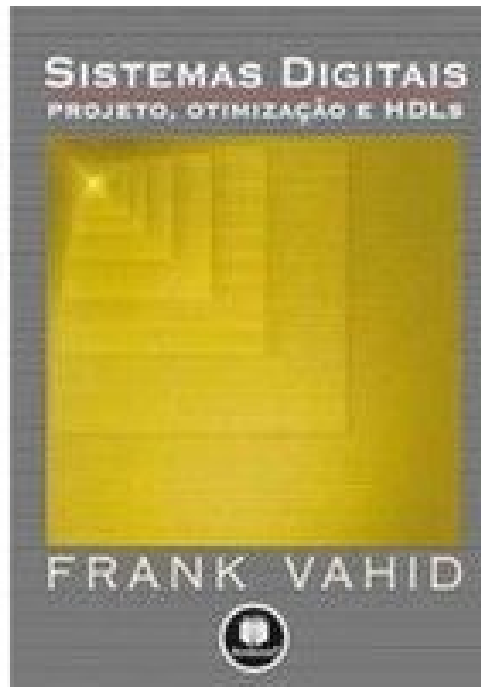


Teoria de Eletrônica Digital 2

Projeto RTL: Questões

Referência

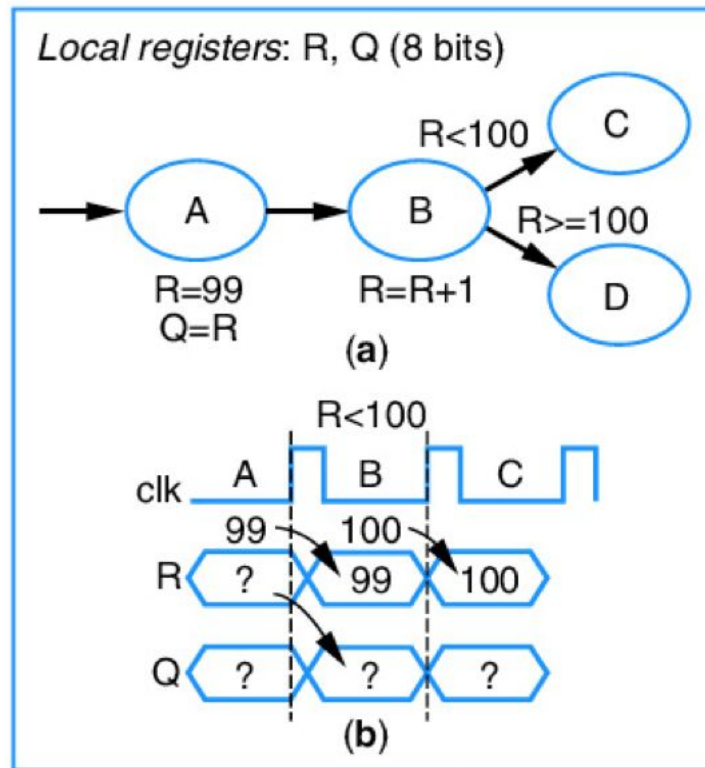
- Vahid – Itens 5.3 e 5.4



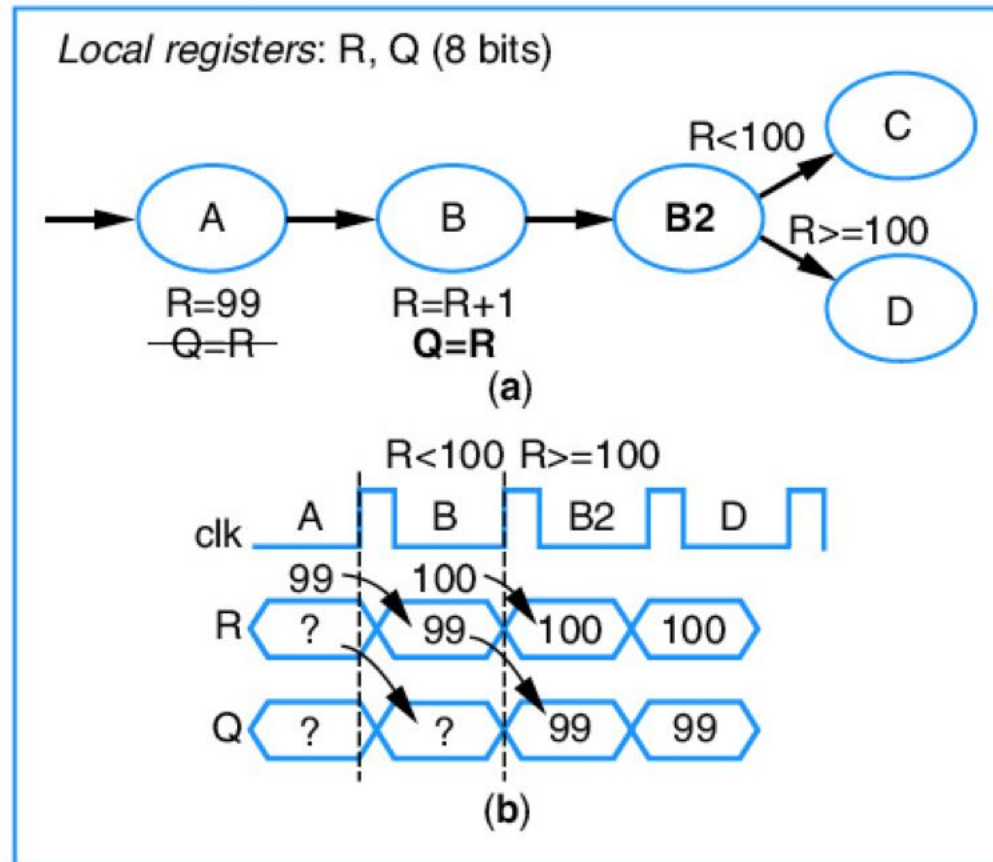
Objetivos

- Abordar questões importantes, como:
 - ❑ armadilhas comuns
