

Organização e Arquitetura de Processadores

Organização do MIPS

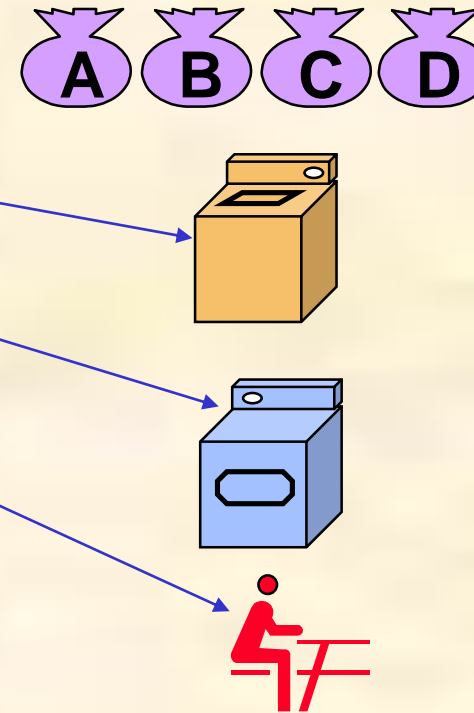
Introdução a Pipeline

**Cálculo do Relógio e Comparação entre
Organizações Monociclo, Multiciclo e Pipeline**

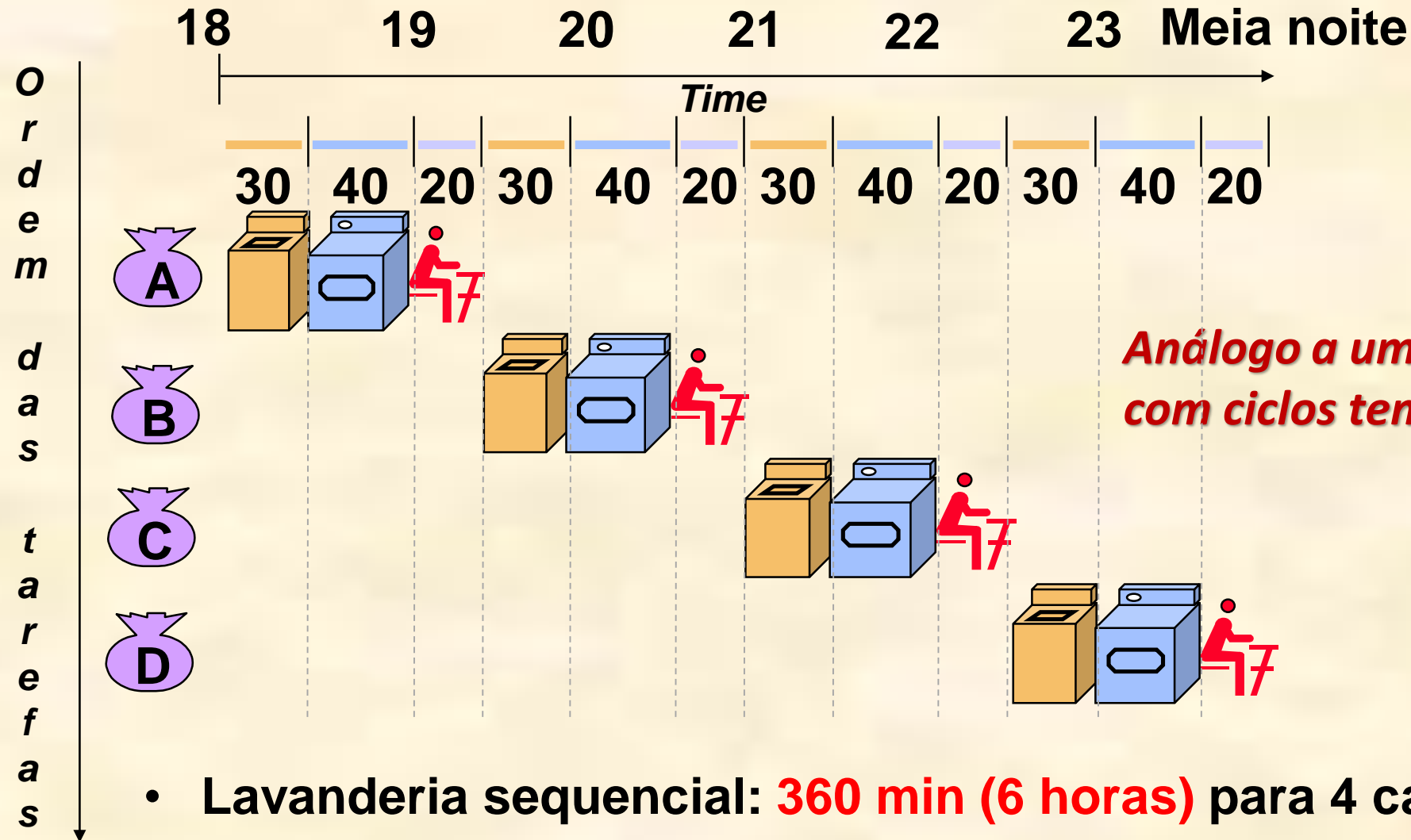
Pipeline é Natural!

- **Exemplo da Lavanderia:**

- Ana, Bruno, Cristiane e Daniela têm cada um uma trouxa de roupas para lavar, secar e dobrar;
- Lavagem leva 30 minutos
- Secagem leva 40 minutos
- Dobragem leva 20 minutos



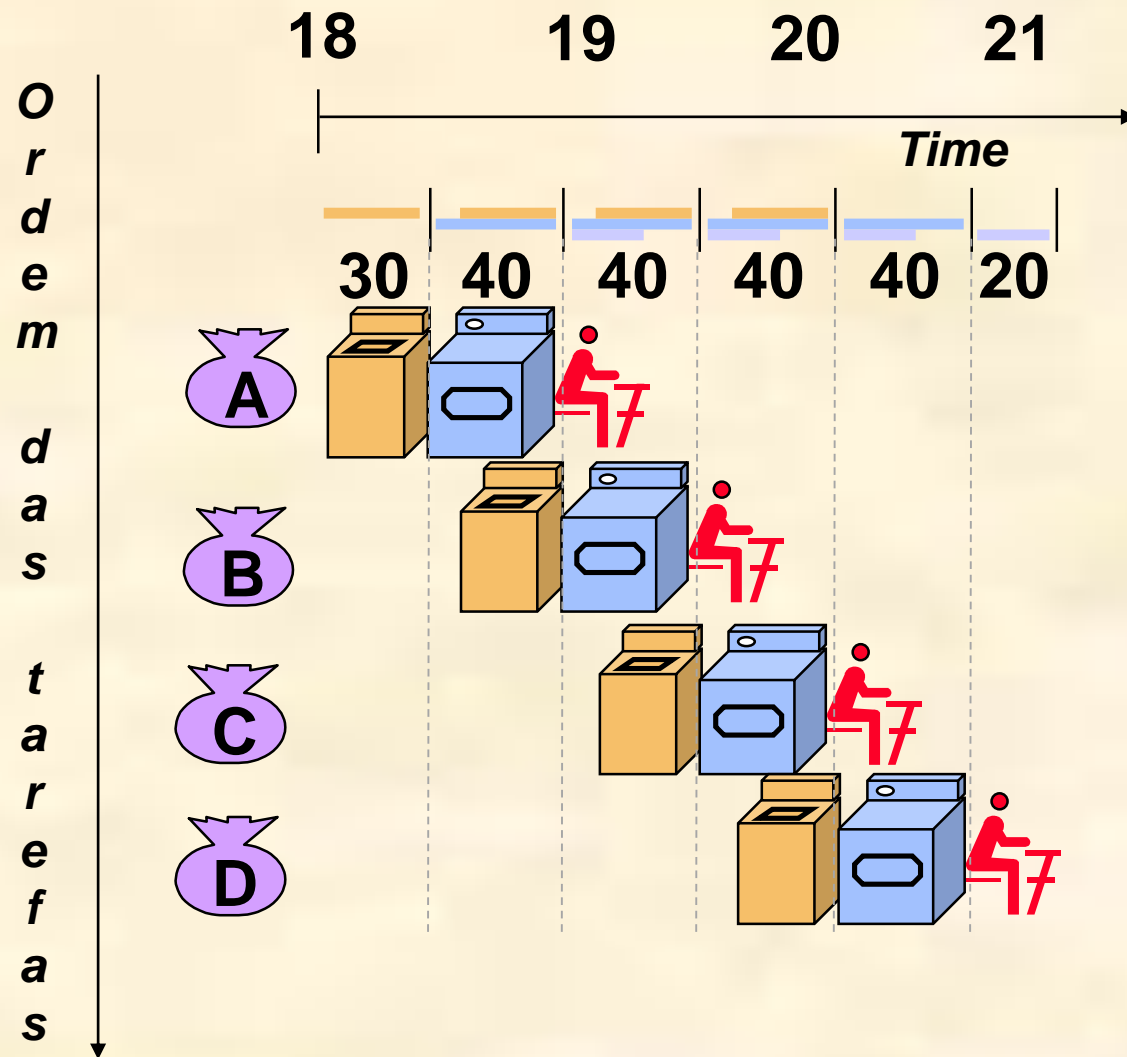
Lavanderia Sequencial



Análogo a um sistema Multiciclo, com ciclos tendo tempos distintos

- Lavanderia sequencial: **360 min (6 horas)** para 4 cargas
- Se aprendessem pipeline, quanto tempo levaria?

