

1. Material de apoio

A bibliografia relevante para este módulo é constituída pelas secções 4.4 e 4.5 do livro “Computer Systems: a Programmer’s Perspective”, de Randal E. Bryant e David O’Hallaron.

2. Latência e débito

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 dos registos que preservam os resultados de cada estágio.

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

$T_{registo}$ – latência dos registos

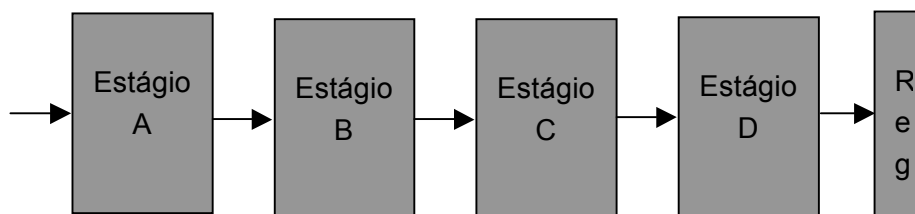
$$T_{cc} = \max(T_{estagio_i}) + T_{registo}$$

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

- **O débito** (número de instruções realizadas por unidade de tempo) é igual à frequência de relógio, uma vez que cada instrução demora um ciclo (i.e., $CPI = 1$)
- **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

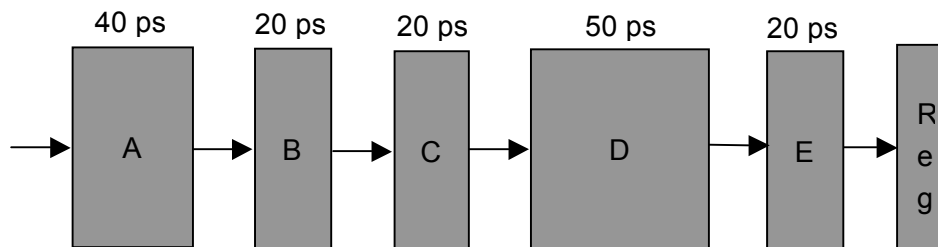
Considere que a lógica combinatória de um processador pode ser decomposta em 4 blocos de igual duração (60 ps) conforme ilustrado na figura.



Sabendo que a latência dos registos é de 20 ps calcule o tempo de execução de uma instrução e o débito para organizações sequencial (i.e., SEQ de ciclo único), 2 e 4 estágios encadeados.

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 o débito? Qual o débito e a latência de cada instrução?
- Qual o máximo débito que pode ser obtido 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 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 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 o débito e tempo de execução do programa?
- Para organizações com 2, 4 e 10 estágios calcule a latência de uma instrução. Tendo em consideração o custo associado ao *stalling* do *pipeline* calcule o débito e tempo de execução deste programa?