 - ❑ determinação da frequência de relógio
 - ❑ otimização de projeto

Atualização de Registradores

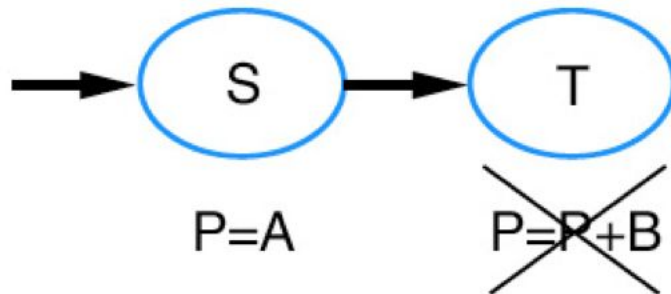


Atualização de Registradores



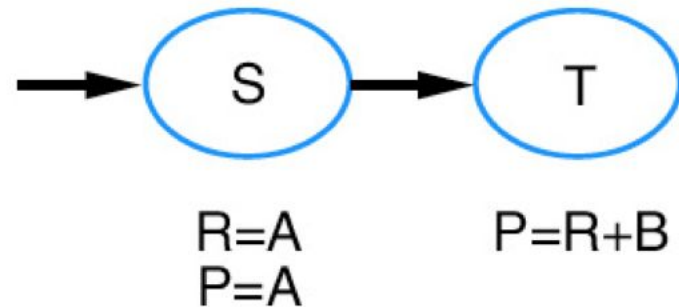
Leitura de Saídas

Inputs: A, B (8 bits)
Outputs: P (8 bits)



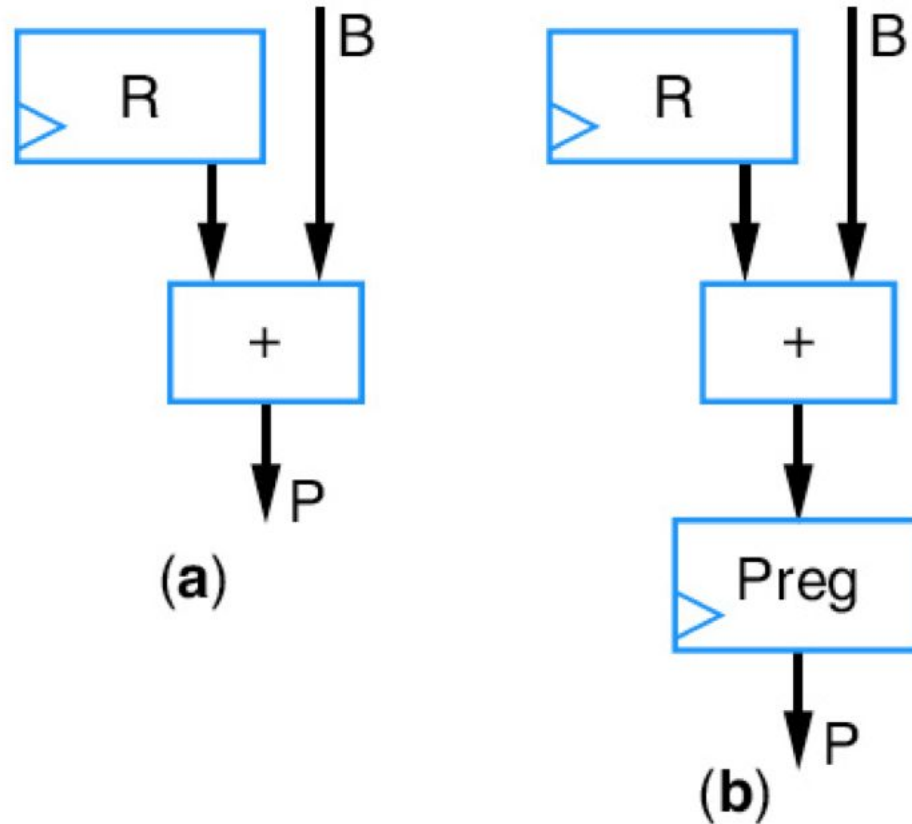
(a)