Lavanderia Pipeline



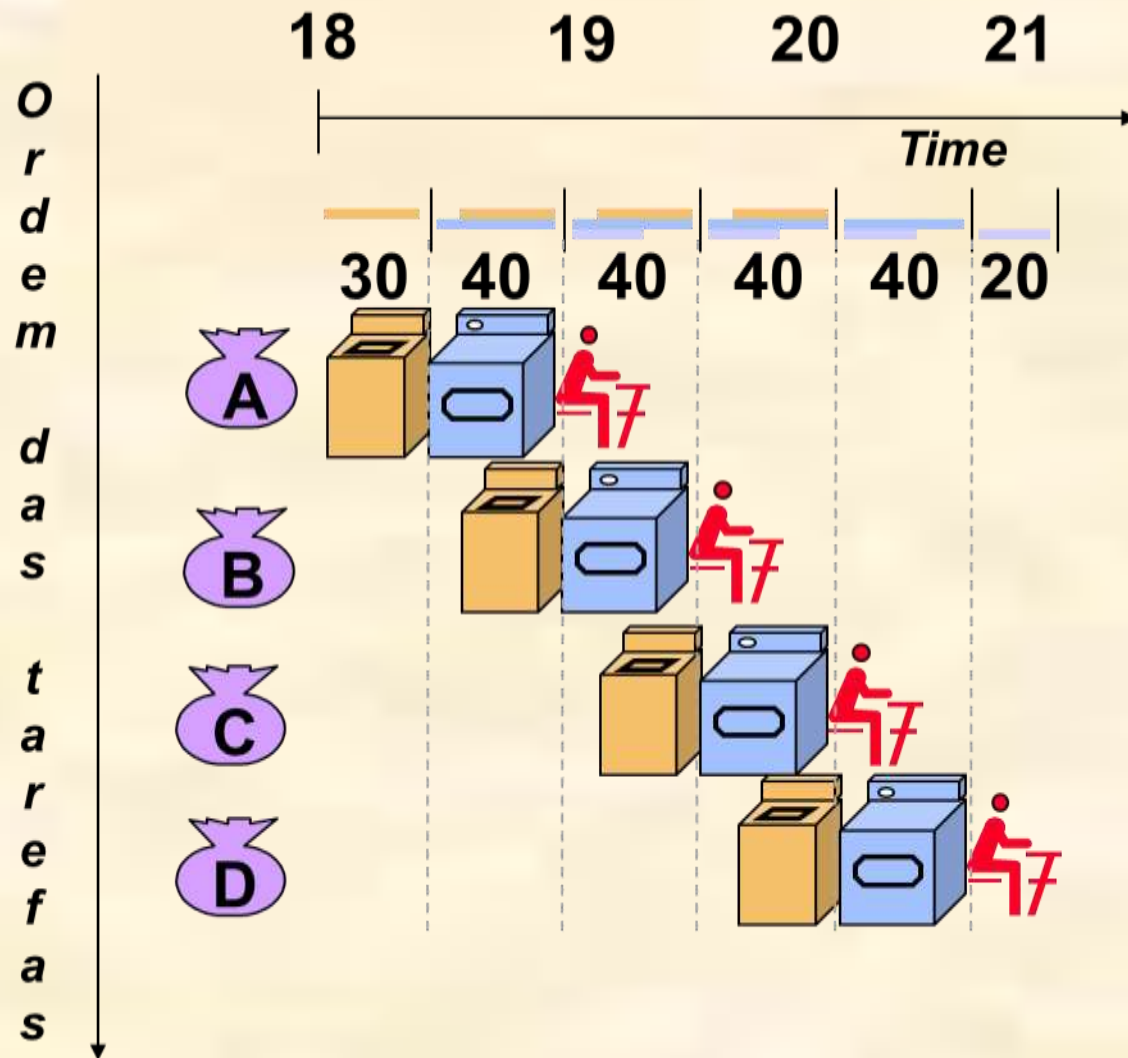
Ciclos são sobrepostos – Maior ciclo determina o atraso

- Lavanderia pipeline levaria **210 min (3.5 horas)** para as mesmas 4 cargas

Definições para Pipelines

- **Pipeline** = em inglês, tubo, oleoduto - instruções entram numa ponta e são processadas na ordem de entrada
- Tubo é dividido em **estágios** ou **segmentos**
- Tempo que uma instrução fica no tubo = **latência**
- Número de instruções executadas na unidade de tempo = **vazão** (em inglês, **throughput**)
- Tempo que uma instrução permanece em um estágio = **ciclo de máquina** - normalmente, igual a um ciclo de relógio (excepcionalmente dois)
- **Balanceamento** – é uma técnica para gerar uniformidade do atraso dos estágios

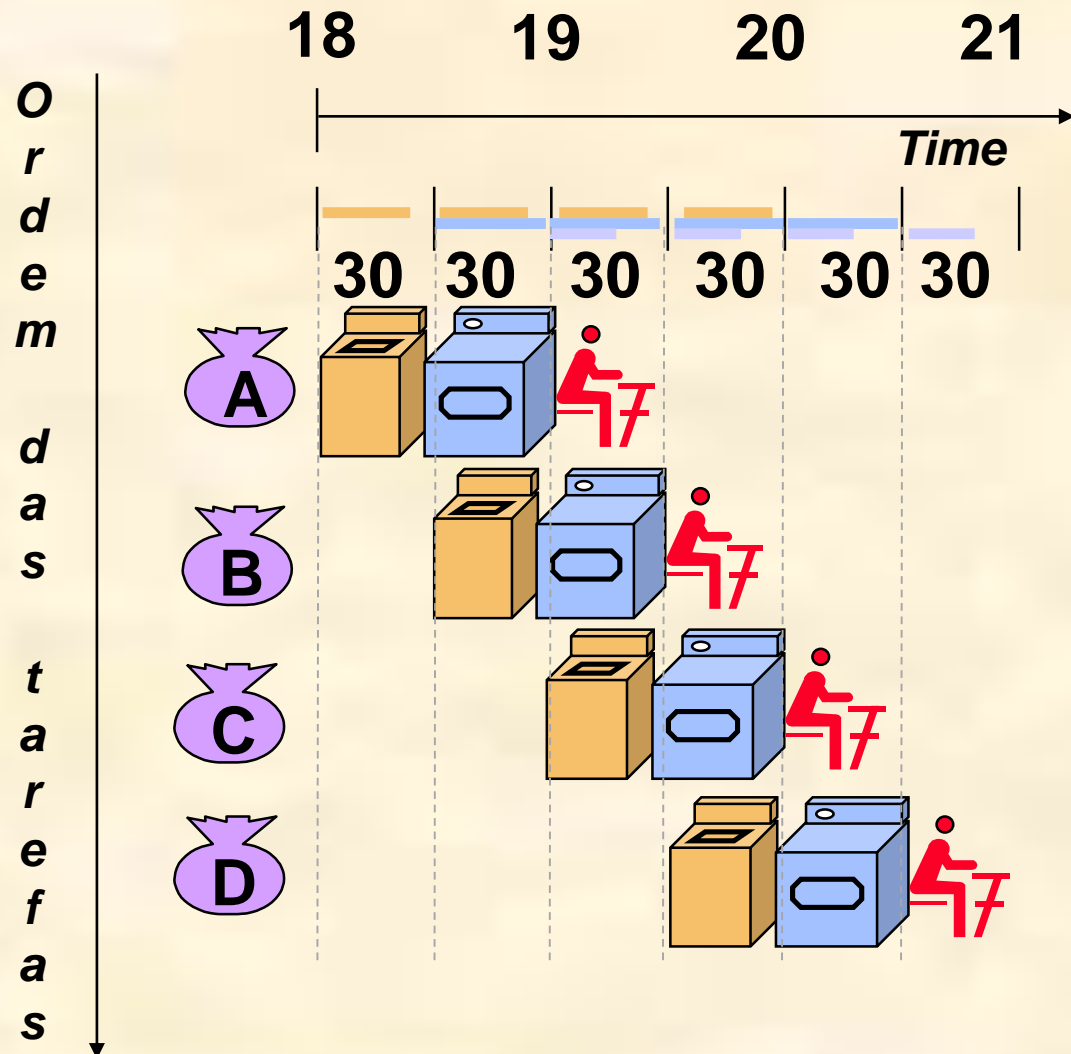
Lavanderia Pipeline



O que aconteceria se este pipeline fosse perfeitamente balanceado?

→ Considere a média dos tempos de cada tarefa para efetuar o balanceamento e calcule o novo valor total para fazer todas as tarefas

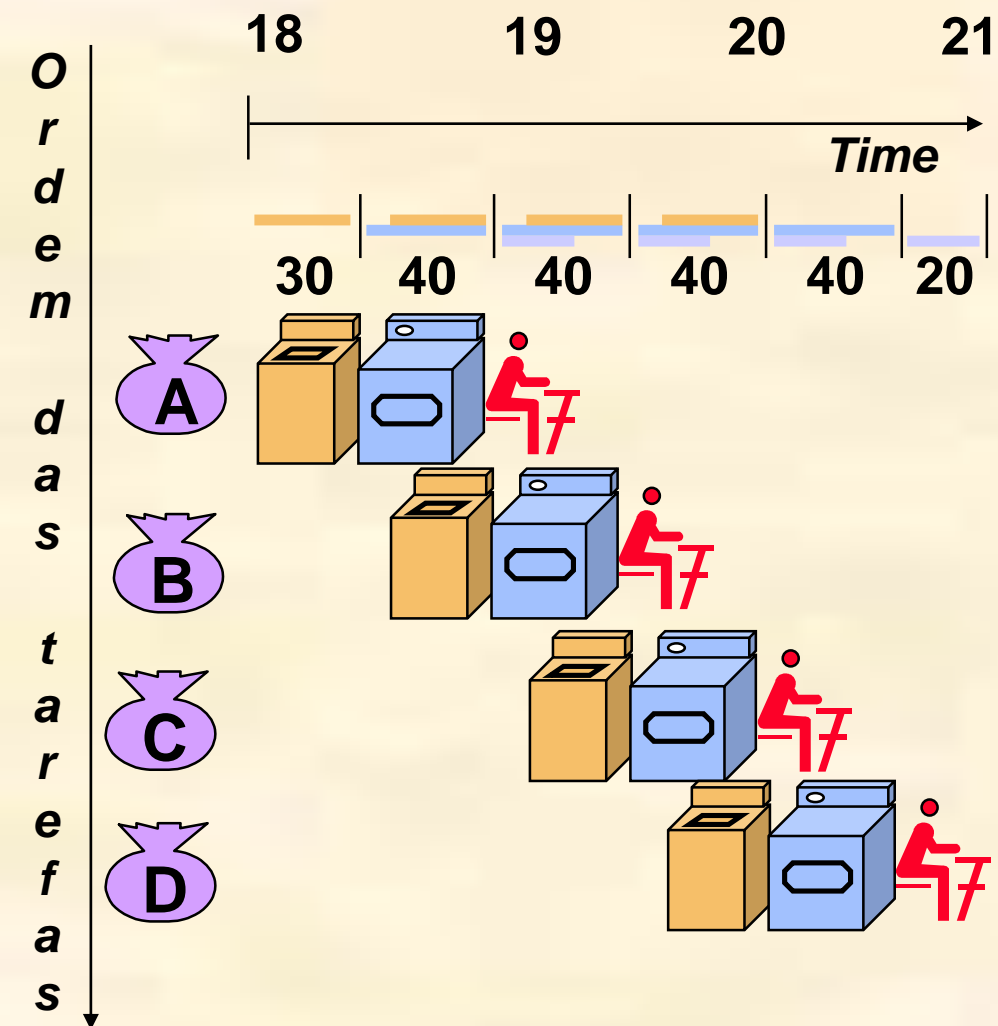
Lavanderia Pipeline



OBS.: O exemplo serve apenas para elucidar a questão de estágios perfeitamente balanceados; contudo, pode não ser prático, pois é necessário mudar as características da secadora ou do processo para operar em apenas 30 min. No caso de mudança do processo, poderia retirar a roupa um pouco úmida da secadora, prevendo que a mesma seja seca no estágio de dobragem

