende-se desenhar várias versões encadeadas, criando sub-blocos de lógica combinatória e acrescentando os registos necessários. Cada novo estágio de *pipeline* criado a partir da versão sequencial incorre em 2 custos:

- tempo de registo e,
 - para este programa, 100 ciclos adicionais devido a dependências de dados e de controlo (causados por eventuais injeções de bolhas (*pipeline staling*)).
- Qual a latência de uma instrução na organização sequencial? Qual a frequência máxima e tempo de execução do programa?
 - Para organizações com 2, 4 e 10 estágios calcule o tempo de execução deste programa.
 - Não esquecendo nunca o custo associado ao *stalling* do *pipeline*, qual o número de estágios que minimiza o tempo de execução?
Sugestão: Preencha a tabela abaixo usando uma folha de cálculo.

#estágios	Tcc	#ciclos	Texec
1			
2			
3			
4			
5			
6			

7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			