Inputs: A, B (8 bits)
Outputs: P (8 bits)
Local register: R (8 bits)



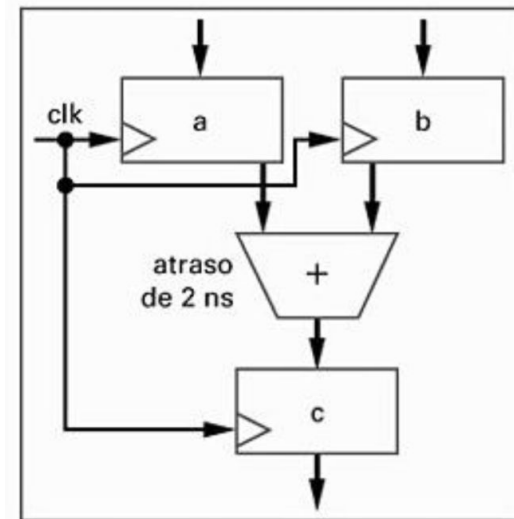
(b)

Saídas em Registradores



Determinando a Frequência

- Frequência determina o quão rápido o sistema irá executar a tarefa especificada
- Não pode ser arbitrariamente elevada.
- Ex: Saídas estáveis após 2 ns.
- $T = 10 \text{ ns}$
- $T = 5 \text{ ns}$
- $T = 1 \text{ ns}$: valor espúrio

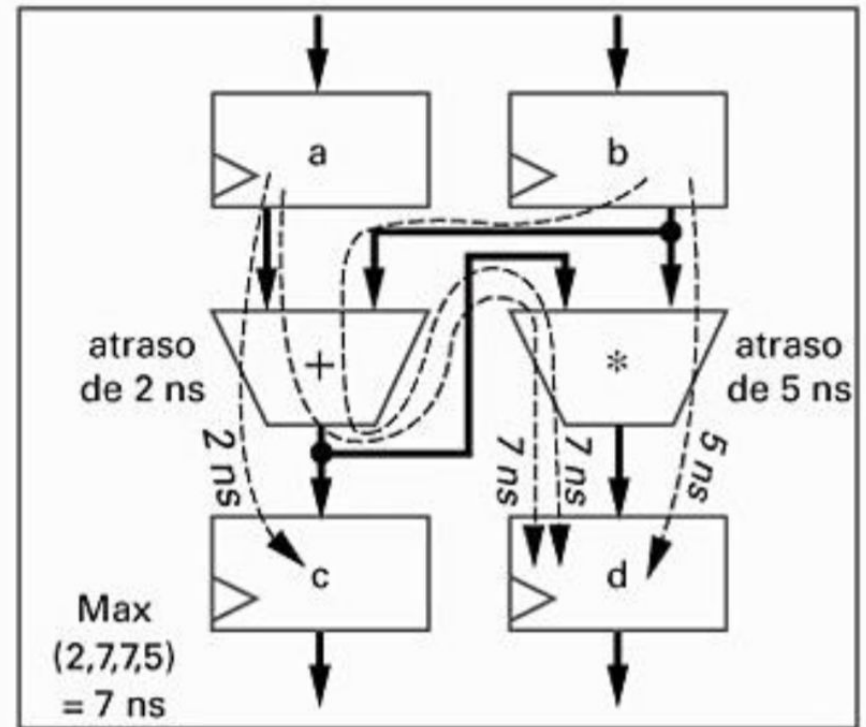


Determinando a Frequência

- Circuito deve ser completamente analisado para encontrar o **caminho crítico**
- **Caminho crítico**: atraso mais longo de um circuito
- Deve-se escolher uma **frequência cujo período é maior** do que o caminho crítico