- Lavanderia pipeline levaria **180 min (3 horas)** para as mesmas 4 cargas

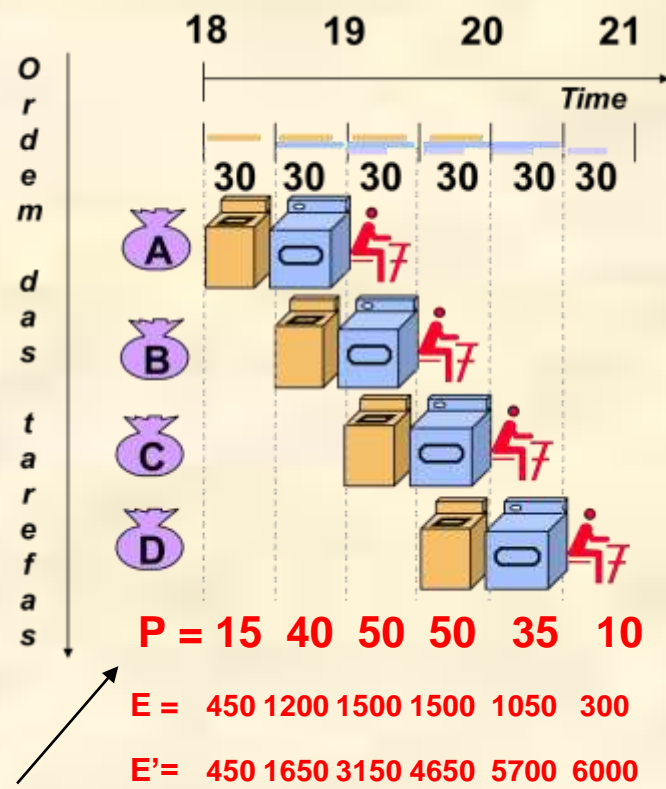
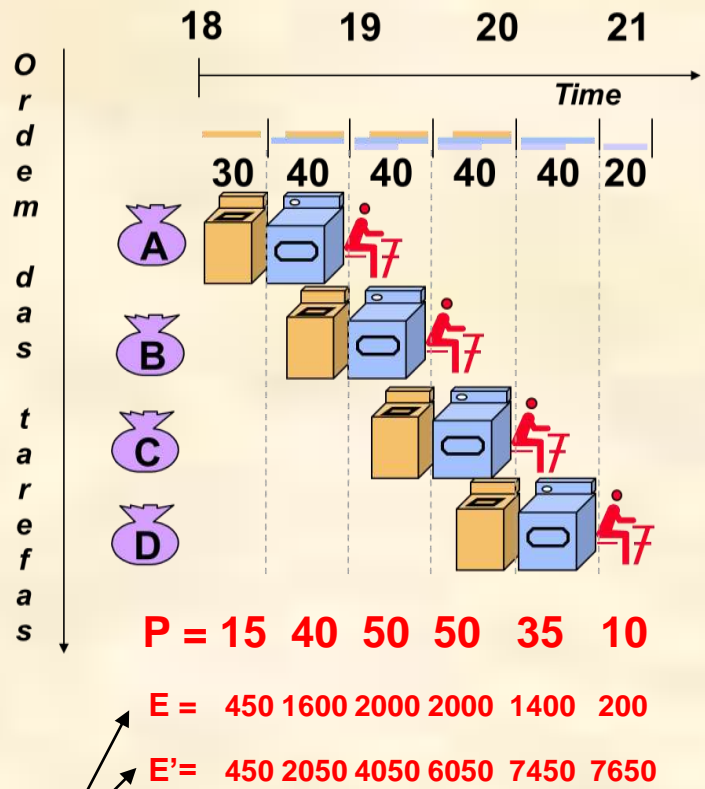
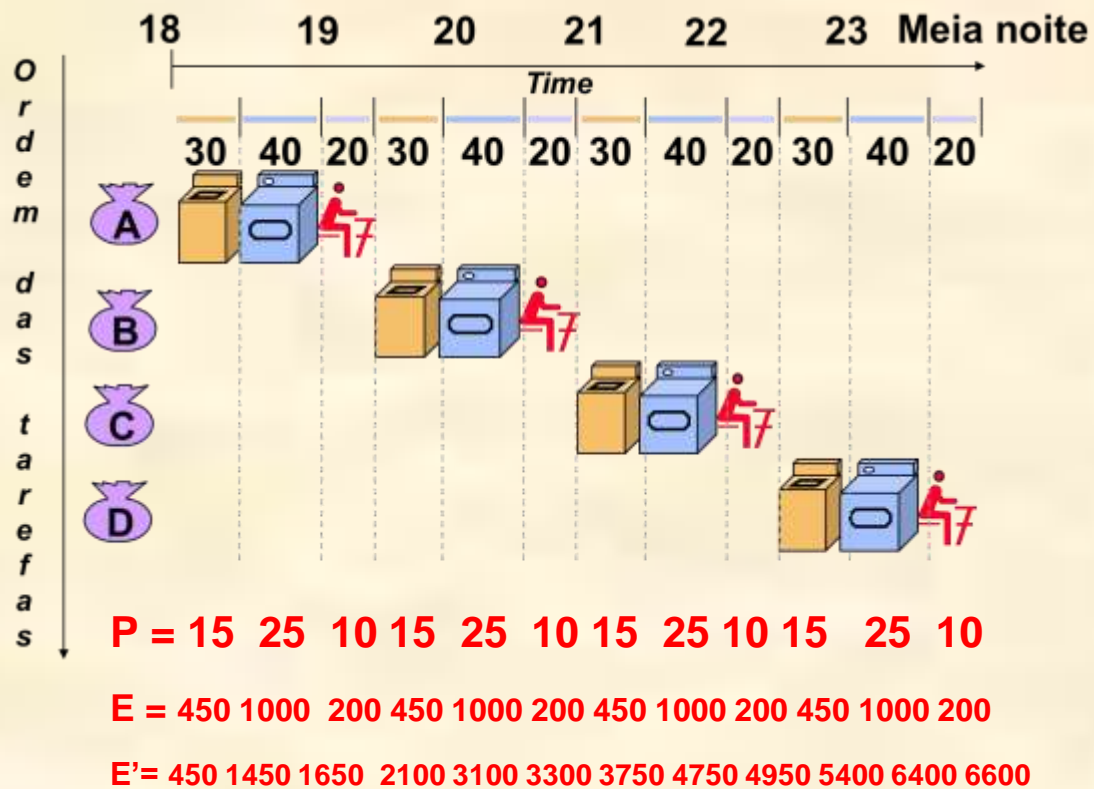
Lições Apreendidas com Pipeline



- Pipeline não reduz a **latência** de uma única tarefa, ajuda no **throughput** de todo o **trabalho**
- A taxa de pipeline é limitada pelo estágio **mais lento**
- Tarefas **múltiplas** operam de forma simultânea
- Aceleração potencial (speedup) = **Número de estágios do pipe**
- Comprimentos desbalanceados de estágios reduz speedup
- Tempo para “**preencher**” o pipeline e tempo para “**drená-lo**” reduzem speedup

Implementação Sequencial e Pipeline – Dissipação de Potência e Consumo de Energia

- Considere lavadora, secadora e dobradora consumindo 15W, 25W e 10W, respectivamente. Calcule o consumo de energia e dissipação de potência para cada implementação da lavanderia



OBS.: O consumo de energia pode reduzir se existir mecanismo que desligue as máquinas que já terminaram sua operação

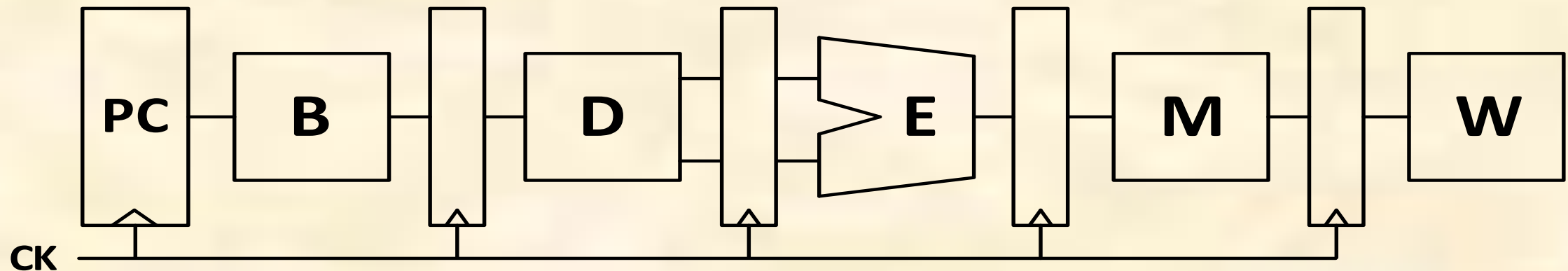
OBS.: Para que as máquinas tenham o mesmo tempo, é provável que a relação de dissipação de potência seja alterada. Logo, este exemplo é só ilustrativo

Visão Geral do Pipeline

- **Princípio básico**
 - Paralelismo em nível de instrução
 - Divide uma instrução em ***N*** estágios
 - ***N*** instruções executadas em **paralelo**, uma em cada estágio
- **Técnica empregada para aumentar a vazão de instruções**
- **Reduz caminhos críticos, acelerando o relógio**
- **Técnica em nível de hardware**
 - Solução “transparente” ao programador de alto nível
 - Mesmo programa de alto nível pode ser usado em uma organização monociclo, multiciclo ou pipeline

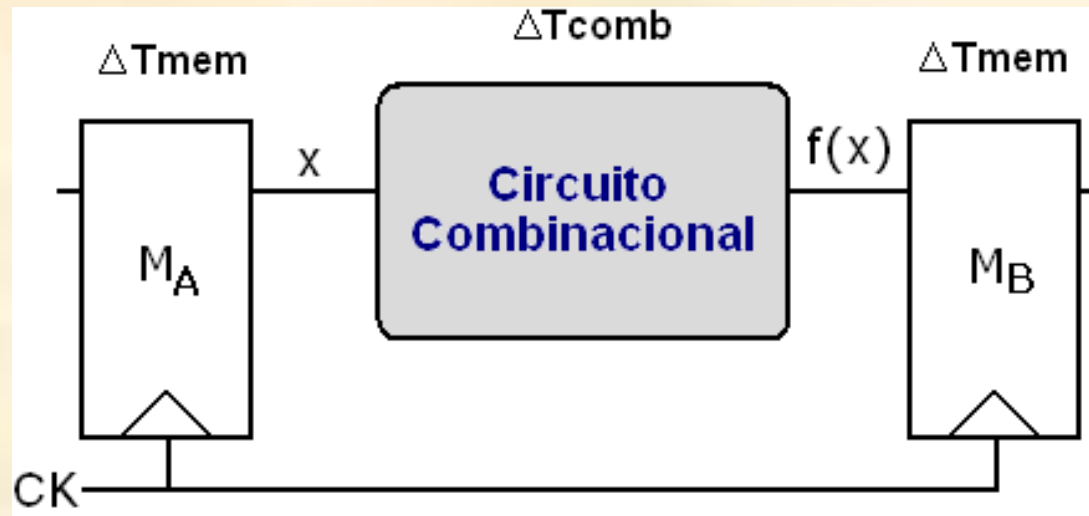
Visão Geral do Pipeline no MIPS

- **Instruções do MIPS básico são executadas em 5 etapas:**
 - 1. Busca da instrução na memória de código
 - 2. Leitura dos registradores em paralelo com a decodificação da instrução
 - 3. Execução de uma operação na ULA ou cálculo de um endereço de memória de dados
 - 4. Acesso à memória de dados para executar load ou store
 - 5. Escrita do resultado da ULA ou da leitura de memória em um registrador
- **No MIPS com Pipeline cada etapa é executada em um estágio do Pipeline**
 - Estágios são separados com uma barreira temporal (registrador / flip-flops)



Frequência de Operação de um Sistema

- A estimativa do período de um relógio é dada pela soma do atraso para a escrita na memória controlada pelo relógio (ΔT_{mem}) com o maior atraso combinacional (ΔT_{comb})

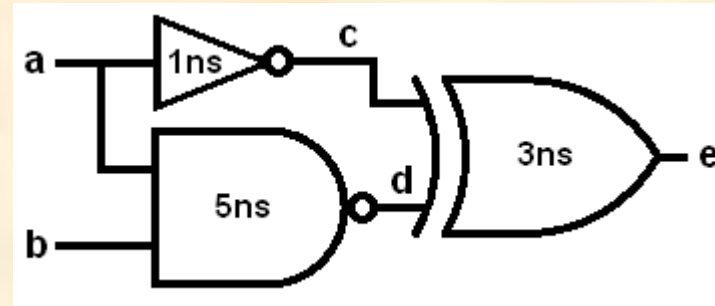


- $T = \Delta T_{\text{mem}} + \Delta T_{\text{comb}}$
- $f = 1 / T \quad \rightarrow \quad f = 1 / (\Delta T_{\text{mem}} + \Delta T_{\text{comb}})$

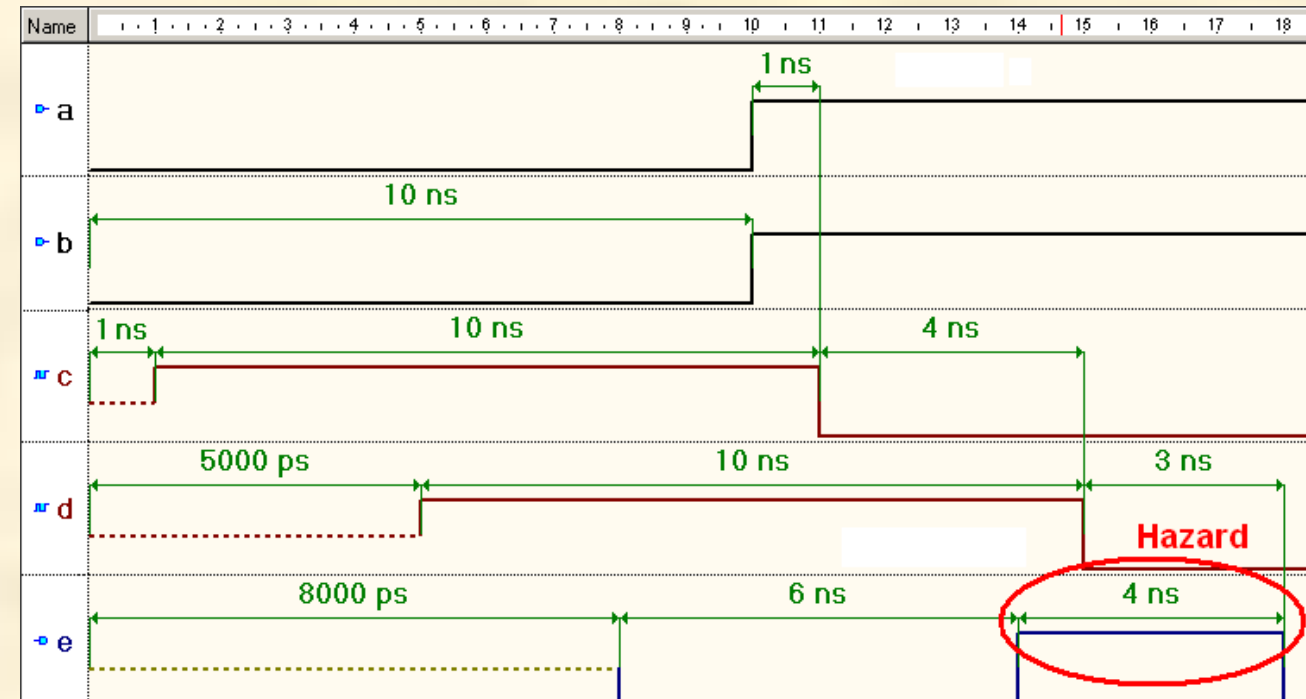
Análise de Hazards

- Não respeitar a frequência mínima determinada pelo caminho crítico pode implicar em registrar informação antiga ou eventuais hazards

```
entity Atrasos is
  port(a, b: in std_logic;
        e: out std_logic);
end Atrasos;
architecture Atrasos of Atrasos is
  signal c, d: std_logic;
begin
  c <= not a after 1ns;
  d <= not (a and b) after 5ns;
  e <= c xor d after 3ns;
end Atrasos;
```

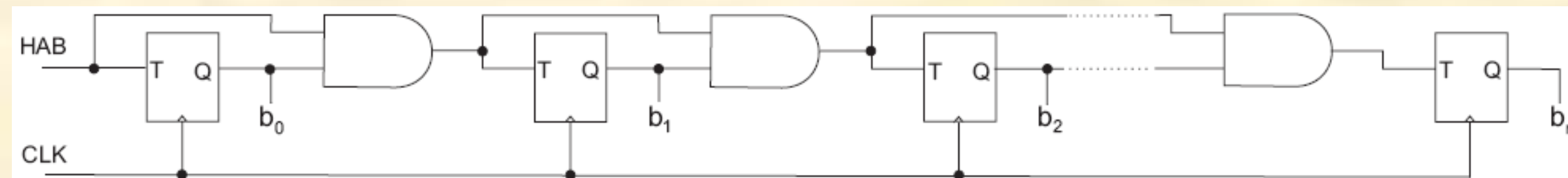


```
entity tb_Atrasos is
end tb_Atrasos;
architecture tb_Atrasos of tb_Atrasos is
  signal a, b, e: std_logic;
begin
  a <= '0', '1' after 10ns;
  b <= '0', '1' after 10ns;
  UUT: entity Atrasos port map (a => a, b => b, e => e);
end tb_Atrasos;
```



Exercício

1. (ENADE 2005) A figura apresenta um circuito contador serial síncrono de $n+1$ bits, construído com flip-flops, portas E e entrada habilitadora HAB
 - Considere que o único parâmetro temporal relevante é o tempo de propagação ou tempo de atraso e que os atrasos são de M segundos para cada flip-flop e de N segundos para cada porta E
 - Qual a expressão da maior frequência, em Hz, da onda quadrada a ser aplicada na entrada CLK para que o circuito funcione convenientemente?



$$\frac{1}{n(M+N)}$$

$$\frac{1}{M + nN}$$

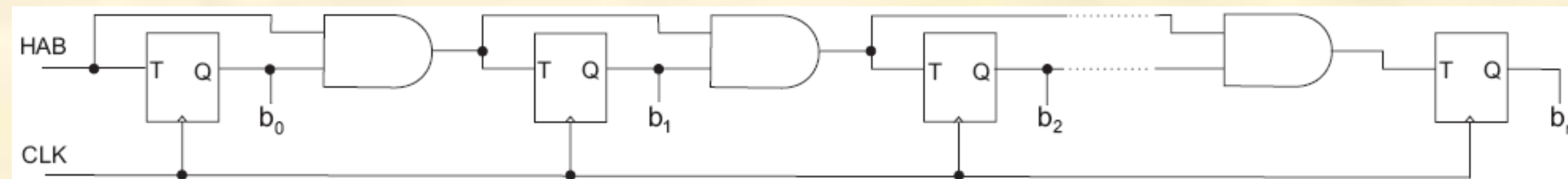
$$\frac{1}{(n+1)(M+N)}$$

$$\frac{1}{M + (n+1)N}$$

$$\frac{1}{(n+1)M + nN}$$

Resposta

1. (ENADE 2005) A figura apresenta um circuito contador serial síncrono de $n+1$ bits, construído com flip-flops, portas E e entrada habilitadora HAB
 - Considere que o único parâmetro temporal relevante é o tempo de propagação ou tempo de atraso e que os atrasos são de M segundos para cada flip-flop e de N segundos para cada porta E
 - Qual a expressão da maior frequência, em Hz, da onda quadrada a ser aplicada na entrada CLK para que o circuito funcione convenientemente?



$$\frac{1}{n(M+N)}$$

$$\frac{1}{M + nN}$$

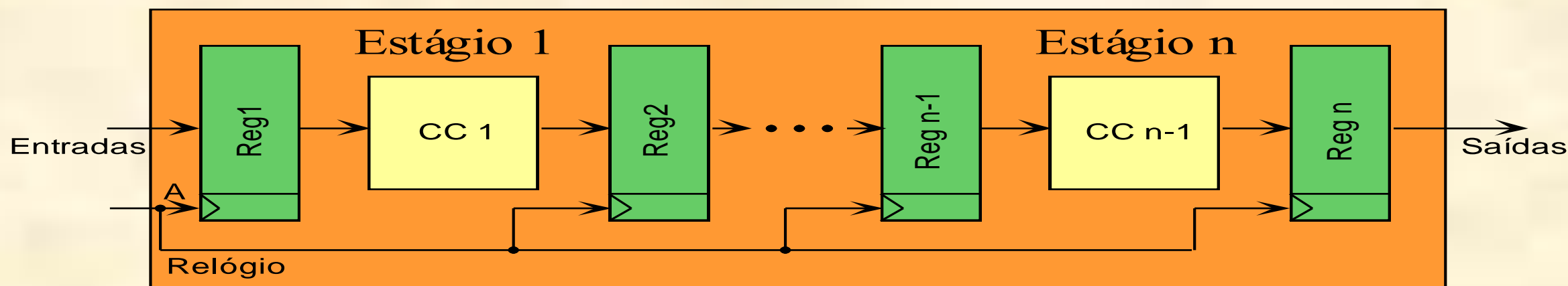
$$\frac{1}{(n+1)(M+N)}$$

$$\frac{1}{M + (n+1)N}$$

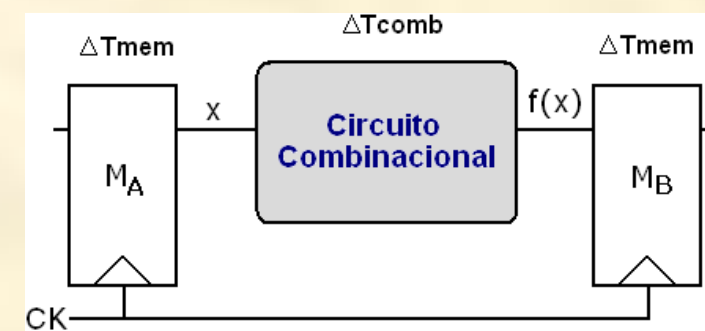
$$\frac{1}{(n+1)M + nN}$$

Organização Geral de um Pipeline e Frequência de Operação

- **Alternância de elementos de memória e blocos combinacionais:**
 - **memória:** armazena dados entre estágios, entre ciclos de relógio
 - **CC:** lógica combinacional que processa informação