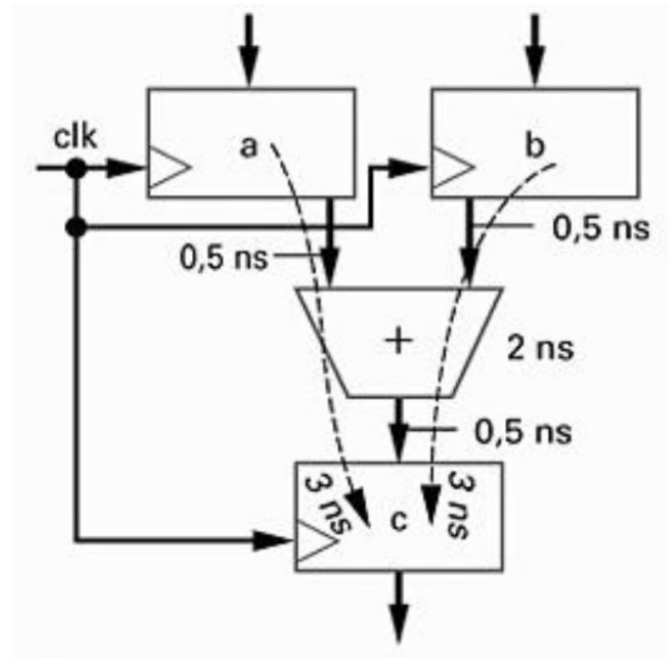
Determinando a Frequência

- 4 caminhos:
 - a, +, c: 2 ns
 - a, +, *, d: 7 ns
 - b, +, *, d: 7 ns
 - b, *, d: 5 ns
- Caminho crítico:
7 ns
- $T \geq 7\text{ns}$



Determinando a Frequência

- Fios também possuem atraso e são somados ao comprimento de um caminho



Determinando a Frequência

- Devem ser considerados também os tempos de *setup* dos registradores
- **Setup:** período de tempo em que as entradas dos registradores devem estar estáveis antes da borda do relógio
- O tempo de setup é somado ao comprimento do caminho
- Escolher um período ainda maior do que o caminho crítico

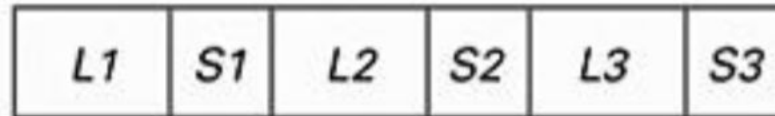
Otimização: Pipelining

- Maior velocidade
- Dividir uma tarefa grande em estágios sucessivos
- Cada estágio produz uma saída que será usada pelo estágio seguinte
- Todos os estágios trabalham ao mesmo tempo

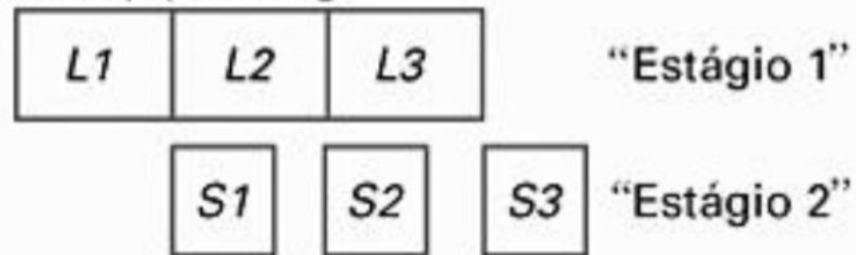
Otimização: Pipelining

- Ex: lavar pratos com um amigo

Sem pipelining:

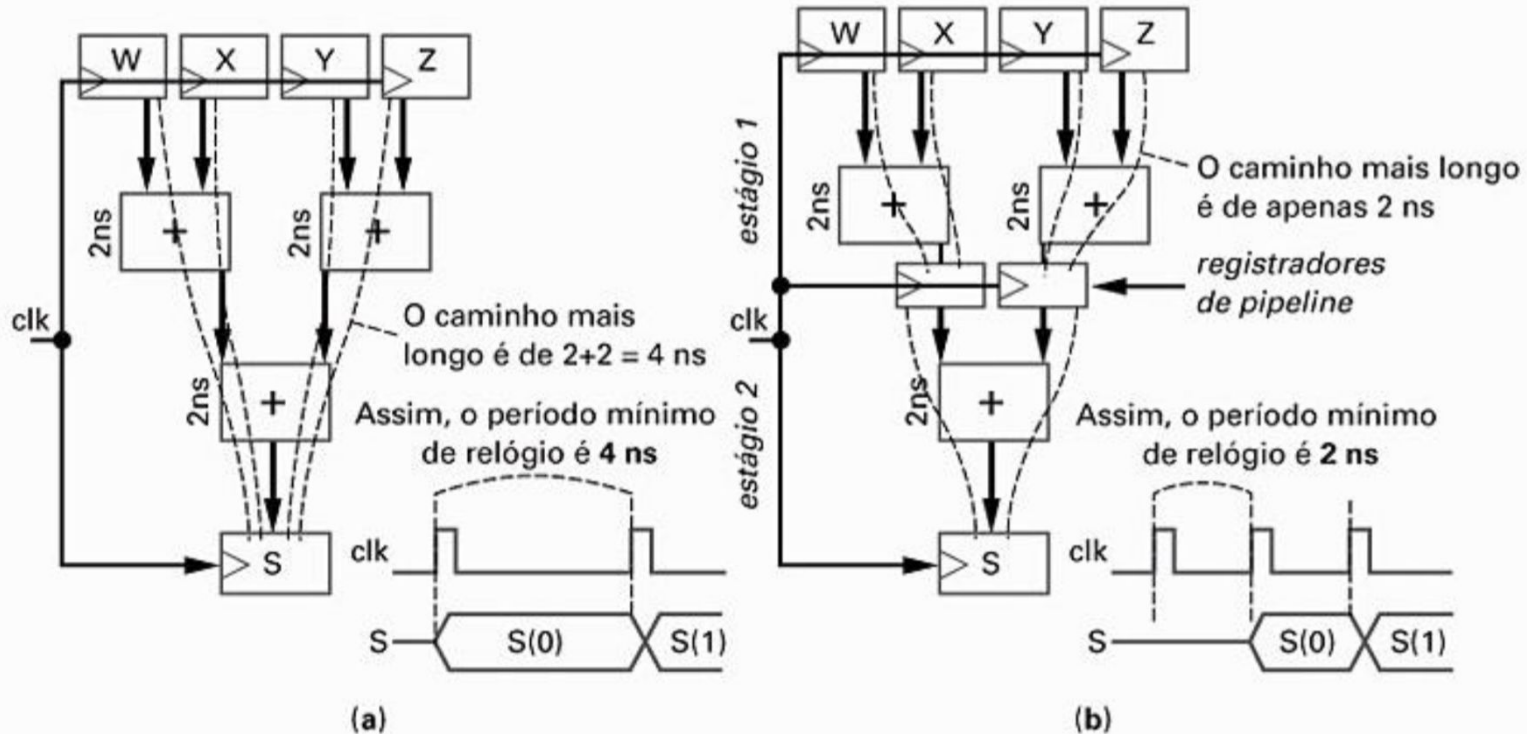


Com pipelining:



Otimização: Pipelining

- Ex: sistema com entrada de dados W, X, Y e Z, e saída $S = W + X + Y + Z$



Otimização: Pipelining

- 4 caminhos possíveis em a
- Todos os caminhos passam por 2 somadores (atraso somador = 2 ns)
- Caminho crítico: 4 ns
- Frequência máxima: $1 / 4\text{ns} = 250 \text{ MHz}$

Otimização: Pipelining

- Versão com pipeline em b, acrescentando-se registradores
- Registradores de pipeline
- Estágios: computações feitas entre os regs de pipeline
- Caminho crítico: 2 ns
- Frequência máxima: 500 MHz
- Desempenho dobrado

Otimização: Pipelining

- Latência: atraso necessário para que os novos dados de entrada transforme—se em novos dados de saída
- Latência em a: 1 ciclo (4 ns)
- Latência em b: 2 ciclos (4 ns)
- Throughput: taxa com a qual novas saídas aparecem no sistema
- Em a: 1 amostra a cada 4 ns
- Em b: 1 amostra a cada 2 ns (= dobro)