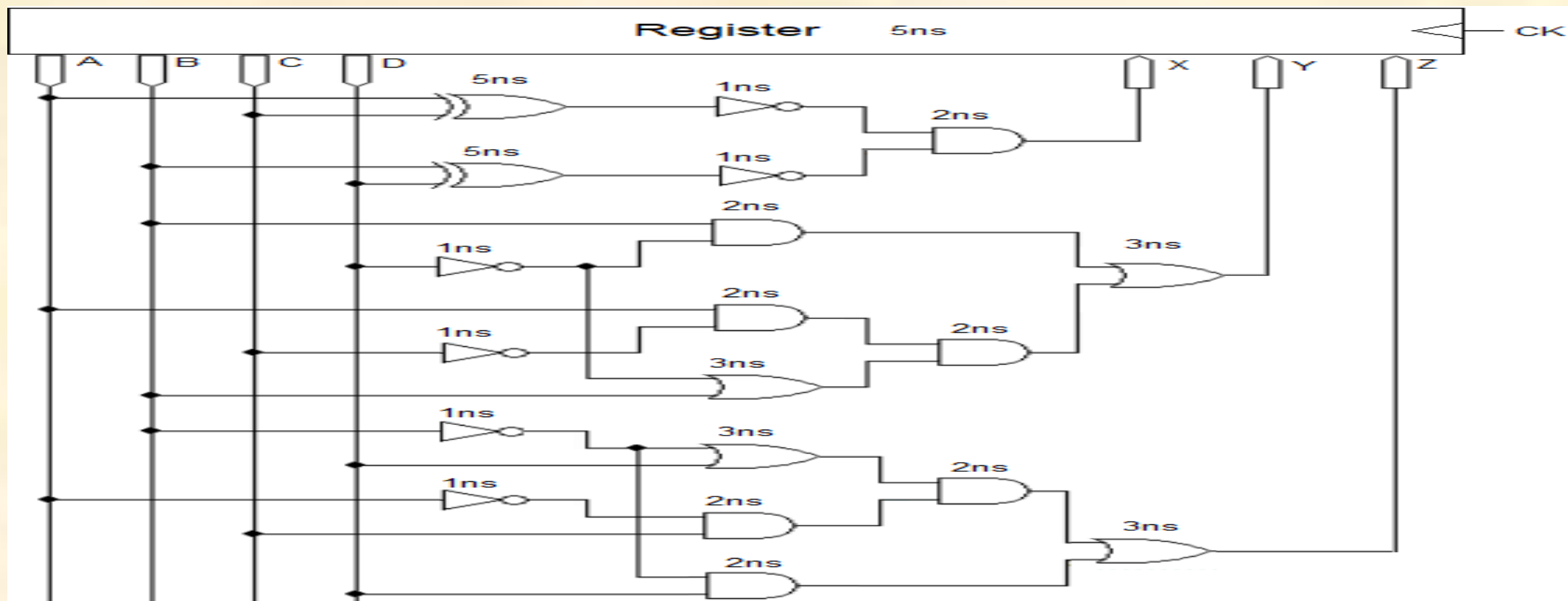


- **Pipeline divide caminho combinacional, reduzindo o caminho crítico → consequentemente reduzindo o período do relógio**



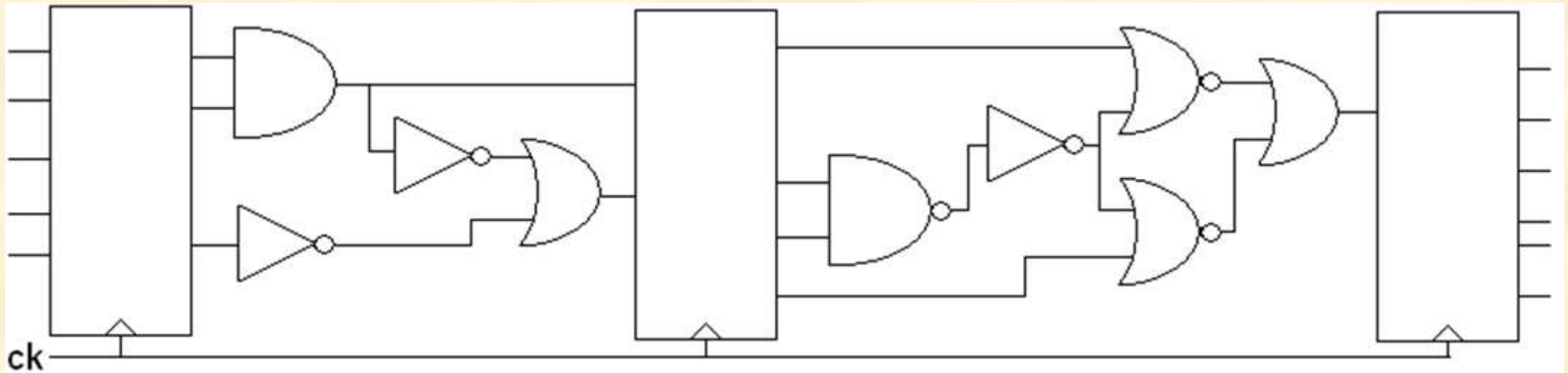
Exercício

1. Analise o circuito e calcule o período mínimo de relógio
2. Insira novas barreiras de forma a conseguir um relógio de 100 MHz



Exercício

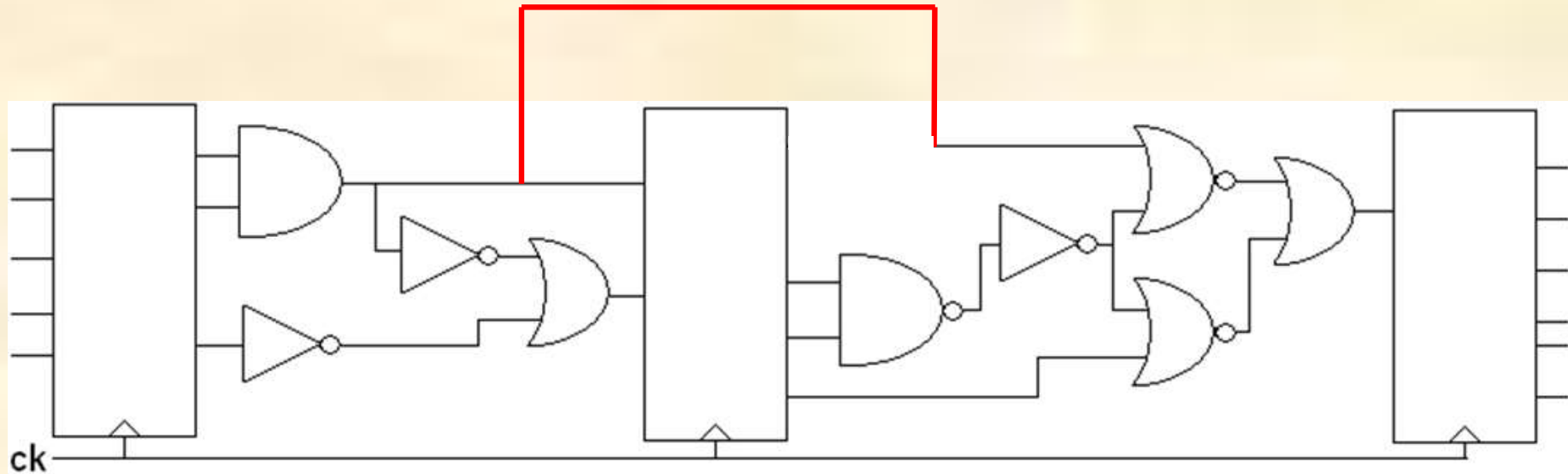
3. Determine a frequência máxima de operação do pipeline ilustrado abaixo. Considerando que portas NOT (inversores) atrasam 1ns, portas NAND (não e lógico) e NOR (não ou lógico) atrasam 3ns, portas AND (e lógico) e portas OR (ou lógico) atrasam 5ns, e as MEMÓRIAS têm atraso de 10ns



4. Explique o que acontecerá se o relógio operar em frequência maior que a calculada

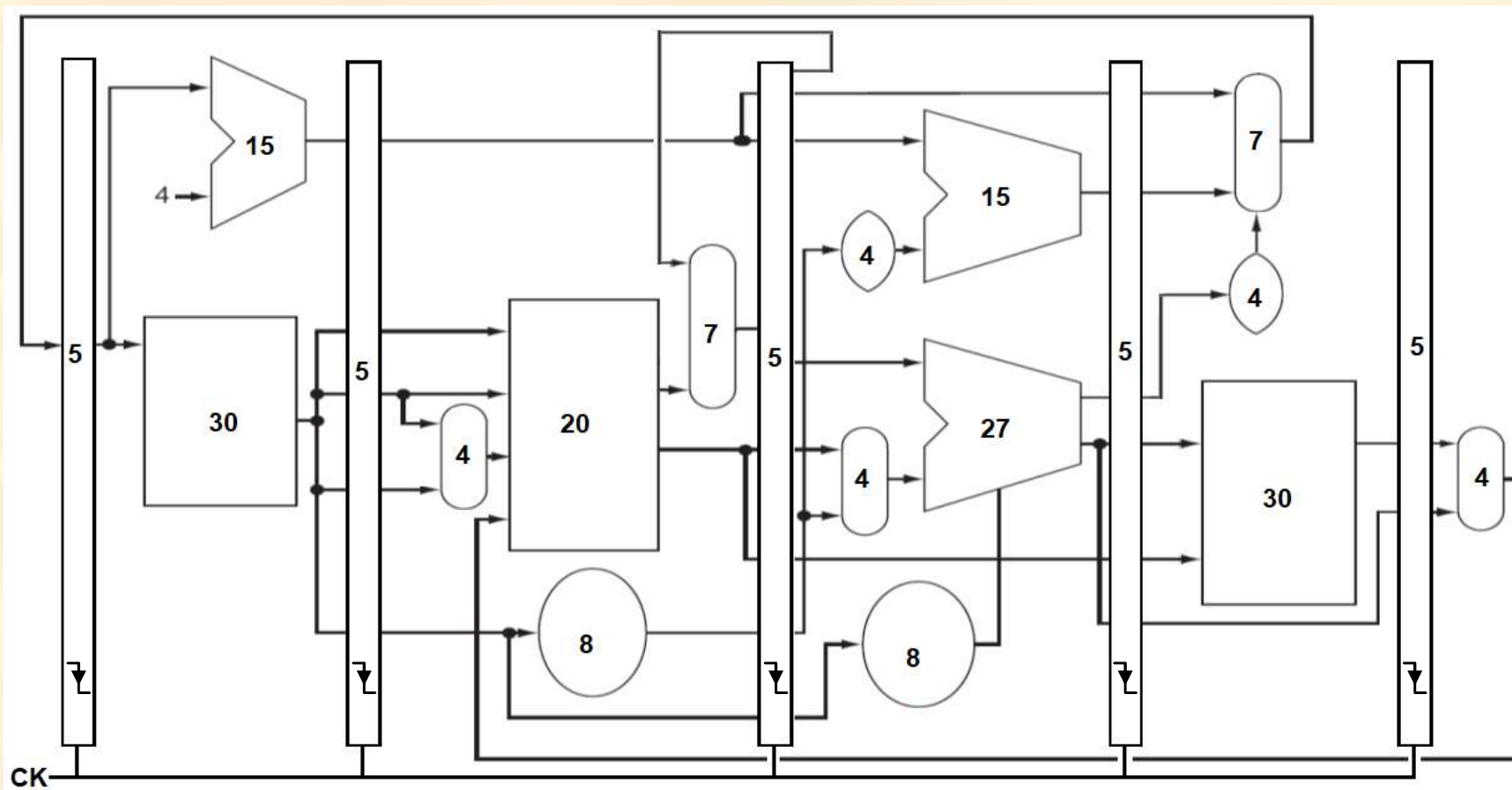
Exercício

5. Faça a mesma análise do circuito anterior, considerando agora o novo caminho de sinal inserido abaixo



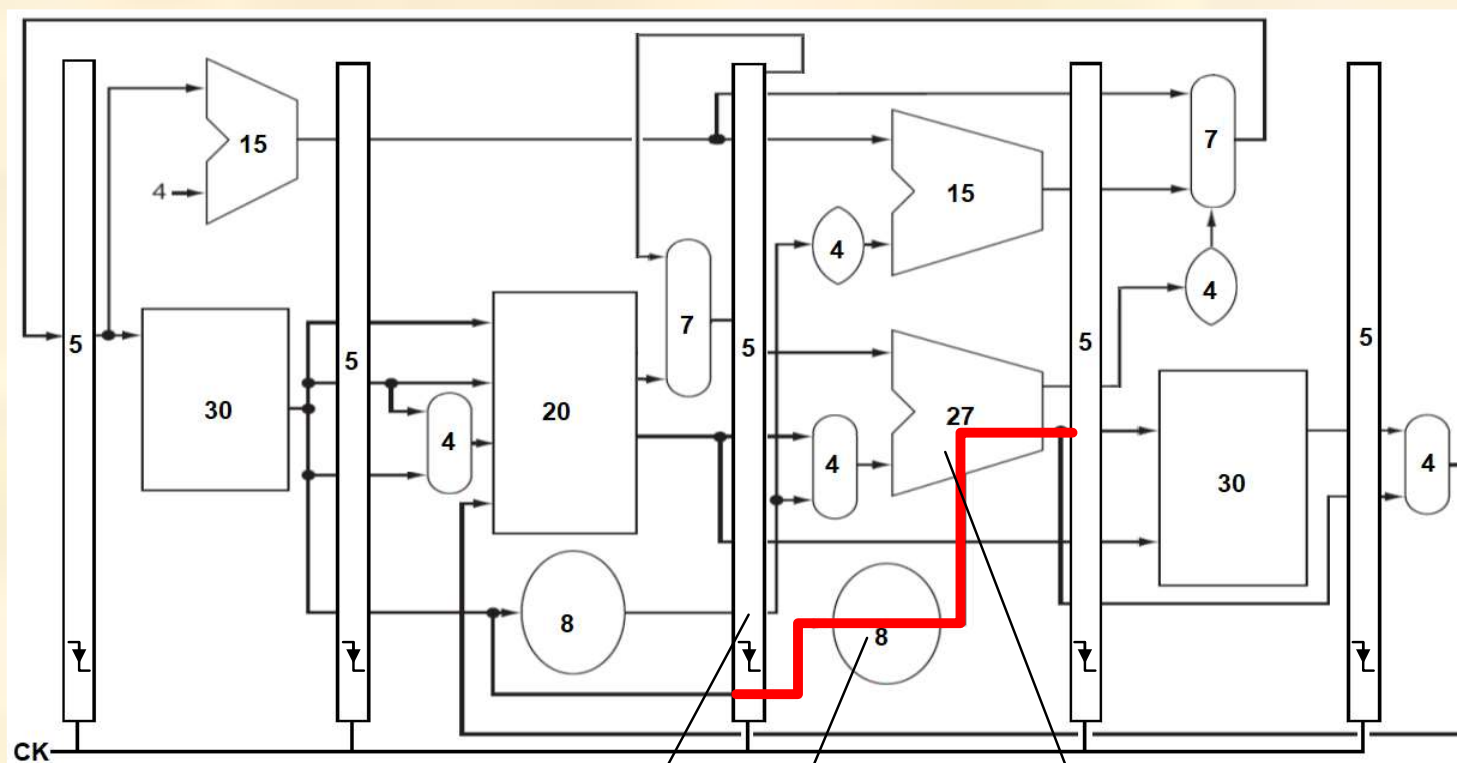
Exercício

6. Qual a frequência máxima de operação (sinal CK) em MHz do circuito abaixo; os números internos aos blocos e barreiras temporais estão em nano segundos (ns). Marque na figura qual é o caminho crítico



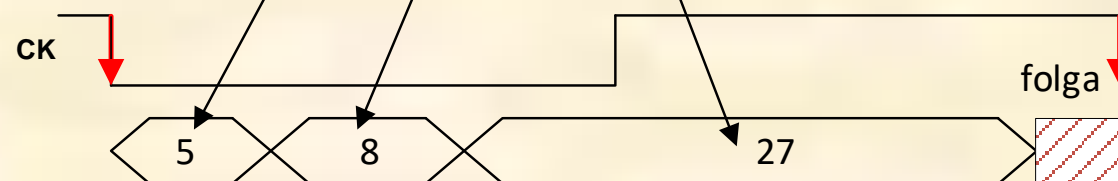
Resposta

6. Qual a frequência máxima de operação (sinal CK) em MHz do circuito abaixo; os números internos aos blocos e barreiras temporais estão em nano segundos (ns). Marque na figura qual é o caminho crítico



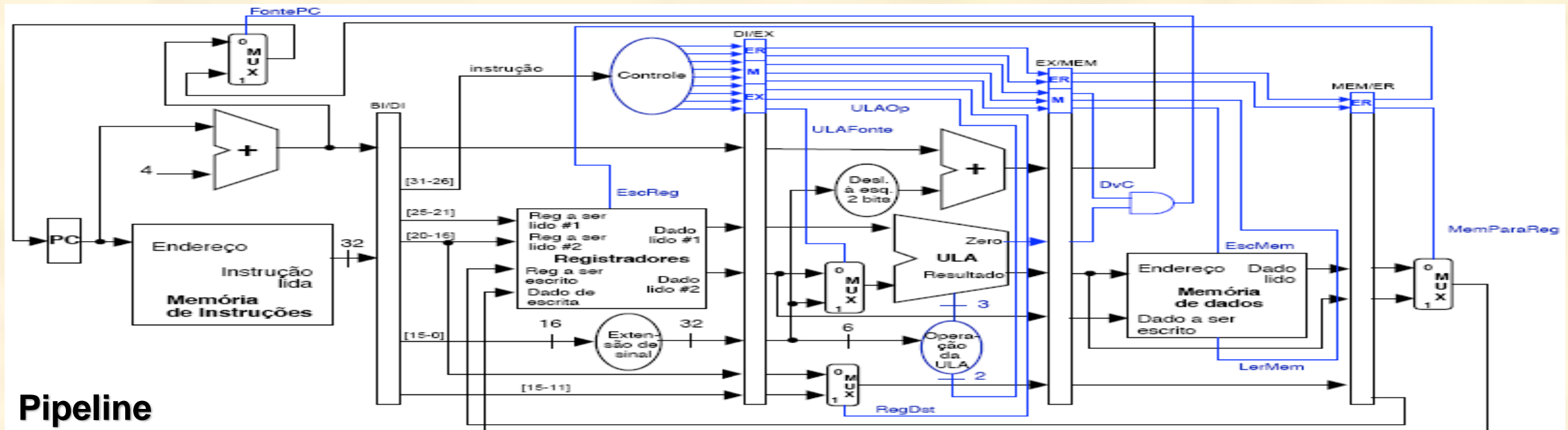
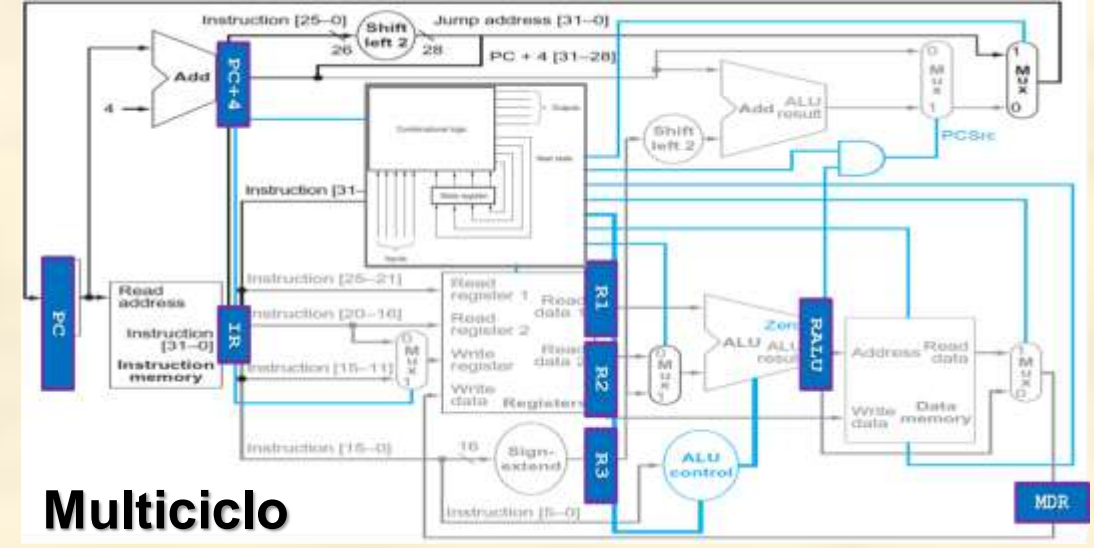
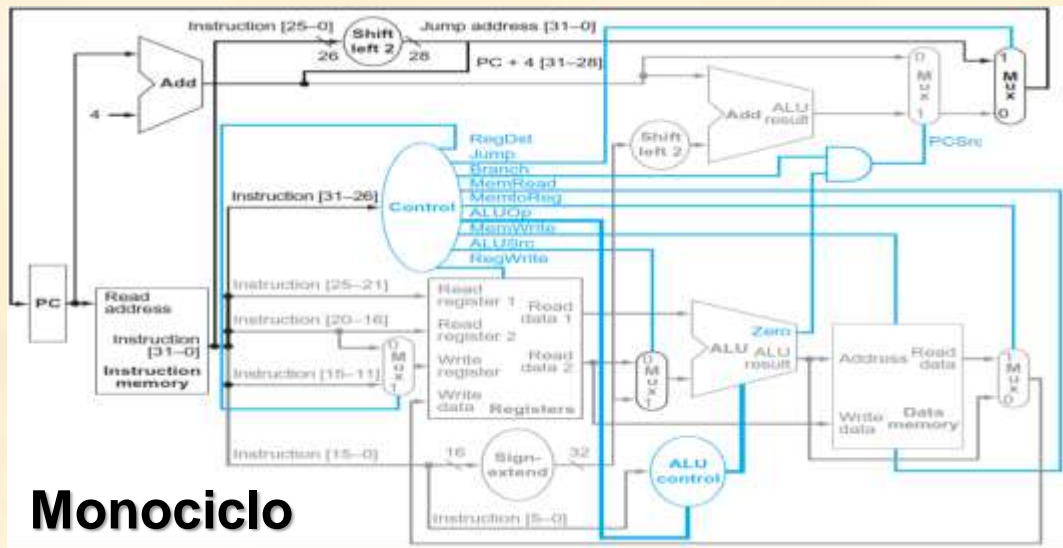
$$t = (8 \text{ ns} + 27 \text{ ns} + 5 \text{ ns}) = 40 \text{ ns}$$

$$f = 1 / t = 1 / 40 \times 10^{-9} = 25 \text{ MHz}$$



Organizações MIPS – Maior Detalhamento

Monociclo, Multiciclo, Pipeline

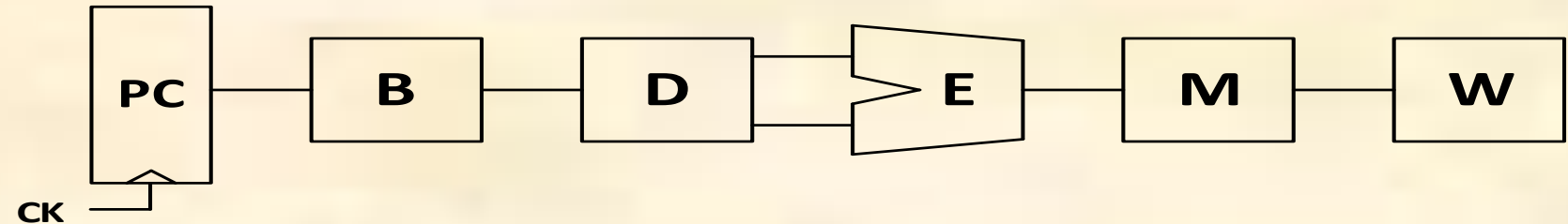


Comparação das Organizações MIPS – Maior Abstração

Monociclo, Multiciclo e Pipeline

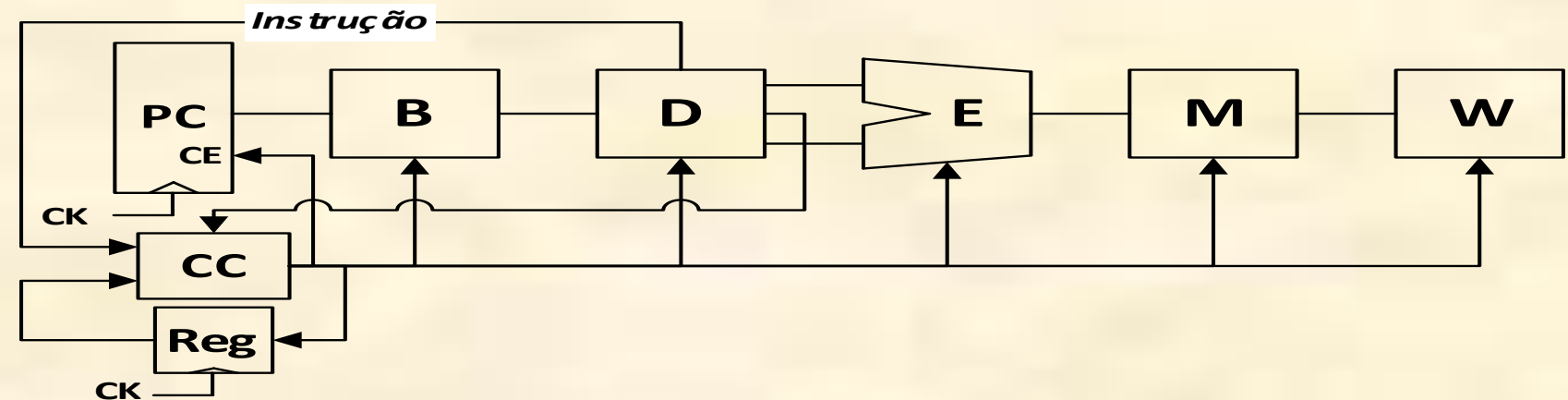
Barreiras Temporais

Monociclo
(apenas o PC)



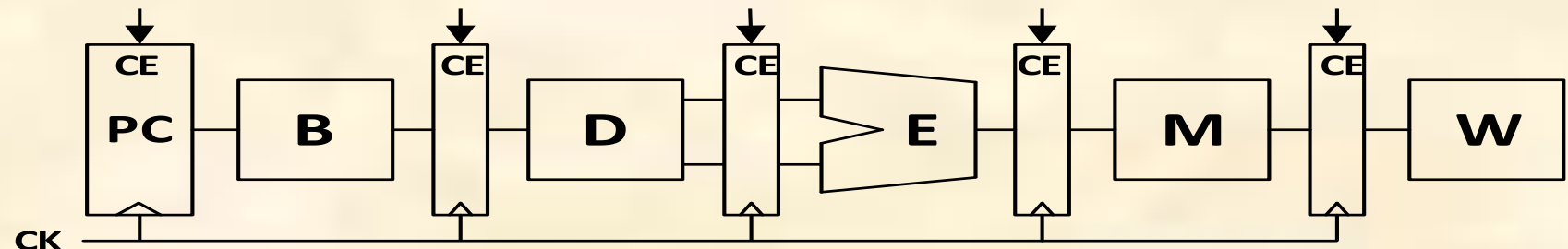
Multiciclo

(PC e maq. de estados finita)



Pipeline

(PC e reg. entre estados)



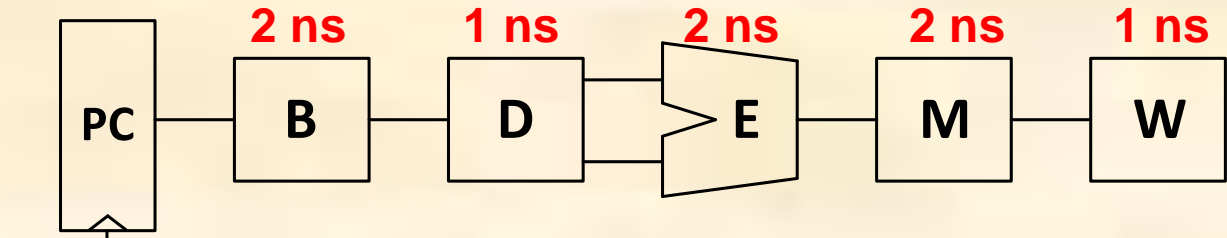
Comparação das Organizações MIPS

Monociclo, Multiciclo e Pipeline

Latência (L) / período (T) das arquiteturas

Monociclo

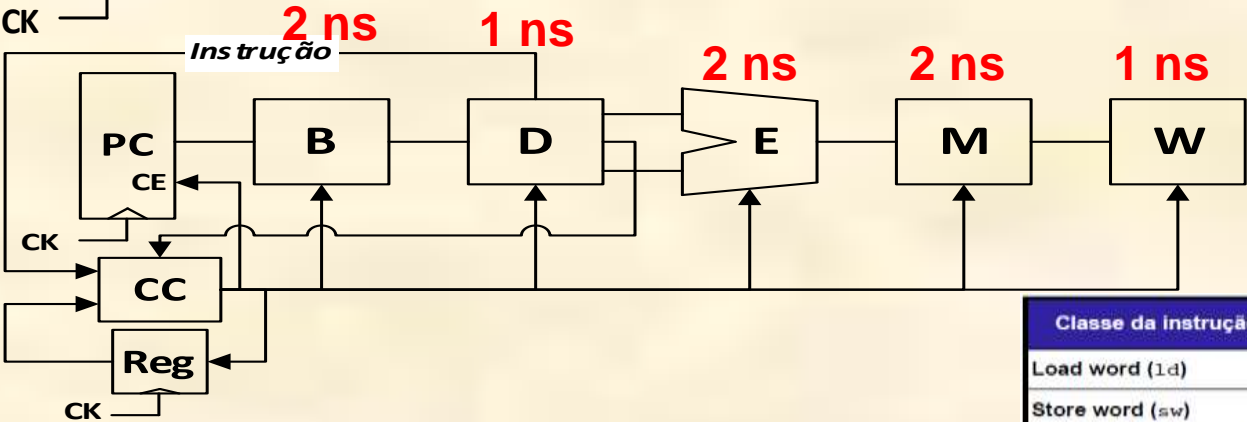
$L = T = 8\text{ ns}$



Multiciclo

$L = 5, 6, 7, 8\text{ ns}$
 $T = 1\text{ ns}^*$

*Estágio 2ns requer divisor de clock

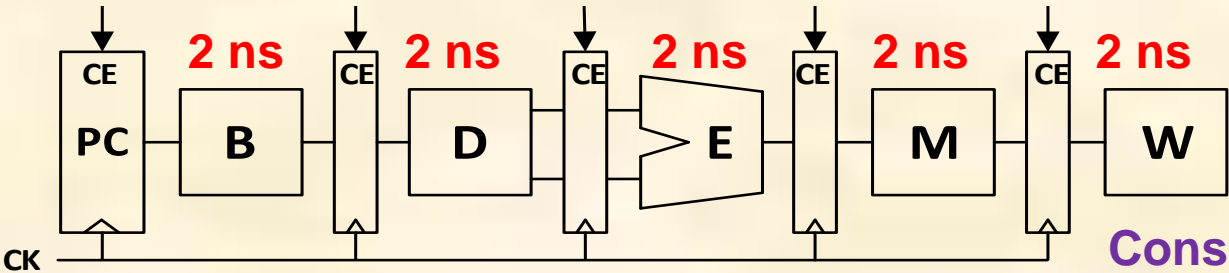


Consideração – FSM com 0 ns

Classe da instrução	Busca da instrução	Leitura de registrador	Operação na ULA	Acesso ao dado	Escrita no registrador	Tempo total
Load word (ld)	2 ns	1 ns	2 ns	2 ns	1 ns	8 ns
Store word (sw)	2 ns	1 ns	2 ns	2 ns	---	7 ns
Formato R (add, sub, and, or, slt)	2 ns	1 ns	2 ns	---	1 ns	6 ns
Branch (beq)	2 ns	1 ns	2 ns	---	---	5 ns

Pipeline

$L = 10\text{ ns}$
 $T = 2\text{ ns}$



Consideração – barreira temporal com 0 ns

Comparação das Organizações MIPS

Execução de um Trecho de Programa em Organização **Monociclo**

Trecho de Programa MIPS:

```
main:  lw    $t2, 0($t1)
      addi   $t3, $zero, 1
      beq    $t4, $zero, fim
      sw     $t5, 0($t1)

fim:
```

Organização Monociclo:

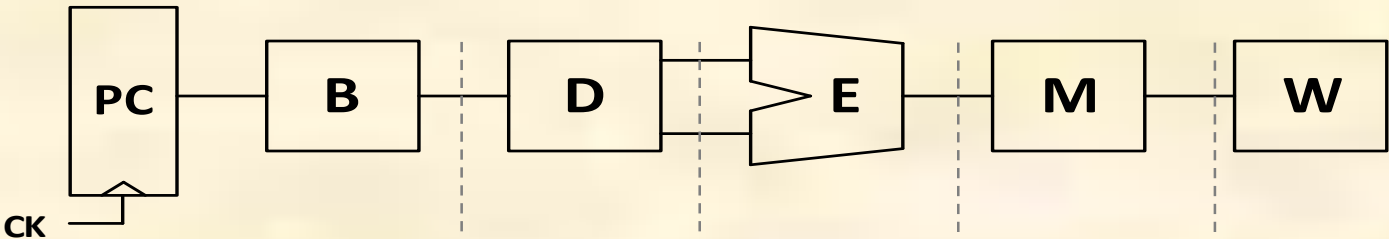


Diagrama de Instruções (Estágio x Ciclo):

Cicl	B	D	E	M	W	Tempo	Tempo acumulado
1	LW	LW	LW	LW	LW	8	8
2	ADDI	ADDI	ADDI	ADDI	ADDI	8	16
3	BEQ	BEQ	BEQ	BEQ	BEQ	8	24
4	SW	SW	SW	SW	SW	8	32

Comparação das Organizações MIPS

Execução de um Trecho de Programa em Organização Multiciclo

Trecho de Programa MIPS:

```
main:    lw      $t2, 0($t1)
        addi    $t3, $zero, 1
        beq     $t4, $zero, fim
        sw      $t5, 0($t1)

fim:
```

Organização Multiciclo:

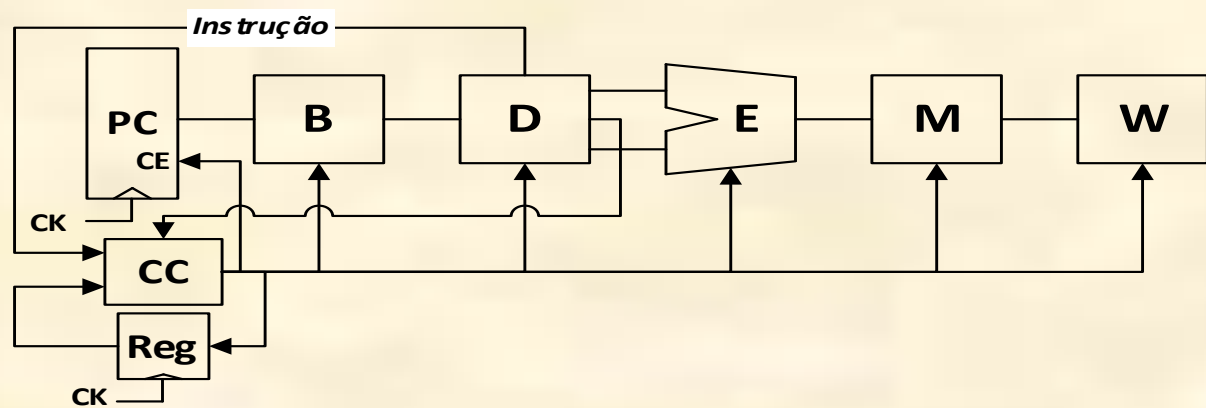


Diagrama de Instruções (Estágio x Ciclo):

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW					2	2
2		LW				1	3
3			LW			2	5
4				LW		2	7
5					LW	1	8
6	ADDI					2	10
7		ADDI				1	11
8			ADDI			2	13
9					ADDI	1	14
10	BEQ					2	16
11		BEQ				1	17
12			BEQ			2	19
13	SW					1	20
14		SW				2	22
15			SW			2	24
16				SW		2	26

Comparação das Organizações MIPS

Execução de um Trecho de Programa em Organização Pipeline

Trecho de Programa MIPS:

```
main:  lw      $t2, 0($t1)
      addi   $t3, $zero, 1
      beq    $t4, $zero, fim
      sw     $t5, 0($t1)

fim:
```

Organização Pipeline:

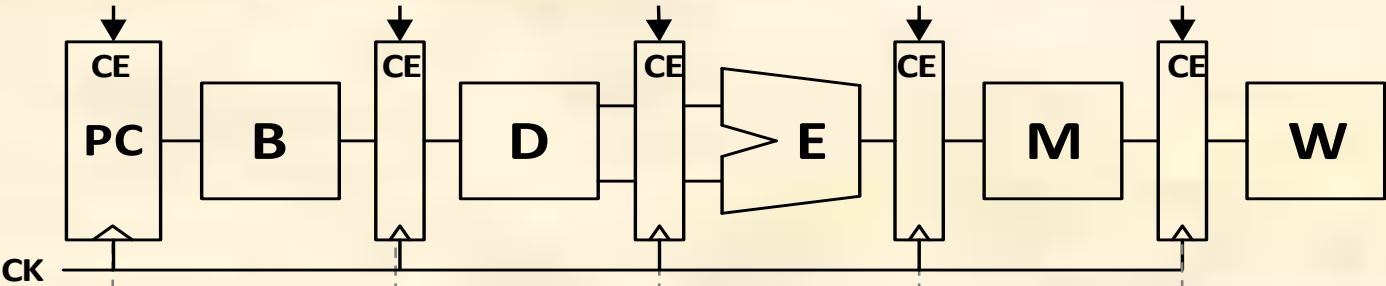


Diagrama de Instruções (Estágio x Ciclo):

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW					2	2
2	ADDI	LW				2	4
3	BEQ	ADDI	LW			2	6
4	SW	BEQ	ADDI	LW		2	8
5		SW	BEQ	ADDI	LW	2	10
6			SW	BEQ	ADDI	2	12
7				SW	BEQ	2	14
8					SW	2	16

Comparação das Organizações MIPS - Monociclo, Multiciclo e Pipeline

Monociclo

Multiciclo

Pipeline

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW	LW	LW	LW	LW	8	8
2	ADDI	ADDI	ADDI	ADDI	ADDI	8	16
3	BEQ	BEQ	BEQ	BEQ	BEQ	8	24
4	SW	SW	SW	SW	SW	8	32

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW					2	2
2		LW				1	3
3			LW			2	5
4				LW		2	7
5					LW	1	8
6	ADDI					2	10
7		ADDI				1	11
8			ADDI			2	13
9					ADDI	1	14
10	BEQ					2	16
11		BEQ				1	17
12			BEQ			2	19
13	SW					1	20
14		SW				2	22
15			SW			2	24
16				SW		2	26

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW					2	2
2	ADDI	LW				2	4
3	BEQ	ADDI	LW			2	6
4	SW	BEQ	ADDI	LW		2	8
5		SW	BEQ	ADDI	LW	2	10
6			SW	BEQ	ADDI	2	12
7				SW	BEQ	2	14
8					SW	2	16

Normalizado
por tempo

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW	LW	LW	LW	LW	8	8
2	ADDI	ADDI	ADDI	ADDI	ADDI	8	16
3	BEQ	BEQ	BEQ	BEQ	BEQ	8	24
4	SW	SW	SW	SW	SW	8	32

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW					2	2
2		LW				1	3
3			LW			2	5
4				LW		2	7
5					LW	1	8
6	ADDI					2	10
7		ADDI				1	11
8			ADDI			2	13
9					ADDI	1	14
10	BEQ					2	16
11		BEQ				1	17
12			BEQ			2	19
13	SW					1	20
14		SW				2	22
15			SW			2	24
16				SW		2	26

Ciclo	B	D	E	M	W	Tempo	Tempo acumulado
1	LW					2	2
2	ADDI	LW				2	4
3	BEQ	ADDI	LW			2	6
4	SW	BEQ	ADDI	LW		2	8
5		SW	BEQ	ADDI	LW	2	10
6			SW	BEQ	ADDI	2	12
7				SW	BEQ	2	14
8					SW	2	16

Normalizado
por ciclo

**Multiciclo implementada com
ciclos de tempos variáveis*

Exercício

1. **(POSCOMP 2003 - 22)** Para que serve a segmentação de um processador (pipelining)?
- a) Permitir a execução de mais de uma instrução por ciclo de relógio
 - b) Aumentar a velocidade do relógio
 - c) Simplificar o conjunto de instruções
 - d) Reduzir o número de instruções estáticas nos programas
 - e) Simplificar a implementação do processador

Resposta

1. (POSCOMP 2003 - 22) Para que serve a segmentação de um processador (pipelining)?

a) Permitir a execução de mais de uma instrução por ciclo de relógio

b) Aumentar a velocidade do relógio

c) Simplificar o conjunto de instruções

d) Reduzir o número de instruções estáticas nos programas

e) Simplificar a implementação do processador

- OBS.: A resposta **b** também seria adequada!!

Exercício

2. **(POSCOMP 2004 - 30)** Ao segmentar um processador, transformando-o num pipeline, obtém-se:
- a) Redução no número de ciclos necessários para executar uma instrução
 - b) Redução no número de ciclos necessários para executar um programa
 - c) Redução no número de ciclos necessários para tratar uma exceção
 - d) Redução no número de ciclos necessários para tratar uma interrupção
 - e) O circuito do processador fica mais simples

Resposta

2. (POSCOMP 2004 - 30) Ao segmentar um processador, transformando-o num pipeline, obtém-se:

- a) Redução no número de ciclos necessários para executar uma instrução
- b) Redução no número de ciclos necessários para executar um programa**
- c) Redução no número de ciclos necessários para tratar uma exceção
- d) Redução no número de ciclos necessários para tratar uma interrupção
- e) O circuito do processador fica mais simples

- OBS.: para a resposta b ser correta é necessário trocar “número de ciclos” por tempo de execução, pois o número de ciclos de um pipeline é sempre superior que o de uma arquitetura monociclo! O que reduz é o período do relógio, conseqüentemente, o tempo de execução do programa

Exercício

3. **(POSCOMP 2014, Questão 45)** Sobre pipelines, assinale a alternativa correta.
- a) Cada estágio do pipeline possui seu próprio tempo de duração
 - b) Um pipeline precisa de registradores para armazenar dados entre estágios
 - c) Dependências de dados irão paralisar o pipeline
 - d) O pipeline é paralisado ao executar uma instrução de desvio
 - e) O tempo de leitura de uma instrução é maior que o tempo de execução

Resposta

3. (POSCOMP 2014, Questão 45) Sobre pipelines, assinale a alternativa correta.
- a) Cada estágio do pipeline possui seu próprio tempo de duração
 - b) Um pipeline precisa de registradores para armazenar dados entre estágios**
 - c) Dependências de dados irão paralisar o pipeline
 - d) O pipeline é paralisado ao executar uma instrução de desvio
 - e) O tempo de leitura de uma instrução é maior que o tempo de execução

Exercício

4. **(POSCOMP 2011 - 28)** Um processador RISC é implementado em duas versões de organização síncrona: uma monociclo, em que cada instrução executa em exatamente um ciclo de relógio, e uma versão pipeline de 5 estágios. Os estágios da versão pipeline são: (1) busca de instrução, (2) busca de operandos, (3) execução da operação, (4) acesso à memória e (5) atualização do banco de registradores. A frequência máxima de operação das organizações foi calculada em 100 MHz para a versão monociclo e 400 MHz para a versão pipeline. Um programa X que executa 200 instruções é usado para comparar o desempenho das organizações. Das 200 instruções, apenas 40% fazem acesso à memória, enquanto as demais operam apenas sobre registradores internos da organização. Considere que o programa não apresenta nenhum conflito de dados ou de controle entre instruções que podem estar simultaneamente dentro do pipeline da segunda organização. Assim, o tempo de execução do programa X nas organizações monociclo e pipeline é, respectivamente:
- a) 2.000 nanossegundos e 510 nanossegundos
 - b) 2.000 nanossegundos e 500 nanossegundos
 - c) 2.000 nanossegundos e 2.300 nanossegundos
 - d) 2.300 nanossegundos e 500 nanossegundos
 - e) 2.300 nanossegundos e 510 nanossegundos

Resposta

4. **(POSCOMP 2011 - 28)** Um processador RISC é implementado em duas versões de organização síncrona: uma monociclo, em que cada instrução executa em exatamente um ciclo de relógio, e uma versão pipeline de 5 estágios. Os estágios da versão pipeline são: (1) busca de instrução, (2) busca de operandos, (3) execução da operação, (4) acesso à memória e (5) atualização do banco de registradores. A frequência máxima de operação das organizações foi calculada em 100 MHz para a versão monociclo e 400 MHz para a versão pipeline. Um programa X que executa 200 instruções é usado para comparar o desempenho das organizações. Das 200 instruções, apenas 40% fazem acesso à memória, enquanto as demais operam apenas sobre registradores internos da organização. Considere que o programa não apresenta nenhum conflito de dados ou de controle entre instruções que podem estar simultaneamente dentro do pipeline da segunda organização. Assim, o tempo de execução do programa X nas organizações monociclo e pipeline é, respectivamente:

- a) **2.000 nanossegundos e 510 nanossegundos**
- b) 2.000 nanossegundos e 500 nanossegundos
- c) 2.000 nanossegundos e 2.300 nanossegundos
- d) 2.300 nanossegundos e 500 nanossegundos
- e) 2.300 nanossegundos e 510 nanossegundos